

大学计算机 与新一代信息技术

云课堂版

安徽省高等学校计算机教育研究会推荐用书
新时代大学计算机课程分级分类教学创新系列教材
新形态数字化教材 丛书总主编 胡学钢 王浩

主编 接标 王秀友 蔡庆华 常雪琴

 上海交通大学出版社
Shanghai Jiao Tong University Press

CS 扫描全能王

3亿人都在用的扫描App

- 模块 5 网络基础与信息检索 83
 - 5.1 计算机网络基础 84
 - 5.2 Internet 技术 93
 - 5.3 信息检索 98
 - 思维启发 104
 - 习题 5 105

进阶篇 算法与数据管理

- 模块 6 程序设计基础 100
 - 6.1 程序的构成要素和编码规范 110
 - 6.2 结构化程序设计 121
 - 6.3 函数与模块 128
 - 思维启发 134
 - 习题 6 135
- 模块 7 数据结构与算法 137
 - 7.1 算法基础 138
 - 7.2 常用的数据结构 142
 - 7.3 常用的算法 154
 - 思维启发 163
 - 习题 7 164
- 模块 8 软件开发与过程管理 166
 - 8.1 软件开发过程的工程化 167
 - 8.2 软件生命周期与软件过程模型 172
 - 8.3 软件项目管理 176
 - 8.4 软件工具与开发环境 182
 - 思维启发 188
 - 习题 8 189
- 模块 9 数据库设计与管理 191
 - 9.1 数据库基础知识 192
 - 9.2 数据模型 194
 - 9.3 关系代数 196
 - 9.4 关系数据库设计方法 198
 - 思维启发 204
 - 习题 9 205

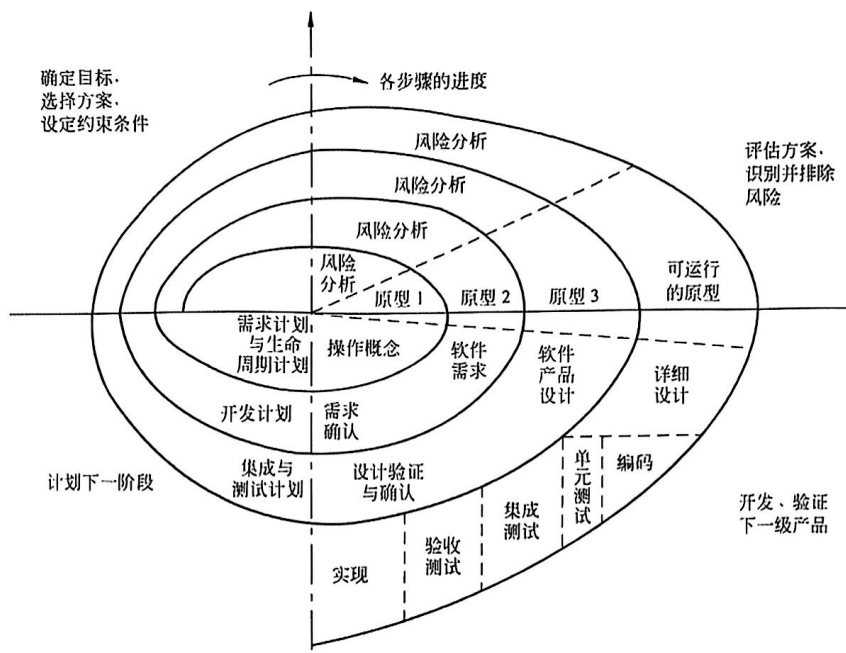


图 8-7 螺旋模型

通过不断地迭代和风险管理，螺旋模型可以逐步完善软件系统，并减少项目失败的风险。螺旋模型强调在开发过程中及时纠正问题和调整方向，以确保软件系统的质量和满足用户需求。螺旋模型适用于大型和复杂的软件项目，尤其是在需求不确定或变化频繁的情况下，它能够更好地应对项目中的风险和不确定性，并提供更灵活的开发方法。然而，螺旋模型的开发过程相对复杂，需要有经验的开发团队和有效的风险管理策略。



思考题

软件什么时候会彻底退役？

8.3 软件项目管理

在软件开发过程中，项目管理在确保项目按时、按质地交付，并满足客户需求和预期等方面起着关键的作用。本节将涵盖过程管理、人员管理和制品管理等项目管理的各个方面。

8.3.1 软件项目管理概述

软件项目通常由一个或多个软件开发团队组成，他们协同工作以实现项目的目标。软件项目的目标可以是开发一个新的软件产品，对现有软件进行改进或升级，或者解决特定的业务需求。软件项目管理是指在完成软件项目目标的过程中，对项目进行规划、组织、协调和控制的活动。它涉及对项目范围、时间、成本、质量、风险和资源等方面的管理，以确保软件项目能够按时、按质量、按预算地完成。

软件项目有明确的目标和交付物，如开发一个特定的软件应用程序或系统。

软件项目在时间、成本和资源方面都有限制。项目团队需要合理分配和管理这些资源，以确保项目的成功交付。

软件项目中的需求往往会随着时间的推移而变化。项目团队需要灵活应对需求变更，并及时调整项目计划和开发过程。

软件项目面临各种风险，如技术风险、进度风险和需求风险。项目团队需要进行风险评估和管理，以减少风险对项目的影响。

软件项目通常需要多个团队成员协同工作。有效的团队协作和沟通对项目成功至关重要。

软件项目管理是对软件项目开发过程中所涉及的过程、人员和制品等要素进行度量和规划、组织和控制的过程，从而确保项目能够按时、按质量、按预算地交付，并满足客户的需求和期望。软件项目管理主要关注过程、人员和制品这三方面的对象。按照软件项目管理关注的对象类别划分，软件项目管理的主要内容如下：

过程管理是指对项目开发过程进行规划、执行和监控的活动，关注项目中各个阶段和活动的顺序、交付物、质量和效率，从而确保项目按照计划进行，并符合相关的标准和规范。

人员管理是指对项目团队成员进行组织、培训、激励和管理的过程，确保团队成员具备所需的技能和能力，并能够高效地协作和执行项目任务。

制品管理是指对软件开发过程中产生的各种制品包括代码、文档、测试用例、构建产物等，进行有效的管理和控制，确保项目的制品可追踪、可控和可重复。

通过对过程、人员和制品进行管理，软件项目管理能够提高项目的可控性和可预测性，确保项目按时交付、符合质量要求，并最大程度地满足客户需求。下面将逐一介绍软件项目管理的各个内容。

8.3.2 过程管理

软件项目当中的过程管理是指对项目开发过程进行规划、执行和监控的活动。它关注项目中各个阶段和活动的顺序、交付物、质量和效率。过程管理的目标是确保项目按照计划进行，并符合相关的标准和规范。通过合理和有效地管理项目过程，可以提高项目的可控性和可预测性，减少风险和延误，并提高项目的成功交付率。受限于篇幅限制，本小节只对项目规划阶段部分任务进行介绍。

1. 软件度量和估算

软件度量是通过定量的方法来描述软件开发过程和制品的简单属性。此处，过程是指各种软件开发活动，如需求分析、设计等，制品是指软件开发过程中所生成的各种文档、数据和程序等。软件度量关注的是已有的数据和指标，通过收集和分析这些指标，可以评估项目的进度、质量和效率，并进行合理的资源分配和决策。常见的软件度量指标包括代码行数、功能点数、文档页数等。软件度量的目的是提供客观的数据支持，帮助项目团队做出准确的决策，发现和解决潜在的问题和风险，并促进团队的学习和持续改进。

软件估算则是在软件开发活动实施前，对软件项目的开发成本、工作量等进行估计。软件估算是针对软件过程和制品的复杂属性的定量描述。估算的目的是帮助软件项目团队签署和项目计划制定。软件估算涉及预测和估计软件开发所需的资源、时间和成本。估算的过程依赖于经验、历史数据和专业知识，以提供对软件项目的合理估计。由于软件是逻辑产品，对其进行估算是一项非常困难的工作。下面介绍一些常用的估算方法。

（1）基于代码行和功能点的估算。通过统计代码行数或功能点数来估计软件开发的工作量和成本。这种方法基于项目的规模来进行估算，但需要具备足够的历史数据和经验支持。

（2）基于经验模型的估算。使用经验模型，如构造性成本模型（constructive cost model），根据软件项目的特征和规模来估计工作量和成本。模型基于已有的经验和统计数据，可以提供相对准确的估算结果。

（3）类比估算。将当前软件项目与过去类似项目进行比较，利用类似项目的估算数据来估计当前项目的工作量和成本。这种方法适用于具有相似特征和规模的项目。

软件度量和软件估算相互关联，它们共同支持软件项目的管理和决策。软件度量提供了对项目过程和产品的实际数据，为软件估算提供依据和参考。准确的软件估算能够为软件度量提供预期目标和基准，使得度量结果能够与估算结果进行比较和评估。通过度量和估算的结合，项目团队可以更好地了解项目的进展情况，及时调整计划和资源分配，并确保项目按时、按质量和按预算完成。

2. 软件项目计划

软件项目计划是指在软件项目开始之前制订的一份详细计划，用于指导和管理软件项目的执行。软件项目计划对软件项目实施所涉及的活动、资源、任务、进度等方面做出预先规划。它是项目管理的重要组成部分，为项目团队提供了一个框架，以确保项目实施的高生产率、软件制品的及时交付和客户的高满意度。一个有效的软件项目计划需要考虑以下方面的内容：

（1）明确软件项目所采用的软件过程。软件过程定义了软件项目开发所需经历的阶段和步骤，以及在每个阶段中执行的活动和任务。

（2）明确软件项目的目标和范围，包括确定项目的交付物、功能需求、非功能需求等。这有助于确保项目的重点和方向。

（3）确定软件项目所需的资源，并进行合理的分配和调度。这包括确定项目团队的组成、技能需求和角色分工，以及项目所需的设备和工具。

（4）在制定软件项目计划时，需要与项目团队和利益相关者进行充分的沟通和协商。计划应该是可行的、具体的，并根据项目的实际情况进行调整和更新。同时，项目计划应该是灵活的，能够适应变化和调整。在项目执行过程中，需要不断监控和更新项目计划，及时调整和采取措施，以确保项目的成功实施。

软件项目计划可以使用不同的描述方式，具体选择哪种方式取决于项目的特点、团队的偏好和组织的要求。甘特图和网络图是两种常见的项目计划描述方式。

甘特图是一种以时间为基准的条形图，用于显示项目的任务、里程碑和关键路径。图 8-8 是甘特图的示意图。图中每个任务在图表上表示为一个条形，其长度表示任务的持续时间，条形的位置表示任务的开始和结束日期。甘特图直观地展示了项目的时间安排和任务依赖关系。

网络图使用节点和箭头表示项目任务之间的逻辑关系和依赖关系。图 8-9 是网络图的示意图。图中每个任务表示为一个节点，箭头表示任务之间的先后关系。逻辑网络图可以帮助识别项目的关键路径和任务间的依赖关系。

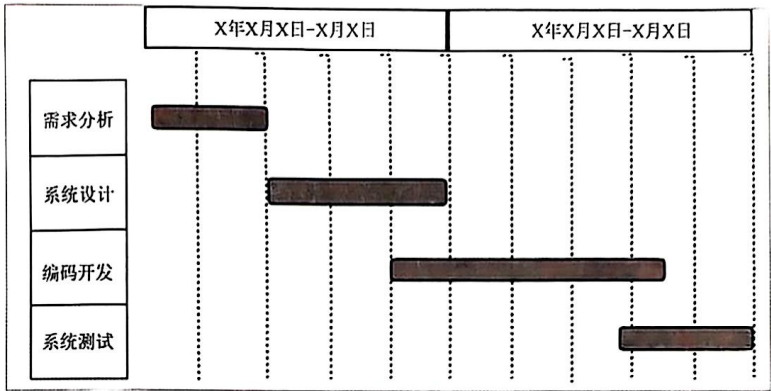


图 8-8 甘特图

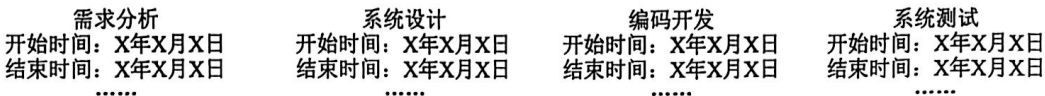


图 8-9 网络图

除了以上描述方式，还可以结合文字说明、图表、图形和其他可视化工具来展示项目计划。无论选择哪种描述方式，关键是确保计划清晰、易于理解，并能够为项目团队和利益相关者提供明确的指导。

8.3.3 人员管理

软件项目的人员管理是指在软件开发项目中，对项目团队成员进行有效的管理和协调，以确保项目的顺利进行和成功交付。通过合理的人员管理，可以建立高效的团队，提高团队成员的工作动力和满意度，从而促进项目的成功实施。

1. 团队结构

团队结构是人员管理中的重要方面。它涉及确定项目团队的组成、角色和职责分配。一个良好的团队结构应该能够充分发挥每个成员的专长和能力，实现协同工作和高效沟通，提高团队的工作效率和质量，减少沟通和协调的障碍，从而推动软件项目的成功实施。以下是 3 种常见的团队结构。

(1) 主程序员团队。这种结构以主程序员为核心，其他成员通常是相对较新的开发人员。主程序员负责项目的整体设计和架构，并指导其他团队成员进行开发工作。这种结构适用于小型项目或需要高度专业知识和经验的项目。

(2) 民主团队。这种结构强调平等和民主，团队成员共同参与项目决策和任务分配。每个成员都有平等的发言权和决策权，团队成员之间的合作和协商是基础。这种结构适用于注重团队合作和创新的项目，能够激发团队成员的积极性和创造力。

(3) 开放团队。这种结构注重团队成员的自主性和创造力。团队成员可以自由选择自己感兴趣的任务和角色，并且可以自由地参与项目的决策和规划。这种结构鼓励团队成员的自主性和创造力，能够激发团队成员的潜力和创新能力。但管理者无法得知工作进展，当然创造性的工作本身也无法掌握进展。这种结构适用于强调个人自由和创新的项目，能够吸引有独特才能和创造力的团队成员。

图 8-10 为主程序员团队、民主团队和开放团队的示意图。

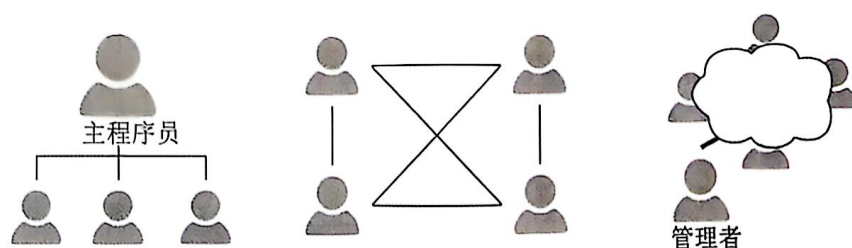


图 8-10 从左到右依次为主程序员团队、民主团队、开放团队

2. 团队管理

团队管理是确保团队整体优于部分之和的关键，它不仅能提高生产力，还能使团队成员从工作中获得快乐。每个团队成员都有自己的角色和责任，因此需要确保每个人清楚自己的职责和任务。通过有效的团队建设和管理，可以提高团队的工作效率、凝聚力和成果质量。持续的成功有助于积累团队的信心，建立团队成员之间的信任，并激励团队士气。为了取得项目的成功，团队成员之间需要建立和谐的交流沟通机制。和谐的沟通需要制度保障，以确保团队成员之间正确且及时地进行沟通。团队管理的关键实践如下。

- (1) 建立良好的沟通和协作机制，确保团队成员之间可以有效沟通和信息共享。
- (2) 定期召开会议、进行团队建设活动，促进团队合作和协作精神。
- (3) 与团队成员共同设定项目目标和里程碑，激励团队成员为实现目标而努力工作。
- (4) 提供适当的奖励和认可，以鼓励团队成员的积极表现和贡献。
- (5) 提供培训和发展机会，帮助团队成员提升技能和知识。
- (6) 支持团队成员参加培训课程、研讨会和专业认证，以提高团队整体的能力水平。
- (7) 定期进行绩效评估，对团队成员的工作表现进行评估和反馈。
- (8) 识别和表彰优秀的表现，提供改进和发展的建议。
- (9) 及时处理和解决团队成员之间的冲突和问题。
- (10) 鼓励开放的沟通和合作，寻求共识和解决方案，以维护团队的和谐和效率。

通过遵循这些实践，可以有效管理团队，促进团队成员之间的合作与协作，从而实现项目的成功。对于不同类型的团队，建设和管理的重点可能会有所不同。然而，共同的原则是确保团队成员具备适当的技能和知识，建立良好的沟通和协作机制，提供培训和发展机会，以及激励和认可团队成员的贡献。

8.3.4 制品管理

软件项目的制品管理是指为了解和控制制品的变更，确保软件制品的质量等目的，对软件开发过程中产生的各种制品进行有效的管理和控制。这些制品包括代码、文档、测试用例、构建产物等。通过有效的软件质量管理和软件配置管理，可以确保制品的质量、可靠性和可维护性，从而推动软件项目的成功实施和后续维护工作的顺利进行。在制品管理过程中，团队应该采用适当的工具和方法来支持制品的管理和控制，以提高团队的工作效率和制品的质量。

1. 软件配置管理

软件配置管理是制品管理的一个重要方面。它涉及对软件制品进行版本控制、变更管理和配置项管理，以确保制品的一致性和可追溯性。配置项是指软件生命周期内需要进行配置管理的制品。配置项包括源代码、文档、测试用例、配置文件等。软件配置管理的主要内容如下。

(1) 配置项标识。识别和描述软件系统中存在的配置项，以便能够对其进行管理和追踪，确保它们的一致性和可追溯性。配置项标识应该能够唯一地标识每个配置项，并提供必要的信息以区分不同的配置项。配置项标识通常包括配置项名称、版本号、修订号。配置项的描述通常包括配置项的版本号、作者、创建日期、评审者、所依赖的配置项等。这些信息有助于跟踪配置项的历史记录和变更情况。

(2) 版本控制。使用版本控制系统（如 Git、SVN，将在 8.4.2 介绍相关内容）可以对软件制品进行管理，包括对代码、文档和构建产物的管理等。通过版本控制，可以跟踪和控制制品的变更历史，包括创建时间、修改时间、作者等信息。版本控制还提供了并行开发和协作的能力。多个开发人员可以在同一个配置项上进行独立的修改，并通过版本控制系统合并这些修改，确保不会发生冲突和数据丢失。

(3) 变更管理。建立变更管理流程，识别和记录所有的变更请求和建议，包括功能需求变更、缺陷修复、技术改进等；从变更的技术可行性、资源需求、时间计划等方面进行评估和分析，确定变更的合理性、影响范围和风险；对变更请求进行审查，并评估其对项目进度和资源的影响；批准变更请求后，将其分配给相应的开发人员或团队实施，包括对软件进行修改、编码、测试和验证，确保变更的正确性和质量；及时向利益相关者和团队成员通知变更的实施和影响；记录和追踪所有的变更请求和实施情况，包括变更的状态、进度、结果等。

2. 软件质量管理

质量管理是指在整个项目开发过程中，通过制订质量标准和实施质量控制措施，确保项目交付的软件产品满足客户的需求和预期。质量管理包括制订质量策略和计划、制订质量标准和指标、执行质量控制活动、进行质量评估和审核等方面。

(1) 质量计划。质量计划是软件项目管理中质量管理的基础。它应该明确定义项目的质量目标和预期的质量标准。质量计划需要明确描述软件项目的质量目标，如性能要求、可用性、可靠性等。质量计划通常使用例如代码覆盖率、缺陷密度、用户满意度等指标来跟踪项目的进展和评估软件的质量水平。为了能及时发现和解决软件中的问题，质量计划需要从缺陷的分类、优先级和状态管理等方面，定义缺陷的报告、跟踪和解决流程。列出计划中代码审查、单元测试、集成测试、系统测试等质量保证活动，明确每个活动的方法和责任人。

(2) 质量保证活动。质量保证活动旨在确保软件开发过程中质量标准的实施，包括代码审查、单元测试、集成测试、验收测试等。代码审查可以通过静态代码分析工具、代码检查列表或专门的代码审查会议进行，进而检查潜在的问题和错误。单元测试通常由开发人员自行完成编写和执行，验证代码的功能和逻辑正确性。在组件或模块集成后进行集成测试，用于验证组件或模块间的协作和整体功能。同时会通过功能测试、性能测试、安全

测试等方式，对整个软件系统进行全面的测试，以验证其符合用户需求和质量标准。在交付前，将由最终用户或客户执行验收测试，验证软件是否满足其需求并符合预期的质量水平。

（3）缺陷管理。缺陷管理是质量管理中的核心环节。缺陷管理需要依赖缺陷报告，项目成员应该能够报告发现的缺陷，包括描述缺陷的详细信息、重现步骤和相关环境。同时，项目团队需要建立一个缺陷跟踪系统，用于记录和跟踪缺陷的状态、优先级和解决进度，从而帮助团队及时响应和解决缺陷。

（4）过程改进。软件项目管理还应关注过程改进，以提高软件开发过程的质量和效率。过程改进收集缺陷数据、测试数据、用户反馈等质量数据。通过分析这些数据，可以识别出问题和瓶颈，找出导致问题出现的根本原因，并找到有针对性的改进措施。根据数据和分析结果，可以对软件开发过程进行优化和改进。这可能涉及调整工作流程、引入新的工具和技术、改进团队协作等。



思考题

你能否列举出所有的软件制品，并逐个解释其作用？

8.4 软件工具与开发环境

软件工具与开发环境指的是在软件开发过程中使用的各种工具和环境，用以提高开发效率、协作和管理软件项目。这些工具和环境涵盖开发、问题追踪、版本控制、构建和部署等多个方面。

8.4.1 集成开发环境

集成开发环境（IDE）就像是开发人员的工作室，提供了一系列方便的工具，让开发人员可以更轻松地编写、调试和管理代码。它们有很多有用的功能，如智能代码补全、语法高亮、代码重构和调试功能，这些都能帮助开发人员更快地写出准确的代码，减少错误。常见的集成开发环境有 Visual Studio Code、IntelliJ IDEA、PyCharm 等。选择适合的集成开发环境可以提高开发效率和代码质量。

1. Visual Studio Code

Visual Studio Code 是一款由微软开发的轻量级但功能强大的集成开发环境，适用于 Windows、macOS 和 Linux 等操作系统。它内置了对 JavaScript、TypeScript 和 Node.js 的支持，并为其他程序设计语言（如 C++、C#、Java、Python、PHP、Go、.NET）提供了丰富的扩展生态系统。Visual Studio Code 设计简单，运行迅速，让开发人员可以高效地编写代码，还有许多插件和主题可供下载，满足不同的开发需求。Visual Studio Code 界面如图 8-11 所示。