


- 
- 19.1 软件配置管理目的
  - 19.2 配置管理内容
  - 19.3 配置管理系统
  - 19.4对修改的管理
  - 19.5版本管理
  - 19.6配置管理工作的度量
  - 19.7总结

# 19.1 软件配置管理目的

- 影响软件正确和完整的主要情况有：
  - 1) 同时修改：
    - 多个程序员同时修改一段代码
  - 2) 共享代码：
    - 几个程序员共同完成的一段共享代码中的错误被修改
  - 3) 公用代码：
    - 在大项目中，一段公用代码被修改后，没有通知到每个程序
  - 4) 多版本：
    - 一个软件版本是一个多个不同的小代码版本不断进化和发布的过程
  - 5) 文档变更：
    - 需求变更会导致文档的变更、代码的修改

- 这些问题完全是项目组织管理不到位或人员之间的工作误解造成的。
  - 1) 当前的软件配置是什么？
  - 2) 每个配置的状态是什么？
  - 3) 如何控制每个配置的修改？
  - 4) 如何把修改情况通知到每个干系人？
  - 5) 哪些更改对本项目或他人的工作会有哪些影响/
  - 6) 其他人的修改是否会影响到本项目？

# 软件配置管理的目的

- 控制上述问题涉及到的修改活动。
- 建立一个正式的管理系统帮助开发人员控制和追踪这些工作及其变更，保证没有不知道的变更和破坏，从而在整体上降低项目的误解和返工成本。
- 解决了团队工作中的误解，就能提高软件开发工作效率，提高最终软件产品的完整性，降低产品的缺陷率。

## 19.2 配置管理内容

- 19.2.1 项目开发过程中的配置项
- 19.2.2 配置管理的基本内容
- 19.2.3 SCM的流程
- 19.2.4 配置管理中的角色

## 19.2.1 项目开发过程中的配置项

- 每个活动输出的中间或最终产品可以是：
  - 1) 计算机程序(源代码和执行码);
  - 2) 软件文档; 以及
  - 3) 数据或其他内容(如调试工具等)。
- 与传统工业生产类比：
  - 这些输出产品都是可配置项(Configurable Items)
  - 通过对这些配置项进行不同的配置, 可以得到各种软件产品版本, 包括错误的和“正确的”版本

# 开发过程中提交物(配置项)的版本变化

工作阶段 配置项	需求关 闭	体系结构 设计关闭	详细设计 关闭	模块测试	系统测试关 闭	系统测试 验收
配置管理计划	1.0	2.0	3.0	4.0	4.0	4.0
测试计划和 测试说明	1.0	2.0	3.0	4.5	5.2	6.3
需求文档	1.0	1.2	1.3	1.4	1.6	1.6
体系结构设计 文档	-	1.0	1.2	1.3	1.3	1.3
详细设计文档	-	-	1.0	1.1	1.2	1.3
用户手册	-	-	-	1.0	1.1	1.1
子系统1	-	-	-	1.0	1.1	1.2
子系统2	-	-	-	1.0	1.1	1.2
编译器	-	-	-	4.3	4.3	4.3
链接器	-	-	-	7.2.5	7.2.5	7.2.5
完整的系统	-	-	-	-	1.0	1.1
发布注释	-	-	-	-	-	1.0

- 从提高软件生产效率的角度讲，如果能够经过管理好这些可配置项，就能组装和创立出新的软件版本，就能更好地满足客户和市场对软件产品能力的需求。
- 如果不能对配置项进行管理，就会误用而导致工作的失误和成本上升。
  - 例如，需求文档的变更，同时有引起了设计、代码、测试用例等一系列的变更，如不掌握这种变更情况，就会导致发布出的软件版本出现错误。

## 19.2.2 配置管理的基本内容

- 每个软件配置项(版本), 就像传统工业的一个零部件版本
- 管理人员把各种类型和用途的零部件进行:
  - 1) 编号,
  - 2) 控制对零部件的修改,
  - 3) 对零部件的状态进行记录, 以及
  - 4) 检查和评审一个产品中使用的零部件是否一致和完整。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/166241032105010212>