# 第15章、文档和配置管理



**考点123:软件文档**

1、信息系统相关信息(文档)按照要求分类： 正式与非正式文档；按项目周期分类： 开发文档、产品 文档、管理文档；(10 上64)(16上63)(16下61)

**2、软件文档分为三类：开发文档、产品文档、管理文档。**

|  |  |  |
| --- | --- | --- |
| **类型** | **作用** | **文档种类** |
| **开发文档** | **描述开发过程本身** | 可行性研究报告和项目任务书；需求规格说明；功能规格说明；设计规 格说明，包括程序和数据规格说明；开发计划；软件集成和测试计划； 质量保证计划；安全和测试信息。( 18下61) |
| **产品文档** | **描述开发过程的产物** | 培训手册；参考手册和用户指南；软件支持手册；产品手册和信息广告。 |
| **管理文档** | **记录项目管理的信息** | 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； 开发团队的职责定义。 |

**3、文档的质量可以分为四级：**

|  |  |
| --- | --- |
| **文档的分级** | **适用范围** |
| **最低限度文档** **(1级文档)** | 适合开发工作量低于一个人月的开发者自用程序。  该文档应包含程序清单、开发记录、测试数据和程序简介 |
| **内部文档** **(2级文档)** | 可用于没有与其他用户共享资源的专用程序。  2级文档还包括程序清单内足够的注释以帮助用户安装和使用程序 |
| **工作文档** **(3级文档)** | 适合于由同一单位内若干人联合开发的程序，或可被其他单位使用的程序。 |

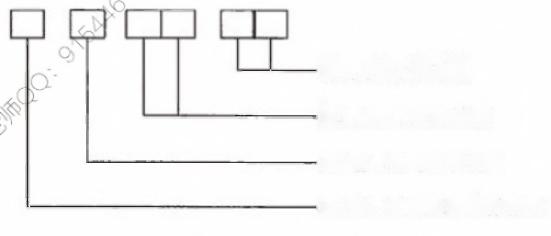
—137—

|  |  |
| --- | --- |
| **正式文档** **(4级文档)** | 适合那些要正式发行供普遍使用的软件产品 |

**4、信息系统项目相关信息(文档)管理的规则和方法：** **(12上64)(17下61)(19上61)**

**(1)文档书写规范：:** 应该遵循统一 的书写规范，包括符号的使用、图标的含义、程序中注释行的使用、 注明文档书写人及书写日期等；( **1** **8** **上** **6** **1** **)**

**(2)图表编号规则：** 对这些图表进行有规则的编号，可以方便图表的查找；( 1 0 上 6 0 )



第5、6位，流水码

第3、4位，文档内容

第2位，各阶段的文档

第1位，生命周期法各阶段

图15- 1 图表编号规则

**(3)文档目录编写标准；**

**(4)文档管理制度：** 应该建立相应的文档管理制度

**考** **点** **1** **2** **4** **:** **配** **置** **管** **理** **活** **动** **、** **配** **置** **项**

**1** **、软件配置管理涵盖了项目的整个生命周期。(** **1** **4** **上** **5** **5)**

2 、**配置管理包括6个主要活动：制定配置管理计划、配置标识、配置控制、配置状态报告、配置审计**、 **发布管理和交付。**

**3、** **典型配置项包括** **项目计划书、需求文档、设计文档、源代码、可执行代码、测试用例、运行软件所**

需的各种数据，它们经评审和检查通过后进入软件配置管理；(11上61)

**注意点：** **测试报告、会议纪要、工作记录不计入配置项的内容；因为一经形成就不好修改了!**

4、所有配置项的操作权限由配置管理人员(CMO) 严格管理，基本原则是：基线配置项向软件开发人员开放

**读权限；非基线配置项向PM,变更控制委员会(CCB)及相关人员开放。(** **1** **6** **上** **6** **4** **)**

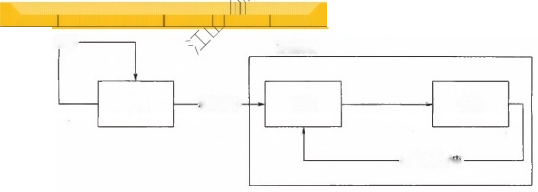
**5、** **基线配置项可能包括所有的设计文档和源程序等** **；非基线配置项可能包括项目的各类计划和报告等；**

—138—



**考点125:配置状态、版本管理**

1 、**配置项的状态可分** **为“草稿”、“正式”和“修改”三**种。配置项刚建立时，其状态为“草稿”。配置项通过 评审后，其状态变为“正式”。此后若更改配置项，则其状态变为“修改”。当配置项修改完毕并重新通过评审 时，其状态又变为“正式”;( 1 1 下 6 6 ) ( 1 2 上 6 6 ) ( 1 4 上 6 5 ) ( 1 4 下 6 5 ) ( 1 5 上 6 5 ) ( 1 5 下 6 5 ) ( 1 6 上 6 5 )



( 1 6 下 6 2 ) ( 1 7 下 6 2 ) ( 1 9 下 6 0 )

一

正式

再次通过评审-

-通过评审-

变更控制

修改

草稿

图15-2 配置项状态变化

① 处于“草稿”状 态的配置项的版本号格式为0 .YZ,YZ 的数字范围为01～99。随着草稿的修正， Yz 的 取 值应递增。 **Yz的初值和增幅由用户自己把握。**

② 处于“正式”状 态的配置项的版本号格式为X . Y,X 为主版本号，取值范围为1～9。 Y 为次版本号，取值 范围为0～9。

配置项第一次成为“正式”文件时，版本号为1.0。

③ 处于“修改”状 态的配置项的版本号格式为X .YZ。配置项正在修改时， 一 般 只 增 大Z 值，X.Y值保持不变。 当配置项修改完毕，状态成为“正式”时，将Z 值 设 置 为 0 , 增 加X.Y 值 。

2、 配置项的版本控制作用于多个配置管理活动之中，如创建配置项、配置项的变更和配置项的评审等。 在项目开发过程中， 绝大部分的配置项都要经过多次的修改才能最终确定下来。对配置项的任何修改都将产 生新的版本。由于我们不能保证新版本 一 定比旧版本“好”,所以不能抛弃旧版本。版本控制的目的是按照 一 定的规则保存配置项的所有版本，避免发生版本丢失或混淆等现象，并且可以快速准确地查找到配置项的

任何版本。 ( 1 3 上 6 5 ) ( 1 3 下 6 5 )



**考** **点** **1** **2** **6** **:** **基** **线**

1、 基线：由一组配置项组成，这些配置项构成一个相对稳定的逻辑实体。 基线中的配置项被“冻结” **了，不能再被任何人随意修改。对基线的变更必须遵循正式的变更控制程序**。

**在建立基线以前，工作产品所有者能快速、非正式的对产品做出变更。在基线建立后，变更要通过评价** **和验证变更的正式程序来控制。**

**2** **、**基线由一组配置项组成，这些配置项构成了一个相对稳定的逻辑实体。基线中的配置项被冻结了， 不能再被任何人随意修改。 基线通常对应于开发过程中的里程碑， 一个产品可以有多个基线，也可以只有一 个基线。通常将总价值给客户的基线称为一个发 行基线“Release”,为内部开发用的基线则称为一个构 造 基 线 “Build” 。配置项是可以更改的，在项目开发过程中， **绝大部分的配置项都要经过多次的修改才能最终确** 定下来，只是需要做好版本控制管理。( 1 5 下 6 3 )

3、一组拥有唯一标识号的需求、设计、源代码文卷以及相应的可执行代码、构造文卷和用户文档构成一 条基线。 **产品的一个测试版本** (可能包括需求分析说明书、 .概要设计说明书、详细设计说明书、已编译的可 执行代码、测试大纲、测试用例、使用手册等) **是基线的一个例子。**

4 、对于每一个基线，要定义下列内容： **建立基线的事件、受控的配置项、建立和变更基线的程序、批** **准变更基线所需的权限。** 在项目实施过程中，每个基线都要纳入配置控制，对这些基线的更新只能采用正式 的变更控制程序。



**考点127:配置库**

1 、 配 置 库 可 以 分 为开 发 库 、 受 控 库 、 产 品 库3 种 类 型 。( 1 7 上 6 2 - 6 3 ) ( 1 8 下 6 2 )

**(1)** **开** **发** **库** **(DevelopmentLibrary),也** **称** **为动** **态** **库** **、** **程** **序** **员** **库** **或** **工** **作** **库，用** **于** **保** **存** **开** **发** **人** **员** **当** **前** **正** **在** **开**

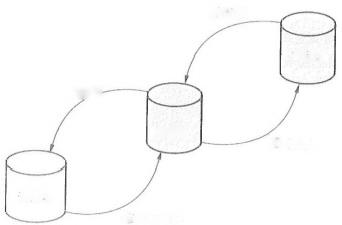
**发** **的** **配** **置** **实** **体，**动 态 虫 的 配 置 项 被 置 于 版 本 管 理 之 下 。 动 态 库 是 开 发 人 员 的 个 人 工 作 区 ， 由 开 发 人 员 自 行 控 制 。 库 中 的 信 息 可 能 有 较 为 频 繁 的 修 改 。( **1** **9** **上** **6** **2** **)**

(2) 受 控 库 (ControlledLibrary), 也 称 为主 库， 包 含当 前 的 基 线 加 上 对 基 线 的 变 更。 受 控 库 中 的 配 置 项 被置于完全的配置管理之下 。 在信息系统开发的某个阶段工作结柬时 ， 将当前的工作产品存入受控库 。

(3) 产 品 库 (ProductLibrary), 也 称 为静 态 库 、 发 行 库 、 软 件 仓 库， 包 含已 发 布 使 用 的 各 种 基 线 的 存 档 ，

**被** **置** **于** **完** **全** **的** **配** **置** **管** **理** **之** **下。** **在** **开** **发** **的** **信** **息** **系** **统** **产** **品** **完** **成系** **统** **测** **试** **之** **后** **，** **作** **为** **最** **终** **产** **品** **存** **入** **产** **品** **库** **内** **，**

**等** **待** **交** **付** **用** **户** **或** **现** **场** **安** **装。(** **1** **6** **下** **6** **6** **)**



③复制

产品库

② Check ou

④ 更新

开发库

③ Cheek in

受控库

图15-3 基于配置库的变更控制

**基于配置库的变更控制流程：** ①将待升级的基线从产品库中取出， 放入受控库；②程序员将欲修改的代码段从受控库中检出( Check out),放入自己的开发库中进行修改。代码被 Check out 后即被 “锁定”,以保证同一段代码只能同时被一个程序员修改，如果甲 正对其修改，乙就无法**Check** **out;** ③程序员将开发库中修改好的 代码段检入( Checkin) 受控库。 Check in 后，代码的“锁定” 被解除，其他程序员可以 Check out 该代码段了；④软件产品的 升级修改工作全部完成后，将受控库中的新基线存入产品库中。

2 、 **配** **置** **库的** **建** **库** **模** **式** **有** **两** **种** **：** **按** **配置** **项** **类** **型** **建** **库** **和** **按** **任** **务** **建**库 。

( 1 ) 按 配 置 项 的类 型 分 类建 库 ， 适 用 于通 用软 件 的 开 发 组 织 。 在 这 样 的 组 织 内 ， 产 品 的 继 承 性 往 往 较 强 ， 工具比较统一 ， 对并行开发有一定的需求 。 使用这样的库结构有利于对配置项的统一管理和控制 ，

同时也能提高编译和发布的效率 。 但由于这样的库结构并不是面向各个开发团队的开发任务的 ， 所以可 能 会 造 成 开 发 人 员 的 工 作 目 录 结 构 过 于 复 杂 ， 带 来 一 些 不 必 要 的 麻 烦 。( 1 8 上 6 2 )

( 2 ) 按开发 任 务 建 立相 应 的 配 置 库 ， 适 用 于专业软 件 的 开 发 组 织 。 在 这 样 的 组 织 内 ， 使 用 的 开 发 工 具 种类繁多 ， 开发模式以线性发展为主 ， 所以就没有必要把配置项严格地分类存储 ， 人为增加目录的复杂 性 。 对于研发性的软件组织来说 ， 采用这种设置策略比较灵活 。

**3** **、** **用** **于** **建** **立** **配** **置** **库** **的** **工** **具** **：VSS** **、CVS** **;** **也** **可** **以** **通** **过** **手** **工** **方** **式** **进** **行** **建** **库**；

**4、** **配** **置** **库** **的** **操** **作** **权** **限**

|  |  |
| --- | --- |
| 权 限 | 内 容 |
| Rcad | 可以读取文件内容，但不能对文件进行变更 |
| Check | 可使用[checkin]等命令，对文件内容进行变更 |
| Add | 可使用[文件追加],[文件重命名],[删除]等命令 |
| Destroy | 有权进行文件不可逆毁坏，清除，[rollback]等命令 |

**17、** **受** **控** **库** **的** **权** **限** **设** **置**

—141—

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 人员  权限 | | 项目经理 | 项目成员 | QA | 测试人员 | 配置管理员 |
| 文档 | Rcad | √ | √ | √ | √ | √ |
| Check | √ | √ | √ | √ | √ |
| Add | √ | √ | √ | √ | √ |
| Destroy | X | X | X | X | √ |
| 代码 | Rcad | √ | VO | √ | √ | √ |
| Check | √ | √ | X | X | √ |
| Add | √ | √ | X | X | √ |
| Destroy | X | X | X | X | √ |

**5、** **产品库的权限设置**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 人员  权限 | 项目经理 | 项目成员 | QA | 测试人员 | 配置管理员 |
| Rcad | √ | √ | √ | √ | √ |
| Check | √ | √ | √ | √ | √ |
| Add | X | X | X | X | √ |
| Destroy | X | X | X | X | √ |

—140—



**考** **点** **1** **2** **8** **:** **配** **置** **标** **识** **、** **配** **置** **管** **理** **计** **划** **、** **配** **置** **状** **态** **报** **告**

**1** **、配置(标识)识别是配置管理员职能，内容：(10上65)** **(**13上62、64)(15上64)(17上60)

(1)识别需要受控的配置项；

(2)为每个配置项指定唯一性的标识号；

(3)定义每个配置项的重要特征；

(4)确定每个配置项的所有者及其责任；

(5)确定配置项进入配置管理的时间和条件；

(6)建立和控制基线；

(7)维护文档和组件的修订与产品版本之间的关系。

2、配置管理计划的主要内容为：

(1)配置管理活动，覆盖的主要活动包括配置标识、配置控制、配置状态报告、配置审计、发布管理与交付；

(2)实施这些活动的规范和流程；

(3)实施这些活动的进度安排；

(4)负责实施这些活动的人员或组织，以及他们和其他组织的关系。

3 、配置状态报告也称配置状态统计，其任务是有效地记录和报告管理配置所需要的信息， 目的是**及时、**

**准确地给出配置项的当前状况，供相关人员了解，以加强配置管理工作。配置状态报告应着重反映当前基线** **配置项的状态，以向管理者报告系统开发活动的进展情况。**

4 **4** **、配置状态报告应该包含以下内容：**

(1)每个受控配置项的标识和状态。一旦配置项被置于配置控制下，就应该记录和保存它的每个后继 进展的版本和状态。

(2)每个变更申请的状态和已批准的修改的实施状态。

(3)每个基线的当前和过去版本的状态以及各版本的比较。

(4)其他配置管理过程活动的记录。



**考** **点** **1** **2** **9** **:** **配** **置** **审** **计**

4 **1** **、配置审计**的实施是为了确保项目配置管理的有效性，不允许出现任何混乱现象，例如：

(1)防止向用户提交不适合的产品，如交付了用户手册的不正确版本；

(2)发现不完善的实现，如开发出不符合初始规格说明或未按变更请求实施变更；

(3)找出各配置项间不匹配或不相容的现象；

(4)确认配置项己在所要求的质量控制审核之后纳入基线并入库保存；

(5)确认记录和文档保持着可追溯性。

4 **2** **、配置审计可分为功能性配置审计和物理配置审计：**

4 **3** **、功能配置审计是进行审计以验证以下几个方面：(** **1** **8** **下** **1** **4** **)**

(I) 配置项的开发已圆满完成

(2)配置项已达到规定的性能和功能特定特性

(3)配置项的运行和支持文档已完成并且是符合要求的

**4、** **物理配置审计是进行审计以验证如下方面：**

(1)每个构建的配置项符合相应的技术文档

(2)配置项与配置状态报告中的信息相对应

4 5 、项目经理组织修改相关的配置项，并在相应的文档或程序代码中记录变更信息。( 1 6 下 6 3 )

—143—

4 **6** **、** **发布管理与交付包含：存储、复制、打包、交付、重建。**( 1 9 下 6 1 )