# 7.3数据控制类别

软件生命周期的数据可以被分配到两个配置管理控制类别之一：控制类别1（CC1）和控制类别2（CC2）。表7-1定义了一套SCM进程有关的活动，每个控制类别表示一个最小的活动，适用于这一类的软件生命周期的数据。 CC2的活动是CC1活动的一个子集。

附件A表为软件生命周期的数据项指定控制类别通过软件级别。

表7-1 SCM进程有关的活动与CC1和CC2数据

|  |  |  |  |
| --- | --- | --- | --- |
| SCM进程活动 | 参考 | CC1 | CC2 |
| 配置鉴定 | 7.2.1 | ● | ● |
| 基线 | 7.2.2.a  7.2.2.b  7.2.2.c  7.2.2.d  7.2.2.e | ● |  |
| 可追溯性 | 7.2.2.f  7.2.2.g | ● | ● |
| 问题报告 | 7.2.3 | ● |  |
| 变更控制-完整性和鉴定 | 7.2.4.a  7.2.4.b | ● | ● |
| 变更控制-跟踪 | 7.2.4.c  7.2.4.d  7.2.4.e | ● |  |
| 变更审查 | 7.2.5 | ● |  |
| 配置状态统计 | 7.2.6 | ● |  |
| 恢复 | 7.2.7.a | ● | ● |
| 已批准变更的保护 | 7.2.7.b．1 | ● | ● |
| 媒体选择，刷新，复制 | 7.2.7.b．2  7.2.7.b．3  7.2.7.b．4  7.2.7.c | ● |  |
| 释放 | 7.2.7.d | ● |  |
| 数据保留 | 7.2.7.e | ● | ● |

# 11.0 软件生命周期数据

数据产生于软件生存期，用于计划、指导、解释、定义或提供活动证据。这些数据保证了软件生存期的开发，系统或设备的验证，快速最软件产品进行验证修改。这部分谈论软件生命周期数据的特性、形式、配置管理控制和内容。

1. 特性：软件生命周期数据应该：
2. 非歧义的：如果信息明确的只允许一个简单的解释，如果必要加一个辅助定义，则它是非歧义的。
3. 完整的：当信息包含必要及相关的要求和描述性材料时，则它是完整的；为有效输入数据定义响应；图表使用标记；条款和计量单位都有规定。
4. 可验证的：如果信息能被人或工具检验其正确性，则它是可验证的。
5. 一致的：如果信息在其内部没有冲突，则它是一致的。
6. 可更改的：如果信息是被构造的且有一个样式在保留原结构的同时可以保证改变的彻底性、一致性和正确性，则它是可更改的。
7. 可追踪的：如果信息组件的起源是可确定的，则它是可追踪的。
8. 形式：软件生命周期数据的形式应该在机载软件或装备的整个服务生命周期里提供软件生命周期数据的高效检索和审查。数据和数据的特定形式应该在软件方面验证计划书里指明。

注1：软件生命周期数据可能会存在多种形式(例如，电子或纸质形式)。

注2：申请人可以以任何他认为方便的方式打包软件生命周期数据项目(例如，作为私人数据项目或作为联合数据项目)。

注3：软件方面验证计划和软件技能概要可能会作为单独的文档被一些验证机构所需要。

注4：“数据”一词指的是证据和其他信息，并不指明这些数据应该采用的格式。

1. 配置管理控制：软件生命周期数据可以被放置在两类相关的应用软件配置管理控制：CC1和CC2。最低限度的控制类别分配给每个数据项，并在附件A中的表指明其软件层的变化。如果有额外的数据项而不仅仅此处描述的数据产生去作为证据辅助验证，那么为了最低限度，它们应该由CC2控制。
2. 内容：在下面的章节提供软件生命周期数据的描述，来确定数据是在软件生命周期中产生的。这个数据描述并不会对开发一个软件产品所需的所有数据进行描述，也不会指定一个特殊的数据打包方法或者在一个包内组织数据。目前提供的软件生命周期数据项内容的描述并不是无所不包的，应该结合文档内容和申请人的适应需求来阅读。

## 11.1 软件方面的验证计划

软件方面的验证计划是验证机构用来确定申请人是否提出一个软件生命周期的主要方法，这个软件生命周期伴随将被开发的软件层的苛刻要求。这个计划应该包括：

1. 系统概述：本节提供一个系统的概述，包括它的功能描述和它的硬件和软件配置、结构、处理器使用情况、硬件和软件的接口以及安全特性。
2. 软件概述：本节简要介绍软件的功能，强调提出的安全性和分区概念。并提供包括资源共享、冗余、容错、避免单事件翻转、定时和调度策略的例子。
3. 验证注意事项：本节提供一个验证基础的概述，包括遵守的方法，因为与软件方面的验证有关。本节也声明了软件层，总结了软件安全评估过程中提供的理由，包括潜在的软件失败的条件。
4. 软件生命周期：本节定义了要使用的软件生命周期，包括软件生命周期的流程的摘要，其细节信息定义在它们各自的软件计划里。摘要解释了将如何满足每个软件生命周期过程的目标，指明将要涉及的组织和组织的责任，以及系统生命周期的进程和验证联络过程的责任。
5. 软件生命周期数据：本节指定在软件生命周期进程中产生并控制的软件生命周期数据。本节也描述这些数据之间的关系或与定义系统的数据之间的关系，软件生命周期数据将提交给验证机构，数据的形式和数据的方法将对验证机构有效。
6. 调度：本节描述申请人使用的方法去给验证机构提供软件生命周期构成中活动的可视性，使可以计划复核。
7. 其他注意事项：本节描述可能影响验证过程的具体注意事项。包括遵守的替代方法、工具条件、以前开发的软件、选项可选软件、用户修改软件、无效代码、COTS软件、现场课转载软件、参数数据项、多版本的不同软件和产品服务历史方面的例子。
8. 供应商的监管：本节描述确保供应商的流程和产出复核批准的软件计划和标准的方法。

## 11.2软件开发计划

软件开发计划(SDP)是对软件开发过程的一个描述，被软件生命周期用来满足软件开发过程目标。它包含于软件方面验证的计划书里。这个计划包括：

1. 标准：确定项目的软件需求标准、软件设计标准和软件代码标准。此外，参考以前开发的软件，包括COTS软件，如果这些标准时不同的。
2. 软件生命周期：软件生命周期过程的描述被用来形成被项目使用被项目使用的特定的软件生命周期，包括软件开发过程中的过渡标准。这个描述有别于在软件方面验证计划书里提供的摘要，因为它提供必要的细节以确保软件生命周期过程的正确实施。
3. 软件开发环境：一个依据软件和硬件来选择软件开发环境的声明，包括：
4. 需求开发方法和使用的工具。
5. 设计方法和使用的工具。
6. 编码方法，编程语言，使用的编码工plan具和当可应用时自动编码产生器的选项和限制。
7. 编译器，连接编辑器和使用的转载机。
8. 所使用工具的硬件平台。

## 11.3软件验证计划

软件验证计划是一个用来满足软件验证过程目标的验证过程的描述。这些过程可能随着软件层次的不同而不同，正如附件A表中定义的那样。这个计划包括：

1. 组织：组织的责任存在于软件验证过程中和与其他软件生命周期过程的接口中。
2. 独立：当需要时，一个建立验证独立方法的描述。
3. 验证方法：一个用于软件验证过程中每个活动的验证方法的描述。
4. 审查方法，包括审查表和其它辅助项。
5. 分析方法，包括可追溯性和覆盖分析。
6. 测试方法，包括选择测试案例的方法，要使用的测试程序和将产生的测试数据。
7. 验证环境：一个设备的测试，测试和分析工具，以及如何使用这些工具和硬件测试设备的描述。4.4。4b节提供指导说明目标计算机和模拟器或仿真器的差别。
8. 过渡标准：进入软件验证过程的过渡标准。
9. 分区注意事项：如果使用分区，验证分区完整性使用的方法。
10. 编译器假设：一个关于申请人提出的关于编译器、连接编辑器或转载机正确性的假设的描述(见4.4.2)。
11. 重新验证方法：对于软件修改，一个关于识别、分析和验证软件受影响的部分与可执行代码的更改部分方法的描述。
12. 以前开发的软件：对于以前开发的软件，一个符合本文档目标的方法的描述。如果其验证过程的最初合规基线不符合本文。
13. 多版本不同软件：一个关于软件验证过程活动的描述，如果使用多版本不同软件。

## 11.4软件配置管理计划

软件配置管理计划建立用于达到整个软件生命周期中SCM过程目标的方法。这个计划包括：

1. 环境：一个关于SCM环境的描述，包括程序、工具、方法、标准、组织责任和接口。
2. 活动：一个关于软件生命周期中SCM过程活动的描述：
3. 配置标识：当它们将被确认时，软件生命周期数据的识别方法（例如，部分编码），以及软件识别和软件或装备识别的关系将被确认。
4. 基线和可追溯性：建立基线的方法，将建立什么样的基线，什么时候建立，建立软件库控制和配置项以及基线的可追溯性。
5. 问题报告：软件产品和软件生命周期过程的问题报告的内容和标识，当他们被写入，关闭问题报告的方法和改变控制活动的关系。
6. 改变控制：配置项和基线进行控制，当他们将被控制，控制他们的问题、变更控制活动，预验证控制，后验证控制，维护基线和配置项的完整性的方法。
7. 改变审查：处理反馈形式和软件生命周期过程的方法；评估和确定优先级次序，批准改变，并处理它们的决议或改变实现的方法；这些问题报告的方法和改变控制活动的关系。
8. 配置状态记录：记录数据是能报告配置管理的状态，定义数据在哪保存，如何将被检验报告，何时将可适用。
9. 存档，检索和释放：完整的控制，释放方法和权威，数据保留。
10. 软件加载控制：软件加载控制保障和记录的描述。
11. 软件生命周期环境控制：用于开发，建设，验证，和加载软件的工具控制，在11.4.b.1节到11.4.b.7节有讲述。这包括控制工具使其合格。
12. 软件生命周期数据控制：CC1和CC2的相关控制。
13. 过渡标准：进入SCM过程的过渡标准。
14. SCM数据：SCM过程中产生的软件生命周期数据的定义，包括SCM记录，软件配置索引和软件生命周期环境配置索引。
15. 供应商的控制：在供应商应用SCM过程需求的方法。

## 11.5 软件质量保证计划

软件质量保证计划建立用来达到SQA过程目标的方法。SQA计划可能包括过程改进的说明和渐进式管理方法。这个计划应该包括：

1. 环境：SQA的环境描述，包括范围，组织责任和接口，标准，程序，工具和方法。
2. 权威：SQA权威，责任和独立的声明，包括软件产品的SQA批准。
3. 活动：这个软件生命周期中为每个软件生命周期过程执行的SQA活动包括：
4. SQA方法，例如，软件生命周期过程中的评论，审计，报告和监测。
5. 有关问题报告，跟踪和纠正措施系统的活动。
6. 软件符合性审查活动的说明。
7. 过渡标准：进入SQA过程的过渡标准。
8. 定时：SQA过程活动的定时涉及软件生命周期过程的活动。
9. SQA记录：SQA过程产生的记录的定义。
10. 供应商的监督：确保供应商的过程和输出会遵照计划和标准的方法的说明。

## 11.6软件需求标准

软件需求标准定义了开发上层需求的方法，规则和工具。这些标准包括：

1. 用来开发软件需求的方法，例如结构方法。
2. 用来表述需求的符号，例如数据流图和正规语言。
3. 约束需求开发过程中多使用的工具的使用。
4. 提供系统进程中派生需求的方法。

## 11.7 软件设计标准

软件设计标准定义了用来开发软件结构和低层需求的的方法，规则和工具。这些标准包括：

1. 使用的设计描述方法
2. 使用的命名规范
3. 允许设计方法上的附加条件，例如，调度，中断和事件驱动架构的使用，动态任务，重新进入，全局数据，异常处理和他们所使用的原理。
4. 设计工具的使用限制。
5. 设计上的限制，例如，排除递归，动态对象，数据别名，致密表达。
6. 复杂性限制，例如，最高层的嵌套调用或有条件的结构，使用无条件分支，代码组件的入口/出口点的数量。

## 11.8 软件代码标准

软件代码标准定义了用来编写软件的编程语言，方法，规则和工具。这些标准包括：

1. 要使用和/或定义子集的编程语言。对于一个编程语言，引用明确定义语法的数据，控制行为，数据行为和语言的副作用。这可能需要限制使用的一些语言特色。
2. 源代码演示标准，例如，线路长度限制、缩进，空白行的使用和源代码文件标准，例如，作者姓名，修订历史记录，输入输出，以及受影响的全局数据。
3. 组件、子程序、变量和常量的命名约定。
4. 强加在允许编码约定上的条件和制约因素，如软件组件和逻辑或数值表达式的复杂性，以及其使用的理由的耦合度。
5. 编码工具的使用限制。

## 11.9软件需求数据

软件需求数据是高层需求包括派生需求的一个定义。数据包括：

1. 给软件分配系统需求，并注意与安全有关的需求和潜在故障条件的说明。
2. 每个操作模式下的功能和操作要求。
3. 性能标准，例如，精确度和准确度。
4. 计时要求和约束
5. 内存大小限制。
6. 硬件和软件接口，例如，协议、格式、输入频率和输出频率。
7. 故障检测和安全监测要求。
8. 分区分配到软件的要求，分区的软件组件之间如何交互，与软件层的每个分区如何交互。

## 11.10设计说明

设计说明是软件体系结构和满足高层要求的低层要求的一个定义。数据包括：

1. 详细描述软件如何满足指定的高层的要求，包括算法，数据结构，如何给处理器和任务分配软件需求。
2. 定义实现需求的软件结构的软件体系结构描述。
3. 输入/输出说明，例如，整个体系结构内部和外部的数据字典。
4. 数据量和控制设计。
5. 资源限制，管理每个资源的策略及其局限性，利润率，衡量这些利率的方法。例如，时间和内存。
6. 调度程序和处理器间/任务间通信，包括硬时间排序，抢占调度，Ada交会和中断。
7. 实现的设计方法和细节，例如，软件装载，用户可修改软件，或多版本不同软件。
8. 防止分区违规的分区方法和手段。
9. 软件组件说明，它们是新的还是以前设计的，如果是以前设计的，参考它们所采取的基准。
10. 软件设计过程中派生的需求。
11. 如果系统包括失效代码，则需要一个关于确保这个代码不在目标计算机上启用的方法的说明。
12. 这些设计决定的基本原理可追溯到与安全有关的系统要求。

## 11.11 源代码

此数据包含用源语言编写的代码。源代码用于编译，连接，以及在集成过程中加载数据区开发集成系统或设备。对于每一个源代码组件，此数据需包含软件标识，包括名称和日期的修订版本和/或适用的版本。

## 11.12可执行目标代码

可执行目标代码包含在目标计算机处理单元直接可用的代码，因此是加载到硬件或系统上的条件。

## 11.13软件验证例子和程序

软件验证例子和程序详细说明软件验证进程活动是如何实现的。次数据应包括的说明：

1. 审查和分析程序：使用除了软件验证计划中说明的广度和深度审查分析方法。
2. 测试例子：每个测试例子，输入，条件，预期结果，以实现所需覆盖范围的准则和通过/失败标准的目的。
3. 测试程序：逐步指导每个测试用例是如何建立和执行的，如何评价测试结果，以及测试环境的使用。

## 11.14软件验证结果

软件验证结果是软件验证进程活动产生的。软件验证结果应该：

1. 为每个审查，分析，测试，指明每个程序在活动中是通过或者失败，以及最终通过/失败的结果。
2. 识别配置项或软件版本审查，分析或测试。
3. 包括测试、审查和分析结果，包括覆盖分析和可追溯性分析。

任何发现的异常都应该记录下来并通过问题报告跟踪。

此外，为软件进程(见2.2.1.f和2.2.2.1.g)产生的软件进程评估信息提供支持的证据应该被认为是软件验证结果。

## 11.15软件生命周期环境配置索引

软件生命周期环境配置索引标识软件生命周期环境的配置。此索引帮助硬件的再生产和软件生命周期环境的重建，复检或软件修改，应该：

1. 确定软件生命周期环境的硬件和其操作系统软件。
2. 确定软件开发过程中使用的工具。示例包括编译器，连接编辑，装载机，数据完整性工具，例如计算并嵌入校验和或循环冗余检查的工具，以及任何与其关联选项的自动代码产生器。
3. 确定用来验证软件产品的测试环境，如软件测试和分析工具。
4. 确定合格的工具和其关联的工具资格数据。

*备注：此数据可能包含软件配置索引。*

## 11.16软件配置索引

软件配置索引(SCI)标识软件产品的配置。应提供特定的配置标识符合版本标识符。

*备注：SCI可以包含一个数据项或一组数据项。SCI可以包含下面列出的项目，或者引用另一个SCI，或其他指定的单个项目和它们版本的配置标识数据。*

SCI应该标识：

1. 软件产品
2. 可执行目标代码和参数数据项文件，如果有的话。
3. 源代码组件
4. 软件产品中以前开发的软件，如果使用了。
5. 软件生命周期数据。
6. 存档和释放媒体。
7. 建立可执行目标代码和参数数据项目文件的说明，如果有的话，例如包括指令和编译连接的数据；用于恢复软件进行再生，测试或修改的程序。
8. 引用软件生命周期环境配置索引，如果它是单独包装的。
9. 可执行代码的数据完整性检查，如果使用了。
10. 程序，方法，和为用户可修改软件修改的工具，如果有的话。
11. 加载软件到目标硬件的程序和方法。

*备注：SCI可能产生一个软件产品版本，或它可能会扩充包含几个替代物或连续的软件产品版本。*

## 11.17问题报告

问题报告是一种方法，它识别并记录软件产品异常行为的解决法案 ，不符合软件计划和标准的过程，以及软件生命周期数据的缺陷。问题报告包括：

1. 配置项和/或有问题的软件生命周期进程活动的标识。
2. 将被修改的配置项标识或将被改变的进程说明。
3. 能够理解并解决问题的问题说明。问题描述应包含足够多的细节，以便潜在安全评估或问题的功能影响。
4. 为解决报告的问题采取的纠正操作的说明。

## 11.18软件配置管理记录

SCM进程活动的结果记录在软件配置管理记录里。示例包括配置标识表，基准或软件库记录，更改的历史记录，归档记录，版本记录。这些示例并不意味着一定要产生这些特定类型的记录。

*备注：由于SCM进程的完整性质，其产出将往往作为其他软件生命周期数据的一部分。*

## 11.19软件质量保证（SQA）记录

软件质量保证SQA处理活动过程的结果记录在SQA记录上。记录的项目一般有：SQA评审或者是审计报告、会议纪要、授权过程偏差记录、软件确认评审记录。

## 11.20软件完成综述

软件完成综述是一项为了展示软件保证计划符合性的主要数据条目。综述一般包括：

1. 系统概述：本节负责提供系统的全貌，包括它的功能描述以及软件硬件的分配情况、体系结构、处理器的使用、软件硬件的接口、安全性。此部分也描述了软件保证计划和系统概述的不同点。
2. 软件概述：本节简要描述了强调安全和划分概念的使用的软件功能，同时也解释了在软件保证书计划里面的软件概述的不同点。
3. 验证注意事项：此部分重述软件方面验证计划里描述的验证注意事项并描述一些差异。
4. 软件生命周期：本节概况性描述实际软件生命周期和解释软件生命周期以及软件生命周期处理计划在软件保证书计划中的不同点
5. 软件生命周期数据：本节描述了软件方面验证计划中制定的任何计划的不同点，为了软件生命周期数据的处理和数据之间的相互关系以及对于其他数据定义的系统，通过保证书授权的可用数据的手段。此部分通过标识符和版本的配置明确提及软件配置索引和软件生命周期环境配置索引。关于配置软件生命周期数据中标识符和特定版本的详细信息都在软件配置索引里面有说明。
6. 其他注意事项：本节描述可能在验证机构的认证方面的特定注意事项。并且说明软件方面验证计划所包含的规划的不同点以及这一方面的考虑。参考应该有数据条目可用，例如出版的论文或者特殊条件。
7. 供应商监管：本节描述供应商怎样处理和输出来服从计划以及标准。
8. 软件标识：本节通过部分号码和版本来标识软件配置。
9. 软件特征：本节描述可执行对象代码的大小规格、时间容限，包括执行时间的最坏的情况，存储容限，资源容限和其他的特征测量手段。
10. 更改历史：如果可用，本节包括软件更改概述，注意这类更改带来的错误会影响到安全，并且确定从以前的保证书开始确认所有的更改和改善的软件生命周期的过程。
11. 软件状态：本节包含了在保证时间里未解决的出错报告的概述。出错报告概述包括每一个问题以及相关的错误的描述、功能限制、操作限制、潜在的负面的安全影响，保证这些就必须让出错报告一直打开，并详述所需要进行的减轻动作。
12. 一致性声明：本节包含一致性的阐述文档和用在证明一致性软件计划上的特定标准方法的总结。此部分也阐述了验证机构和任何软件计划偏差、标准的额外规则，并且此说明文档不包含其他的软件完成概述。

## 11.21数据追踪

数据追踪确立生命周期数据条目内容的相互联合。数据追踪应该提供给证明双向性的联合，双向性联合包含：

1. 系统需求分配给软件和高级需求。
2. 高层需求和低层需求。
3. 低层需求和源代码。
4. 软件需求和测试实例。
5. 测试实例和测试过程。
6. 测试过程和测试结果。

## 11.22参数数据项文件

参数数据项文件由一个数据形式构成，这个数据形式通过处理目标节点单元的直接可用。软件生命周期数据应该为每一个参数数据项目提供实例化。如果隔离封装，此数据应该包含 到可执行的对象代码的软件完成总结的参考。

UML,

SCM进程活动的结果记录在软件配置管理记录里。示例包括配置标识表，基准或软件库记录，更改的历史记录，归档记录，版本记录。这些示例并不意味着一定要产生这些特定类型的记录。

*备注：由于SCM进程的完整性质，其产出将往往作为其他软件生命周期数据的一部分。*

* 建立配置管理模型的考虑

不同的软件由于其不同的特点，在拟定配置管理计划时会有很大差异。

由于软件规模的不同，也不可能采用同样的配置管理方式

而且目前业界不存在 统一的配置管理系统

但是由于只要属于 适航软件，就必须遵守DO178C的标准，

在研究Do178c的基础上，建立符合Do178c标准的配置管理模型。

这个模型的作用不是进行配置管理过程，由于以上原因。它的作用

* 1. 进行适航软件开发的配置管理人员能够依照此配置管理计划为原型建立符合目标要求的配置管理计划。
  2. 对已生成的软件生命周期数据与软件配置管理计划进行检查。可以用于审定过程，也可以用于开发过程期间的检查等。

难点：软件生命周期数据的接口。

* 通过 模型 对配置管理过程目标、审查单项 之间的对应关系。

配置管理目标——某些部分不能用机器证明

* 辅助工具的实现

对已生成的数据进行检验的方法：代入Do-178c的基础系统进行检验。

Q1:生命周期数据输入的一致性。

Q2:从哪边的改变作为状态改变的驱动：多方面。提交申请等等。

适航配置管理过程有以下特点：

1. 软件生命周期数据。有一部分已规定好。软件合格审定计划中对软件生命周期和配置项进行了说明

问题：现阶段配置管理不存在同意的系统，怎样保证软件生命周期数据（包括配置管理记录，变更申请等的一致性）

开发方要符合目标就要提供证据：1.必须有这方面的证据。2.即使不能直接使用，也应该是易于处理的格式，便于加工 。3在此模型中要将一些配置管理记录元素化， 如变更单上必须有变更申请时间，申请人，审批人等。

问题：为什么不直接使用研发方的配置管理系统进行配置管理记录的审查。

研发方的配置管理系统可能是不符合D0178c标准的。

1.假设符合标准——>2.必须有格式化的证据。——>3.对能够进行机器验证的部分进行验证

虽然还没有对开发期间配置管理过程做出明确要求，但要满足Do178c的标准，使用手工进行配置管理是不可能的。