ICS 03.220  
CCS G563

团体标准

T/CECC 000—2025

机载软件适航要求符合性规范

Compliance Specifications for Airworthiness Requirements of Software in Airborne

（征求意见稿）  
2025-xx-xx

2025-xx-xx 发布

2025-xx-xx 实施

|  |  |
| --- | --- |
| 中国电子商会 | 发布 |

目  次

[前  言 IV](#_Toc1909)

[1 范围 5](#_Toc16914)

[2 规范性引用文件 5](#_Toc10562)

[3 术语和定义 5](#_Toc12749)

[3.1 变更控制 Change control 5](#_Toc5193)

[3.2 商用现成工具 Cots tool 5](#_Toc905)

[3.3 配置管理 Configuration management 5](#_Toc26376)

[3.4 功能开发保证级别 FDAL 6](#_Toc5455)

[3.5 项目开发保证级别 IDAL 6](#_Toc20871)

[3.6 过渡标准 Transition criteria 6](#_Toc5047)

[4 缩略语 6](#_Toc11188)

[5 机载软件系统需求 6](#_Toc27808)

[5.1 系统功能识别与分解原则 6](#_Toc23737)

[5.1.1 系统功能和需求的开发 6](#_Toc4866)

[5.1.2 系统需求分解数据 7](#_Toc16085)

[5.2 系统架构匹配机制与系统需求分配 7](#_Toc26625)

[5.2.1 概述 7](#_Toc23680)

[5.2.2 系统架构开发和系统需求分配到项目 7](#_Toc3374)

[5.3 安全目标与开发保障等级分配要求 7](#_Toc30229)

[5.3.1 初步系统安全性评估 7](#_Toc6720)

[5.3.2 开发保障等级分配 8](#_Toc25104)

[5.4 系统级需求验证方法与追踪性控制 9](#_Toc15065)

[5.4.1 验证方法 9](#_Toc30780)

[5.4.2 验证计划 9](#_Toc8075)

[5.4.3 验证跟踪 10](#_Toc10028)

[5.5 软件需求完整性要求 10](#_Toc975)

[5.5.1 概述 10](#_Toc3774)

[5.5.2 软件高层需求要求 10](#_Toc22812)

[5.5.3 软件低层需求要求 10](#_Toc3959)

[6 机载软件开发与验证 11](#_Toc26422)

[6.1 软件生命周期阶段划分规范 11](#_Toc32247)

[6.1.1 概述 11](#_Toc29802)

[6.1.2 生命周期的阶段划分 11](#_Toc25305)

[6.2 软件设计与编码规则 11](#_Toc23651)

[6.2.1 概述 11](#_Toc12981)

[6.2.2 软件设计阶段规则 11](#_Toc29345)

[6.2.3 软件编码阶段规则 11](#_Toc1828)

[6.3 软件验证目标 11](#_Toc13810)

[6.4 软件配置管理与质量保证目标 12](#_Toc26871)

[6.4.1 概述 12](#_Toc17488)

[6.4.2 软件配置管理目标 12](#_Toc31992)

[6.4.3 软件质量保证目标 12](#_Toc15125)

[6.5 软件生命周期文档清单及交付要求 12](#_Toc12276)

[6.5.1 概述 12](#_Toc19810)

[6.5.2 软件生命周期文档清单 12](#_Toc9188)

[6.5.3 软件生命周期数据交付要求 13](#_Toc27436)

[6.6 适航符合性确认流程与核查依据 13](#_Toc28511)

[6.6.1 概述 13](#_Toc25936)

[6.6.2 认证联络流程 14](#_Toc2509)

[6.6.3 合规性证明 14](#_Toc8086)

[6.6.4 认证流程 14](#_Toc20970)

[7 机载软件工具合规性 14](#_Toc10133)

[7.1 工具分类与资格等级判定方法 14](#_Toc18836)

[7.1.1 工具分类 14](#_Toc32711)

[7.1.2  TQL等级 15](#_Toc14601)

[7.2 工具运行集成过程与环境安装要求 15](#_Toc10337)

[7.2.1 工具运行集成过程 15](#_Toc15603)

[7.2.2 工具运行集成过程目标 15](#_Toc31643)

[7.2.3 工具运行集成活动 15](#_Toc7672)

[7.3 工具运行验证与确认过程 15](#_Toc24977)

[7.3.1 工具运行验证和确认过程目标 15](#_Toc1448)

[7.3.2 工具运行验证和确认过程活动 16](#_Toc16362)

[7.4 商用工具资格确认方法 16](#_Toc1023)

[7.4.1 合格的COTS工具 16](#_Toc27615)

[7.4.2 COTS工具鉴定方法 16](#_Toc12758)

[7.4.3 工具开发人员的鉴定活动 17](#_Toc22877)

[7.4.4 工具用户的工具鉴定活动 17](#_Toc24565)

[7.4.5 开发人员和用户活动的协调 18](#_Toc22614)

[7.5 工具配置管理 18](#_Toc32680)

[7.5.1 工具配置管理过程目标 18](#_Toc31104)

[7.5.2 工具配置管理过程活动 18](#_Toc6403)

[7.6 工具合规性数据 19](#_Toc15882)

[7.6.1 工具资格认证流程和其他整体流程数据 19](#_Toc1079)

[7.6.2 工具开发过程和相应验证活动期间产生的工具确认数据 23](#_Toc4632)

[7.6.3 工具操作要求过程、工具操作集成过程以及相应验证和确认活动期间生成的工具鉴定数据 24](#_Toc1706)

[参 考 文 献 26](#_Toc20485)

前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中国电子商会提出并归口 。

本文件由西北工业大学和中国电子商会信息工程测试专业委员会发起。

本文件起草单位：

本文件主要起草人：

机载软件适航要求符合性规范

1. 范围

本文件规定了机载软件的需求标准、开发与验证标准、以及工具合规性标准。

本文件适用于规范机载软件的需求分析、设计开发、工具使用与验证流程，评估机载软件的功能完整性、安全性与适航符合性。

1. 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件,仅注日期的版本适用于本文件。凡是不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

SAE ARP4754B-2023 民用飞机和系统开发指南

RTCA DO-178C 机载系统和设备软件的合格审查考虑

RTCA DO-330 软件工具鉴定注意事项

1. 术语和定义

MH/T 1039-2011、GB/T 25069-2022、GB/T 18041-2000界定的以及术语和定义适用于本文件。



**变更控制 Change control**

(1)在正式确定配置项的配置标识或正式确定基线后，记录、评估、批准或否决以及协调对配置项的变更的过程。(2)在正式确定配置项的配置标识或正式确定基线后，系统地评估、协调、批准或否决并实施已批准的配置项变更的过程。



**商用现成工具 Cots tool**

供应商通过公共目录列表提供的商用工具。为特定软件项目新开发的、经过合同谈判的工具不是COTS工具。



**配置管理 Configuration management**

(1)这一过程包含以下步骤：a.确定并定义系统配置项；b.在软件生命周期中管控这些配置项的发布与变更；c.记录并上报配置项状态及问题报告；d.核验配置项的完整性和准确性。(2)该规范通过技术指导与行政监督，对以下环节实施管理：a.确定并记录配置项的功能与物理特性；b.控制这些特性的变更；c.记录并汇报变更控制流程的执行状态。



**功能开发保证级别 Fdal**

对功能执行的开发保证任务的严格程度。



**项目开发保证级别 Idal**

对项目执行的开发保证任务的严格程度；例如，IDAL 是 DO-178C/ED-12C中的适当软件级别，以及DO-254/ED-80目标中需要满足的项目设计保证级别。



**过渡标准 Transition criteria**

软件规划过程中定义的、为进入流程而必须满足的最低条件。

1. 缩略语

下列缩略语适用于本文件。

COTS：商用现成产品(Commercial Off-The-Shelf)

FDAL：功能开发保证等级(Function Development Assurance Level)

IDAL：项目保证等级(Item Development Assurance Level)

PASA：飞机初步安全评估(Preliminary Aircraft Safety Assessment)

PSSA：系统初步安全评估(Preliminary System Safety Assessment)

TAS：工具完成情况总结(Tool Accomplishment Summary)

TCI：工具配置索引(Tool Configuration Index)

TCM：工具配置管理(Tool Configuration Management)

TOR：工具操作要求(Tool Operational Requirements)

TQL：工具资质等级(Tool Qualification Level)

TQA：工具质量保证(Tool Quality Assurance)

TQP：工具合格计划(Tool Qualification Plan)

1. 机载软件系统需求
   1. **系统功能识别与分解原则**
      1. **系统功能和需求的开发**

系统功能可以是系统的预期行为，例如系统的预期操作、防止不希望的功能行为，以及提供其他系统使用的输出。例如，飞机的电力供应是一个系统级别的功能，而设备内部的电源则不是系统级别的功能。

通过功能分配和相关的故障条件分类，可以确定实现安全目标所需的进一步具体系统要求。在这个阶段，随着各种功能组合和分配的影响被考虑，将产生衍生需求和额外假设。这些需求反过来可能会影响飞机级别的功能要求。

这项活动的成果是一套针对每个飞机系统的具体要求，包括相关接口，如安全要求和PASA过程中的开发保证水平。接口定义为每个输入都有一个来源，每个输出都有一个目的地。

* + 1. **系统需求分解数据**

系统需求的分解和分配是按照系统流程进行的。以下数据作为需求分解的一部分传递给项目流程：

1. 分配给软件项目的资源。
2. 项目的开发保证水平和相关故障条件的描述（如适用）。
3. 系统描述。
4. 设计约束，包括功能隔离、分离、其他外部接口元素的数据/模型、分区要求和任何项目开发独立性要求。
5. 在适用情况下，将在项目开发阶段执行系统验证活动。
6. 系统过程对项目过程提供的任何数据的可接受性证据，这些数据被用于系统过程进行的任何活动或评估；例如，系统过程评估项目过程提供的衍生需求，以确定是否存在更高层次的功能或安全影响。
   1. **系统架构匹配机制与系统需求分配**
      1. **概述**

软件设计标准应具有系统架构匹配机制与系统需求分配的相关要求。

* + 1. **系统架构开发和系统需求分配到项目**

在实际操作中，系统架构的开发与系统需求分配到各个项目之间紧密相连，形成一个迭代的过程。需求分配过程中产生的需求可能与系统相关，也可能与项目相关。随着每个周期的推进，对衍生需求的理解和识别逐渐加深，系统级需求分配到项目中的理由也更加清晰。当所有需求都能在最终架构中得到满足时，整个过程就完成了。系统架构的开发和需求分配确保了项目能够实现所有预期的系统功能。

系统架构定义了特定项目设计的结构和边界，以满足既定的需求。在实施前，可能会考虑多个候选系统架构。

这些候选架构可通过以下因素进行评估：

1. 技术准备度。
2. 实施时间表。
3. 可生产性。
4. 合同义务。
5. 经济性。
6. 先前经验。
7. 行业先例。

随后，通过功能和性能分析对候选架构进行迭代评估。特别是，PASA和PSSA过程用于评估每个系统分配的顶级安全目标的可行性。

随着系统架构工作的进展，技术、体系结构、系统、设备和项目接口、系统约束（物理、环境等）、集成和实施选择所要求的内容会更加清晰。

该分配工作的输出是一个系统架构，其中要求分配到项目中，包括来自PASA/PSSA过程的安全要求和开发保证等级。

* 1. **安全目标与开发保障等级分配要求**
     1. **初步系统安全性评估**

初步系统安全性评估（PSSA）过程是对提议的系统架构进行系统评估，旨在确定故障、错误或外部事件如何导致系统级别的安全失效条件，并探讨如何实现系统级别的安全目标。PSSA过程的目标是根据PASA的安全目标和分配的要求，评估提议的系统架构，并提出系统级别的安全要求和假设。

通过PSSA过程，确定系统、子系统和项目的安全要求，以指导架构开发，确保满足安全目标和要求。PSSA过程可能采用多种安全分析方法来确定这些要求。此外，PSSA还确定了系统功能和项目所需的FDAL和IDAL分配。

PSSA过程确定了可能需要哪些保护策略来满足安全目标。这些保护策略可能包括冗余、监控、分区、开发保证严格性、内置测试和与安全相关的维护任务/间隔。

PSSA过程是互动的，并与设计定义紧密相关。就像开发过程是迭代的，PSSA过程也是迭代的。PSSA过程从设计初期开始，通过评估系统架构来识别并提出系统级的安全需求。随后，这些系统级的安全需求会被分配到各个子系统，最终子系统的具体要求会被细化到具体的设备或项目上。PSSA对IDALs（独立性、可验证性、可用性、可维护性、可测试性）的分配决定了硬件和软件开发保证的严格程度。此外，PSSA还会生成项目需求，包括但不限于安全性、可靠性、独立性和分离性。在PSSA过程中，会考虑常见的原因因素。同时，需要特别注意潜在故障及其暴露时间。一旦通过PSSA过程确定了系统级的安全需求，且实施的子系统/设备/项目的详细设计已经成熟，就可以启动系统级安全分析过程。

* + 1. **开发保障等级分配**

开发错误通过建立开发保障过程来解决。该过程旨在确保飞机和系统的开发过程以足够严谨的方式进行，从而减少可能影响飞机安全的开发错误。开发保障水平反映了对开发过程的严格程度，以确保在飞机和系统功能及项目开发过程中，如果这些项目在服役中暴露，可能会对安全产生不利影响的错误发生概率被控制在可接受的安全水平。

飞机/系统功能或项目的发展保障水平不仅适用于该飞机/系统功能或项目的开发过程，而且适用于与可能影响正在检查的功能或项目的其他飞机/系统功能或项目相互关联的所有其他飞机/系统功能或项目的接口的开发。

开发保障等级的分配取决于故障条件的严重程度分类，同时考虑到开发过程之间可能存在的独立性，这可以限制开发错误的后果。故障条件分类越严重，为将开发错误的可能性限制在可接受的置信水平所需的开发保证等级就越高。

考虑到最高级故障条件严重性分类，开发保证等级分配的一般原则如下：

当涉及灾难性故障条件时，分配原则为：

1. 如果灾难性故障条件可能由飞机/系统功能或部件的潜在开发错误导致，则相关开发保证过程被指定为A级。
2. 如果灾难性故障可能由两个或多个独立的飞机/系统功能或部件之间的开发错误组合导致，那么任一开发保证过程应被指定为A级，或者两个开发保证过程至少被指定为B级。其他独立的飞机/系统功能或部件应被指定不低于C级的开发保证。确认两个或多个独立的飞机/系统功能或部件实际上独立的开发保证过程应保持A级。

当涉及危险故障条件时，分配原则为：

1. 如果危险故障条件可能是由于飞机/系统功能或部件的可能开发错误导致的，则相关开发保证过程至少被指定为B级。
2. 如果两个或多个独立的飞机/系统功能或部件之间的可能开发错误组合可能导致危险故障条件，那么至少一个开发保证过程应被指定为B级，或者两个开发保证过程应被指定为C级。其他独立的飞机/系统功能或部件应被指定为不低于D级的开发保证。用于确认两个或多个独立的飞机/系统功能或部件实际上独立的开发保证过程应保持B级。

当涉及主要故障条件时，分配原则如下：

1. 如果主要故障可能是由于飞机/系统功能或部件的可能开发错误导致，那么相关开发保证过程至少被指定为C级。
2. 如果一个主要故障可能是由于两个或多个独立的飞机/系统功能或部件之间的潜在开发错误导致，那么至少应将一个开发保证过程定为C级，或者将两个开发保证过程定为D级。确保这些功能或部件确实独立的开发保证过程应保持在C级。

当涉及次要故障条件时，分配原则如下：

1. 如果一个次要故障可能是由于飞机/系统功能或部件的可能开发错误导致的，那么相关开发保证过程至少被指定为D级。
2. 如果一个次要故障是由两个或多个独立的飞机/系统功能或部件之间的可能开发错误组合导致的，则至少分配一个D级开发保证过程。

当涉及无安全性影响的故障条件时，分配原则为：

1. 如果一个或多个可能的飞机/系统功能或项目开发错误可能导致无安全性影响的故障条件，则相关开发保证过程可被指定为E级。
   1. **系统级需求验证方法与追踪性控制**
      1. **验证方法**

这些方法包括可追溯性、分析、建模、测试、相似性和工程审查。

每个需求及其需求集也应接受工程审查。虽然可追溯性和工程审查可能足以确保需求的完整性和正确性，但某些需求可能需要额外的验证方法来达到这一目标。对于FDAL A和B需求，当可追溯性或派生需求的需求理由，加上工程审查不足以确保正确性和完整性时，应采用额外的验证方法。以下提供了一些可能需要额外验证方法的需求类型示例：

1. 复杂状态机。
2. 安全监测器定义（例如阈值、确认时间、持续性）。
3. 性能和容差定义。
4. 动态行为（例如振荡频率、潜伏期、时间延迟）。
5. 基于需要确认实现的假设的数值（例如准确度值、检测阈值、响应时间、瞬态时间、最小/最大边界）。

确认计划应描述基于被确认的具体功能，确认方法的应用程度。

* + 1. **验证计划**

需求验证计划应贯穿整个开发过程。该计划应说明如何证明需求是完整和正确的，以及如何管理假设。除非另有证明，否则确认计划应包括对以下内容的描述：

1. 拟采用的方法。
2. 要收集或生成的数据。
3. 应记录的内容（如总结、评审或调查）。
4. 及时获取需求确认信息的途径。
5. 确认标准（完整性和正确性检查）。
6. 确认状态的跟踪过程，包括对要求进行变更时。
7. 与确认相关的角色和职责。
8. 关键验证活动时间表。
9. 在不同的设计级别和开发阶段管理假设的方法。
10. 用于提供需求定义与确认活动独立性的方法。
11. 确认活动期间如果发现任何问题，应遵循的流程。
12. 验证总结中包含的数据。

确认过程的某些方面也可作为验证的一部分，应与验证计划协调。

* + 1. **验证跟踪**

为了跟踪需求验证过程的状态，需要一个验证矩阵或其他适当的方法。验证的详细程度应根据需求所涉及功能的开发保证水平来确定，并应在验证计划中明确描述。最终数据应包含在验证总结中。具体格式由申请人决定，但可能包括：

1. 需求。
2. 需求来源。
3. 相关功能。
4. 开发保障等级。
5. 所用验证方法。
6. 验证支持性证据参考。
7. 确认结论（有效/无效）。
   1. **软件需求完整性要求**
      1. **概述**

软件设计标准应具有需求完整性的相关要求。

* + 1. **软件高层需求要求**

软件高层需求须满足的要求如下：

1. 符合系统要求。确保软件所执行的系统功能得到定义，系统的功能、性能和安全性相关需求由高层需求满足，并且派生需求及其存在的原因得到明确定义。
2. 准确性和一致性。确保每个高层需求准确、明确且足够详细，并且这些需求之间不相互冲突。
3. 与目标计算机的兼容性。确保高层需求与目标计算机功能之间不存在冲突，尤其是系统响应时间和输入/输出硬件。
4. 可验证性。确保每个高层需求都能得到验证。
5. 符合标准。确保在软件需求过程中遵循软件需求标准。
6. 可追溯性。确保分配给软件的功能、性能和安全相关需求被开发为高层需求。
7. 算法方面。确保所提出的算法的准确性和行为，特别是在不连续性领域。
   * 1. **软件低层需求要求**

软件低层需求须满足的要求如下：

1. 符合高层需求。确保底层需求满足高层需求，且派生需求及其存在的设计依据被正确定义。
2. 准确性和一致性。确保每个低层需求准确无误，且低层需求之间不相互冲突。
3. 与目标计算机的兼容性。确保低层需求与目标计算机的特性之间不存在冲突，特别是总线负载、系统响应时间以及输入/输出硬件等资源的使用。
4. 可验证性。确保每个低级需求都能得到验证。
5. 符合标准。确保在软件设计过程中遵循软件设计标准。
6. 可追溯性。确保高层需求和派生需求被开发成低层需求。
7. 算法方面。确保所提出的算法的准确性和行为，特别是在不连续性领域。
8. 机载软件开发与验证
   1. **软件生命周期阶段划分规范**
      1. **概述**

软件设计标准应具有生命周期阶段划分规范的相关要求。

* + 1. **生命周期的阶段划分**

软件生命周期的阶段包括：

1. 软件规划阶段，该阶段定义并协调项目中软件开发和集成活动。
2. 软件开发阶段，该阶段负责生成最终的软件产品，具体包括软件需求分析、软件设计、软件编码以及集成测试。
3. 软件集成阶段，该阶段包括软件验证、软件配置管理、软件质量保证以及认证联络。重要的是，集成阶段与软件规划和开发阶段在整个软件生命周期中同步进行。
   1. **软件设计与编码规则**
      1. **概述**

软件设计标准应具有设计与编码规则的相关要求。

* + 1. **软件设计阶段规则**

软件设计阶段的规则是：

1. 软件体系结构和低级需求是根据高级需求开发的。
2. 定义并提供给系统过程的衍生低级需求，包括系统安全评估过程。
   * 1. **软件编码阶段规则**

软件编码阶段的规则是：

1. 源代码是根据低级需求开发的。
   1. **软件验证目标**

软件验证过程的目标是检测并报告在软件开发过程中可能引入的错误。消除这些错误是软件开发过程的一部分。软件验证过程旨在验证：

1. 分配给软件的系统需求已开发为满足这些系统需求的高级需求。
2. 高级需求被转化为软件架构，而低级需求则满足这些高级需求。如果在高级需求与低级需求之间存在一个或多个层次的软件需求，那么这些层次的需求将依次开发，确保每个较低层次的需求都能满足其上级需求。如果代码直接从高级需求生成，则不适用上述规则。
3. 软件体系结构和低级需求已开发成源代码，以满足低级需求和软件体系结构。
4. 可执行对象代码满足软件需求（即预期功能），并提供对不存在非预期功能的信心。
5. 可执行对象代码对软件需求具有鲁棒性，因此能够正确响应异常输入和条件。
6. 用于执行该验证的方法在技术上正确且完整，适用于软件级别。
   1. **软件配置管理与质量保证目标**
      1. **概述**

软件设计标准应具有配置管理与质量保证目标的相关要求。

* + 1. **软件配置管理目标**

软件配置管理阶段的目标是：

1. 每个配置项及其后续版本均被明确标记，以便为配置项的控制和参考建立基础。
2. 为后续软件生命周期过程活动定义基线，允许对配置项进行参考、控制和可追溯性。
3. 问题报告过程记录软件计划和标准的不合规性、软件生命周期过程输出的缺陷、软件产品的异常行为，并确保这些问题得到解决。
4. 变更控制提供在整个软件生命周期中记录、评估、解决和批准变更。
5. 变更评审确保对问题和变更进行评估、批准或不批准，批准的变更得到实施，并通过软件计划过程中定义的问题报告和变更控制方法向受影响的过程提供反馈。
6. 状态会计为软件生命周期过程的配置管理提供数据，涉及配置识别、基线、问题报告和变更控制。
7. 归档和检索确保了与软件产品相关的软件生命周期数据在需要复制、重新生成、重新测试或修改软件产品时能够被检索。发布活动的目标是确保仅使用授权软件，特别是在软件制造过程中，同时这些软件也应归档并可检索。
8. 软件负载控制确保可执行对象代码和参数数据项文件（如有）在安全可控的条件下加载到系统或设备中。
9. 软件生命周期环境控制确保用于生产软件的工具被识别、控制和可检索。
   * 1. **软件质量保证目标**

软件质量保证阶段目标通过确保这些阶段按照批准的软件计划和标准执行，从而为软件生命周期生产符合其需求的软件提供信心。软件质量保证阶段的目标是获得以下保证：

1. 制定软件计划和标准，并进行审查，以符合需求并保持一致性。
2. 软件生命周期过程，包括供应商的生命周期过程，均符合批准的软件计划和标准。
3. 满足软件生命周期过程的转换标准。
4. 对软件产品进行一致性审查。
   1. **软件生命周期文档清单及交付要求**
      1. **概述**

软件设计标准应具有软件生命周期文档清单及交付要求的相关要求。

* + 1. **软件生命周期文档清单**

软件生命周期文档清单通常包括以下内容，可根据申请人的需求进行调整：

1. 计划过程输出：
2. 软件认证方面规划(PSAC)
3. 软件开发计划(SDP)
4. 软件验证计划(SVP)
5. 软件配置管理计划(SCMP)
6. 软件质量保证计划(SQAP)
7. 开发过程输出:
8. 软件需求标准
9. 软件设计标准
10. 软件代码标准
11. 高级和低级软件需求
12. 软件设计描述
13. 源代码
14. 可执行目标代码
15. 验证过程输出:
16. 软件验证案例和程序
17. 软件验证结果
18. 评审和分析结果
19. 测试结果和覆盖率数据
20. 配置管理和质量保证过程输出：
21. 配置索引
22. 问题报告
23. 质量保证记录
24. 认证联络过程输出：
25. 软件成果总结(SAS)
    * 1. **软件生命周期数据交付要求**

在软件生命周期中，数据用于计划、指导、解释、定义、记录或提供活动的证据。这些数据支持软件生命周期过程、系统或设备认证，以及认证后的软件产品修改。软件生命周期数据交付要求如下：

1. 明确无误：如果信息是用只能允许一种解释的术语来写的，那么它就是明确无误的，必要时可借助定义。
2. 完整性：当信息包含必要和相关的需求或描述性材料；对有效输入数据的范围进行了定义；使用的图表进行了标注；术语和测量单位被定义时，信息即为完整。
3. 可验证性：如果信息能够被一个人或工具检查其正确性，则该信息是可验证的。
4. 一致性：如果信息中不存在冲突，则认为信息是一致的。
5. 可修改性：如果信息是结构化的，并且具有可以完全、一致和正确地进行更改的风格，同时保留结构，那么信息是可修改的。
6. 可追溯性：当能够确定其组成部分的来源时，该信息即具有可追溯性。
7. 形式：软件生命周期数据的形式应确保在机载系统或设备的使用寿命期间，能够高效检索和审查软件生命周期数据。
   1. **适航符合性确认流程与核查依据**
      1. **概述**

软件设计标准应具有适航符合性确认流程与核查依据的相关要求。

* + 1. **认证联络流程**

认证联络过程的目标是：

1. 在整个软件生命周期中，建立申请者和认证机构之间的沟通和理解，以协助认证过程。
2. 通过批准软件认证计划达成关于合规手段的共识。
3. 提供合规性证明。
   * 1. **合规性证明**

申请人通过向认证机构提供软件生命周期数据，证明其软件生命周期过程符合软件计划。认证机构的审查可能在多个地点进行，例如申请人的设施、申请人的供应商的设施或认证机构的设施。这可能包括与申请人及其供应商的讨论。申请人安排这些审查，并根据需要提供软件生命周期数据。活动包括：

1. 解决认证机构审查过程中提出的问题。
2. 向认证机构提交软件完成情况摘要和软件配置索引。
3. 提交或提供认证机构要求的其他符合性数据或证据。
   * 1. **认证流程**

适航符合性认证流程如下：

1. 认证依据：认证机构与申请人协商，为待认证产品制定认证依据。该依据不仅定义了特定的法规，还规定了超出已发布法规的任何特殊条件。当认证产品进行修改时，认证机构会评估这些修改对原始认证依据的影响。在某些情况下，修改后的认证依据可能不会改变；然而，原有的合规方式可能不再适用于证明修改后的符合性，因此可能需要调整。
2. 软件方面的认证：认证机构评估软件认证计划的完整性和与已同意的合规方法的一致性，以满足认证基础。认证机构确保申请者提出的软件级别与系统安全评估过程的输出及其他系统生命周期数据一致。认证机构将告知申请者在获得认证前需解决的软件计划问题。
3. 合规性判定：在认证前，认证机构会确认待认证的产品，包括其系统或设备的软件部分，符合认证标准。对于软件部分，认证机构通过审查软件成就总结和合规性证据来完成这一过程。认证机构使用软件成就总结作为软件认证的概览。认证机构可自行决定在软件生命周期期间审查软件生命周期过程及其输出。
4. 机载软件工具合规性
   1. **工具分类与资格等级判定方法**
      1. **工具分类**

不再使用“开发工具”和“验证工具”的传统命名，而是通过工具对软件生命周期过程的影响进行分类，并据此判定其TQL。

工具按其功能影响划分为以下三类判断标准：

a)可插入错误的工具：

1. 工具输出直接作为最终软件产品的一部分，可能插入错误。
2. 替代了开发过程的某一环节。
3. 需采用最高级别的资格认证（TQL-1~3，依据软件等级）。

b)可未能发现错误的工具：

1. 工具自动化了某个验证过程，可能未能发现错误。
2. 若其输出结果未再独立验证，并用于简化或替代其他验证/开发过程，将对安全有更高风险。
3. 通常需满足TQL-4 要求或TQL-5。

c)可未能发现错误但影响较小的工具：

1. 工具仅支持特定验证活动，且不会替代其他过程，或其输出已通过其他手段验证。
2. 一般仅需TQL-5（最低级别）即可。
   * 1. **TQL等级**

若工具能消除、减少或自动化流程，且其输出未按规定进行验证，则适用的TQL如表1所示。根据工具使用场景及其在软件生命周期各阶段的潜在影响，共划分为五个等级：TQL-1为最严格级别，TQL-5为最宽松级别。评估特定工具的影响时，需按标准1至标准3的顺序依次考量各项指标。工具质量等级应尽早与审批部门协调确定。

表1 工具鉴定等级确定

|  |  |  |  |
| --- | --- | --- | --- |
| 确保等级 | 准则 | | |
| 1 | 2 | 3 |
| AL1 | TQL-1 | TQL-4 | TQL-5 |
| AL2 | TQL-2 | TQL-4 | TQL-5 |
| AL3 | TQL-3 | TQL-5 | TQL-5 |
| AL4 | TQL-4 | TQL-5 | TQL-5 |
| AL5 | TQL-4 | TQL-5 | TQL-5 |

* 1. **工具运行集成过程与环境安装要求**
     1. **工具运行集成过程**

工具运行集成过程的目的是将工具可执行对象代码安装到工具运行需求中指定的运行环境中，并结合用户手册等补充说明，确保工具在该环境中按预期运行。当满足计划的过渡条件时，可启动或重新启动该过程。输入要素包括工具可执行对象代码（必要时包含所有相关文件）和用户手册（可能作为工具需求的一部分）。主要输出成果为安装到运行环境中的工具可执行对象代码以及工具安装报告。

* + 1. **工具运行集成过程目标**

工具运行集成过程的目标是：将工具可执行对象代码安装到工具操作环境中。

* + 1. **工具运行集成活动**

该过程的活动包括：

1. 工具可执行对象代码（包括所有相关文件，如配置文件）应安装到工具操作环境中。
2. 在工具操作集成过程中检测到的不适当或错误输入应提供给产生这些不适当或错误输入的过程。
3. 应生成工具安装报告。
   1. **工具运行验证与确认过程**
      1. **工具运行验证和确认过程目标**

工具运行验证与确认过程旨在确保其输出结果及功能符合软件生命周期需求。该过程包含验证与确认两个环节。

验证目标包括：检测并报告开发过程中可能引入的错误，而错误修正属于开发过程的常规操作。验证目标具体为：

1. 工具运行需求需完整、准确、可验证且保持一致性。
2. 工具在运行环境中安装后的功能及输出结果须符合运行需求。

确认目标则涉及：

1. 核实工具运行需求是否充分且正确，以消除、减少或自动化PSAC（过程、系统和架构）中识别的流程。
2. 确保工具在运行环境中满足软件生命周期流程的需求。
   * 1. **工具运行验证和确认过程活动**

工具运行验证与确认目标通过审查、分析和测试的综合手段得以实现。审查、分析及测试用例的开发，用于评估工具操作需求的内部一致性和完整性。测试过程的执行则展示了工具可执行对象代码是否符合操作需求，并验证其与工具运行环境的兼容性。

将工具集成到工具运行环境中，应能验证工具运行要求的正确性和完整性。

工具运行验证和确认过程的输入包括工具运行要求、工具可执行对象代码和工具验证计划。

工具运行验证和确认过程的输出是工具运行验证用例和程序以及工具运行验证和确认结果。

工具运行验证和确认活动包括：

1. 工具运行要求验证：应对工具运行要求进行评审和分析。
2. 相关过程的识别和证明：工具操作要求应证明通过使用该工具消除、减少或自动化的过程的覆盖范围。
3. 工具运行环境中的验证和确认：安装在工具运行环境中的工具符合软件生存周期过程的需求。该活动既是工具可执行对象代码对工具运行要求的符合性验证，也是工具运行要求的确认。活动包括：
4. 应开发测试用例和程序，以确保工具满足工具操作要求。这些测试用例和程序应基于需求。
5. 应在工具可执行对象代码安装到工具操作环境后执行测试程序。
6. 应审查测试结果，以确保测试结果正确，并解释实际结果与预期结果之间的差异。
7. 为了验证工具操作要求，应执行该工具。所选的输入集应代表实际工具使用及其在工具操作环境中的接口。此验证应确认工具操作要求是正确和完整的。
   1. **商用工具资格确认方法**
      1. **合格的COTS工具**

COTS工具通常是在任何特定软件项目之外开发的，并且预期由多个用户使用。

本节指南涵盖用户手册中描述的、由工具用户可配置的COTS工具，而用户修改的工具不在本节指南范围内。

与任何其他需要鉴定的工具一样，COTS工具需要符合与必要TQL相适应的目标。但是，为了满足不同过程（工具操作开发过程、工具开发过程和所有相关验证过程）的目标，应该提议在工具用户和工具开发者之间进行责任分离。

* + 1. **COTS工具鉴定方法**

商用工具的开发通常缺乏针对软件生命周期的具体操作需求文档。为向用户清晰传达工具功能与预期用途，开发者需提供包含软件生命周期适用操作要求的开发者需求文档。用户通过该文档判断工具是否满足软件生命周期需求，并提交包含工具实际使用要求的TOR文件。值得注意的是，用户提供的TOR往往比开发者提供的TOR更为详尽。

对于COTS工具，工具鉴定活动通常由两个实体执行：

1. 工具开发人员：开发工具的实体。
2. 工具用户：在给定软件项目范围内使用工具的实体。

为了满足所有工具鉴定目标，下面定义了COTS工具开发人员和工具用户活动。应注意的是，对于TQL-5鉴定，目标可能无需来自工具开发人员的任何工具鉴定数据而得到满足。

* + 1. **工具开发人员的鉴定活动**

为了提供合格的工具，工具开发人员应：

1. 本文件的指导开展活动，并生成必要的工具生命周期数据，以证明已满足分配给工具开发人员的所有目标。
2. 提供一个开发人员-TOR，以确定工具中实现的功能：
3. 所有预期工具操作环境的描述。
4. 输入文件的描述，包括格式、语言定义等。
5. 输出文件的描述，包括格式和内容。
6. 工具所有预期功能行为的要求（从软件生存周期过程的角度）。
7. 需要解决工具应检测的异常激活模式或输入不一致问题。这些需求应考虑这些模式对工具功能和输出的影响。
8. 适用的用户信息，例如用户手册或安装指南（若这些文档未作为工具要求数据的一部分提供，则参考该手册或指南）。
9. 提供一套与TQL相适应的初步工具鉴定数据：
10. 开发人员TQP：开发人员TQP应限于分配给工具开发人员的活动。此外，开发人员-TQP应描述工具用户如何使用初步鉴定数据。。
11. 开发人员TCI：开发者TCI应限于工具开发者执行活动期间提供的数据。
12. 开发人员TAS：开发TAS应限于分配给工具开发人员的活动。
    * 1. **工具用户的工具鉴定活动**

商用工具的选用通常在软件规划阶段完成。在此过程中，用户需要评估该工具是否能满足软件开发生命周期流程的需求，以及其与操作环境的兼容性。同时，还需结合软件项目实际情况，评估COTS工具生命周期数据的可用性和相关性，并确认是否具备支持适当TQL认证的初步验证数据集。

之前已通过鉴定的COTS工具不会自动被视为适用于需要批准的其他项目。

如果现有工具生命周期数据不足以证明符合适用于工具开发者的全部目标，则工具用户应补充数据或提出替代方法。该方法应在TQP中描述。

为了评估工具运行环境中的商用工具，工具用户应补充开发人员的活动。

1. 用户TOR应包括但不限于以下内容：
2. 开发者TOR内容或对适用的开发者TOR内容的引用。
3. 工具使用背景描述，包括与其他工具的接口以及工具输出文件在最终软件中的集成。
4. 工具操作环境描述（工具将被安装的位置）。这包括对开发人员-TOR中包含的预期工具操作环境进行分析。
5. 工具操作使用说明（包括所选选项、参数值、命令行）。
6. 工具在特定操作环境中的操作输入描述。此描述的目的是确定所有可能的输入数据及其相关结构。此描述的完整性和准确性将允许开发工具操作验证数据，涵盖工具的操作使用。
7. 规定工具输出行为的性能要求。
8. 用户TQP应包括但不限于以下内容：
9. 对开发商TQP的引用。
10. 工具开发人员和工具用户在目标和活动方面的职责划分。
11. 用户TCI应包括但不限于以下内容：
12. 对开发人员TCI的引用。
13. 工具用户生成数据的识别。
14. 用户TAS应包括但不限于以下内容：
15. 对开发者TAS的引用。
16. 确定在工具执行过程中必须使用的所有选项。
17. 证明工具用户执行的工具安装操作和工具配置操作符合工具开发者提供的用户手册中的建议的演示（分析或审查）。
18. 工具开发人员和工具用户在目标和活动之间的责任划分与TQP的偏差。
19. 工具开发人员和工具用户验证活动的结果。
    * 1. **开发人员和用户活动的协调**

工具用户将对工具开发者执行的目标进行评估，因为工具用户最终负责鉴定。

TQP应参考开发人员TQP并提供协调手段，TAS和TCI也应参考开发人员TAS和开发人员TCI。

在工具符合性审查期间，工具用户应分析工具开发者的符合性审查报告，并在必要时验证审查期间提出的纠正措施的状态。

* 1. **工具配置管理**
     1. **工具配置管理过程目标**

工具配置管理过程目标是：

1. 每个配置项及其后续版本都进行了明确的标记，从而为配置项的控制和参考建立了基础。
2. 为工具的进一步生命周期过程活动定义了基准，允许对配置项进行参考、控制和追溯。
3. 问题报告过程记录工具计划和标准的流程不合规性，记录工具生命周期过程输出的缺陷，记录工具产品的异常行为，并确保这些问题得到解决。
4. 变更控制规定了在整个工具生命周期内记录、评价、解决和批准变更。
5. 变更审查确保对问题和变更进行评估，批准或否决问题和变更，批准的变更得到实施，并通过工具规划过程中定义的问题报告和变更控制方法，向受影响的流程提供反馈。
6. 状态会计为工具生命周期过程的配置管理提供数据，包括配置标识、基线、问题报告和变更控制。
7. 存档和检索确保在需要复制、再生、重新测试或修改工具产品时，能够检索到与工具产品相关的工具生命周期数据。发布活动的目标是确保在软件生命周期过程中仅使用授权的工具，此外还应存档并可检索。
8. 工具生命周期环境控制确保用于生产工具本身的其他工具被识别、控制和可检索。
   * 1. **工具配置管理过程活动**

工具配置管理过程包含配置识别、变更控制、基线建立及工具产品存档等环节，其中涉及相关工具生命周期数据。该过程不会因认证机构对工具产品的认可而终止，而是会持续贯穿最终软件所涉及的系统或设备的整个使用寿命周期。若供应商负责执行工具生命周期管理活动，则需将配置管理措施延伸至供应商层面。

1. 配置标识需包括以下活动：
2. 应为工具生命周期数据建立配置识别。
3. 应为每个配置项、每个单独控制的配置项组件以及构成工具产品的配置项组合建立配置标识。
4. 在实施变更控制和可追溯性分析之前，应确定配置项。
5. 在配置项被其他工具生命周期过程使用或被其他生命周期数据引用之前，应先确定该配置项的配置。
6. 变更控制需包括以下活动：
7. 变更控制应通过提供防止变更的保护措施来维护配置项和基线的完整性。
8. 变更控制应确保对配置项的任何变更都需要对其配置标识进行变更。
9. 对基准和变更控制下的配置项进行变更以生成衍生基准时，应记录、批准并跟踪这些变更。问题报告与变更控制相关，因为所报告问题的解决可能会导致对配置项或基准的变更。
10. 基线建立需包括以下活动：
11. 应为用于认证信用的配置项建立基准。（可建立中间基准，以帮助控制工具生命周期过程活动）。
12. 应在受控软件库（物理、电子或其他）中建立基线，以确保其完整性。一旦建立基线，应防止其被更改。
13. 应遵循变更控制活动，以从已建立的基线中得出衍生基线。
14. 如果需要为工具生命周期过程活动或与先前基线开发相关数据提供信用，则基线应可追溯至其来源的基线。
15. 如果需要为工具生命周期过程活动或与先前配置项开发相关联的数据提供信用，则配置项应可追溯到其衍生的配置项。
16. 基线或配置项应可追溯到其标识的输出或与其关联的过程。
17. 归档检索和释放需包括以下活动：
18. 与工具产品相关的工具生命周期数据应可从批准来源（例如，开发组织或公司的档案）中获取。
19. 应建立程序以通过以下方式确保存储数据的完整性（无论采用何种存储介质）：

确保不会进行未经授权的更改。

选择能最大限度减少再生错误或劣化的存储介质。

防止数据随时间推移而丢失或损坏。根据所使用的存储介质，这可能包括定期运行介质或刷新存档数据。

将重复拷贝存储在物理上独立的存档中，以最大程度降低灾难发生时丢失的风险。

1. 应验证复制过程以生成准确的副本，并且应存在确保无错误地复制工具可执行对象代码的程序。
2. 应在软件生存周期过程范围内使用前识别和发布配置项。
3. 应建立数据保留程序，以便对工具进行修改。
   1. **工具合规性数据**
      1. **工具资格认证流程和其他整体流程数据**
4. PSAC中的工具特定信息：
5. 工具的识别及其预期用途，包括其在软件生存周期过程中的影响。
6. 通过使用工具消除、减少或自动化流程所申请的认证信用的详细信息。
7. 证实工具中所采用的任何技术或理论（例如数学理论）的成熟度和技术背景，以显示其适用性。
8. 为工具提出的TQL和支持性依据。
9. 工具来源（例如，内部、COTS、第三方）。
10. 工具确认所涉及的相关方及其具体职责，包括由谁负责满足特定目标。
11. 工具操作要求定义过程、工具操作集成过程和工具操作验证与确认过程的描述（或参考说明这些过程将被描述的位置）。
12. 工具将使用的工具操作环境描述。
13. 若工具认证数据被重复使用，则需明确该工具的先前应用情况、重复使用的策略和依据，以及任何适用的重新认证活动。对于第三方工具或商用工具，供应商可提供先前应用的相关信息，因为用户可能无法获取此类信息。
14. 参考TQP，或者如果没有生成TQP（对于TQL-5)，参考支持工具鉴定的数据。
15. 工具资格计划（TQP）：
16. 工具的识别，以及适用时的用户配置。
17. 资格考虑因素，包括拟议的TQL和符合本文件目标的方法。
18. 工具、其接口和架构的功能概述，以及任何外部组件的标识。
19. 工具操作环境的描述，如果不同，则工具验证环境的描述。
20. 描述申请人将使用的手段，以便向认证机构提供工具生命周期过程活动的可见性，从而可以计划工具审查。
21. 工具生命周期描述和将要执行的确认活动。
22. 将生成工具确认数据。
23. 可能影响鉴定过程的任何其他考虑因素，例如停用代码、COTS工具、重复使用、工具鉴定（用于开发或验证工具的其他工具）、替代鉴定方法、工具服务历史以及确保工具确定性的方法。
24. 工具生命周期过程中的组织责任。
25. 如果使用供应商，则应确保供应商的流程和输出符合批准的工具计划和标准。
26. 工具开发计划：
27. 标准：确定工具要求、工具设计标准和项目工具代码标准。
28. 工具生命周期描述：指用于构建项目所需特定工具生命周期的描述，包含工具开发流程的过渡标准。该描述与TQP中提供的摘要不同，它提供了足够详细的开发活动说明，以确保工具生命周期流程的正确实施。
29. 工具开发环境：关于所选工具开发环境的硬件和软件说明。
30. 工具验证计划：
31. 独立性：在需要时，对建立验证独立性的方法的描述。
32. 验证方法：对工具验证过程的每个活动所使用的验证方法的描述。
33. 过渡标准：计划中定义的进入工具验证过程的过渡标准。
34. 保护注意事项：如果使用了保护，用于验证保护完整性的方法。
35. 复验方法：用于修改，描述识别、分析和验证工具受影响区域的方法。
36. 工具产品中先前开发的组件：对于这些组件，如果验证过程的初始符合性基线不符合本文档，则描述满足本文档目标的方法。
37. 工具配置管理计划：
38. 环境：对将使用的TCM环境的描述，包括程序、工具、方法、标准、组织职责和接口。
39. 活动：工具生命周期中TCM过程活动的描述。
40. 过渡标准：进入TCM流程的过渡标准。
41. TCM数据.：由TCM流程生成的工具生命周期数据的定义，包括TCM记录、工具配置索引和工具生命周期环境配置索引。
42. 供应商控制：将TCM过程要求应用于供应商的手段。
43. 工具质量保证计划：
44. 环境：对TQA环境的描述，包括范围、组织职责和接口、标准、程序、工具和方法。
45. 权限：TQA的权限、责任和独立性的声明。
46. 活动：针对每个工具生命周期过程以及整个工具生命周期所要执行的TQA活动。
47. 过渡标准：进入TQA过程的过渡标准。
48. 时序：TQA过程活动与工具生命周期过程活动之间的时序关系。
49. TQA记录：指由TQA流程产生的记录的定义。
50. 供应商监督：描述确保供应商的流程和输出符合已批准工具计划和标准的方法。
51. 工具要求标准
52. 开发工具需求所使用的方法，例如结构化方法。
53. 用于表达需求的符号，例如数据流图和正式的规范语言。
54. 开发工具需求所用工具的限制。
55. 用于分析衍生工具要求对工具操作要求的影响的方法。
56. 工具设计标准：
57. 拟采用的设计描述方法。
58. 要使用的命名约定。
59. 开发工具设计描述所用工具的限制。
60. 复杂性限制，例如嵌套调用或条件结构的最大级别、无条件分支的使用以及代码组件的入口和出口点的数量。
61. 工具代码标准：
62. 要使用的编程语言或定义的子集。对于一种编程语言，指明明确界定该语言的语法、控制行为、数据行为和副作用的数据。这可能需要限制使用该语言的某些功能。
63. 工具源代码呈现标准（例如，行长度限制、缩进和空白行使用）和工具源代码文档标准（例如，作者姓名、修订历史、输入和输出以及受影响的全局数据）。
64. 组件、子程序、变量和常量的命名约定。
65. 对允许的编码惯例所施加的条件和约束，例如软件组件之间的耦合程度、逻辑或数值表达式的复杂性以及使用这些表达式的原因。
66. 开发工具源代码所用工具的限制。
67. 工具生命周期环境配置索引：
68. 确定工具开发环境，如工作站和操作系统。
69. 确定用于开发提交进行鉴定的工具的工具（例如编译器和链接编辑器）。
70. 如果与工具开发环境不同，应确定工具验证环境。
71. 确定合格工具及其相关工具认证数据。这些工具是在提交进行认证的工具的生命周期过程中使用的工具。
72. 工具配置索引：
73. 工具产品。
74. 工具可执行对象代码，包括所有相关文件。
75. 每个工具源代码组件。
76. 如果使用，工具产品中先前开发的组件。
77. 工具生命周期数据。
78. 如果单独包装，则参考工具生命周期环境配置索引。
79. 存档和发布媒体。
80. 构建工具可执行对象代码（包括编译和链接的指令和数据等项目）的说明，以及恢复工具以进行再生、测试或修改所使用的步骤。
81. 工具问题报告：
82. 识别观察到问题的配置项或工具生命周期过程活动。
83. 确定要修改的配置项目（或描述要更改的过程）。
84. 问题描述，使问题得以理解并解决。
85. 为解决所报告问题而采取的纠正措施描述。
86. 工具配置管理记录：TCM过程活动的结果记录在TCM记录中。示例包括配置标识列表、基线、变更历史报告、存档记录和发布记录。这些示例并不意味着需要生成这些特定类型的记录。
87. 工具质量保证记录：TQA过程活动的结果记录在TQA记录中，这些记录可能包括TQA审查或审计报告、会议记录、授权过程偏差记录和工具符合性审查记录。除了涉及工具开发的TQA记录外，还将有涉及工具使用的软件质量保证记录。
88. 工具完成情况总结：
89. 工具概述：简要描述该工具的主要功能、输入和输出，以及与TQP中提出的工具概述的任何差异。
90. 资格考虑事项：重申TQP中所述工具的TQL，还确定操作环境。
91. 工具生命周期：总结实际的工具生命周期，并解释与TQP中确定的工具鉴定活动之间的差异。
92. 工具生命周期数据说明：详细阐述工具生命周期数据生成过程中与技术质量计划（TQP）提案的差异、各数据之间的关联性及其与其他定义工具数据的关系，以及向认证机构提供数据的具体方式。通过配置标识符和版本号，明确引用适用的工具配置索引及工具生命周期环境配置索引。
93. 附加注意事项：总结可能需要认证机构注意的确认问题，并解释与技术质量方案中关于此类注意事项的建议之间的任何差异。应参考适用的数据项，例如特殊条件或解释性材料。
94. 供应商监督：描述供应商过程和输出如何符合批准的工具计划和标准。
95. 工具标识：通过部件编号和版本识别工具配置。
96. 变更历史记录：如适用，包括影响工具功能的变更总结，并确定自上次鉴定以来对工具生命周期过程的任何变更或改进。
97. 工具状态：包含批准时未解决的工具问题报告摘要以及对问题对工具功能的影响的评估。此外，还应包括工具的任何功能限制。
98. 符合性声明：包含对本文档的符合性声明以及用于证明工具计划中规定的标准符合性的方法总结。还涉及认证机构做出的其他裁定，以及工具完成总结中未涵盖的工具计划、标准和本文档的任何偏离。
99. 软件完成情况总结中的工具特定信息：
100. 工具的识别。
101. 通过工具使用寻求的认证信用的详细信息，即要消除、减少或自动化的软件过程。
102. 参考工具完成情况总结。
103. 工具开发、验证和集成过程与工具计划的符合性声明，包括工具用户活动。
104. 分析工具的开放问题报告，以确保工具的行为仍然符合工具操作要求。
105. 描述PSAC中发现的任何工具使用差异。
106. 软件生存周期环境配置索引中的工具特定信息：
107. 工具的识别。
108. 参考工具配置索引和TAS。
     * 1. **工具开发过程和相应验证活动期间产生的工具确认数据**
109. 工具要求：
110. 工具功能和技术特征的描述，包括操作模式。
111. 用户说明、安装说明、错误信息列表和约束。通常将其打包为用户手册。用户手册可能是工具要求的一部分，也可能打包在一份或多个文档中。
112. 描述用户定制工具能力的要求。
113. 功能需求，具有适当详细程度。
114. 如必要，为符合工具操作环境而制定的具体要求。
115. 解决失效模式和对不一致输入的响应的具体要求。
116. 工具在异常工作条件下的预期响应。
117. 对于一组工具，该组内各工具之间的接口要求。
118. 工具设计说明：
119. 工具架构的描述，定义实现需求的工具结构。
120. 详细说明如何将工具需求分配到工具架构中。
121. 软件体系结构内部和外部的输入/输出描述（例如数据字典）。
122. 设计的数据流和控制流。
123. 调度程序。
124. 使用时的防护措施。
125. 软件组件的描述，不论是新开发的还是以前开发的，如果是以前开发的，则需引用其基准。
126. 实现工具要求并可追溯至工具要求的低级工具需求。
127. 由工具设计过程产生的衍生低级工具需求。
128. 工具源代码：
129. 用脚本或源语言编写的代码。
130. 链接指令。
131. 用于从工具源代码生成工具可执行对象代码的编译器指令（如果使用，这取决于所使用的语言）。该数据应包括工具标识符，包括名称和修订日期或版本，如适用。
132. 工具可执行对象代码：工具可执行对象代码是工具运行环境中处理单元可直接使用的工具形式。需要说明的是，根据所选源代码语言的不同，可能无需进行编译和链接操作。例如使用脚本语言实现的工具时，其可执行对象代码应理解为该工具的可执行格式。相关文件（如运行工具所需的参数配置文件）均视为包含在可执行对象代码中。此类代码通常在工具开发环境中生成。若同一工具需在多个运行环境中使用，系统会自动生成多种配置方案。
133. 工具验证用例和程序：
134. 评审和分析程序：除工具验证计划中的描述外，所用评审或分析方法的范围和深度。
135. 测试用例：每个测试用例的目的、输入集、条件、预期结果以达到所需覆盖标准以及通过/未通过标准。
136. 测试步骤：关于如何设置和执行每个测试用例、如何评价测试结果以及要使用的测试环境的逐步说明。
137. 工具验证结果：
138. 对于每次评审、分析和测试，应说明活动期间通过或未通过的每个流程以及最终的通过/失败结果。
139. 确定被评审、分析或测试的配置项或软件版本。
140. 包括测试、评审和分析的结果，包括覆盖范围分析和可追溯性分析。
141. 记录并跟踪使用问题报告流程发现的任何差异。

g)追踪数据：

1.工具操作要求和工具要求。

2.工具要求和低级工具要求。

3.低级工具要求和工具源代码。

4.工具要求和低级工具要求及其相关测试用例。

5.测试用例和测试步骤。

6.测试步骤和测试结果。

**7.6.3工具操作要求过程、工具操作集成过程以及相应验证和确认活动期间生成的工具鉴定数据**

a)工具操作要求：

1.工具使用场景描述，包含与其他工具的接口及输出文件与最终软件的集成方式。

2.工具运行环境描述（安装位置）。

3.输入文件说明（格式、语言定义等）。

4.输出文件说明（格式与内容）。

5.所有工具功能及技术特性需求（满足软件开发生命周期流程）。

6.针对工具应检测的异常激活模式或输入不一致问题的要求，需考虑这些模式对工具功能和输出的影响。（本项不适用于TQL-5）

7.用户信息（如用户手册、安装指南或相关引用），若未包含在工具需求数据中。

8.工具操作说明（含选项选择、参数值、命令行等）。

9.规定工具输出行为的性能要求。

d)工具安装报告：

1. 确定工具运行环境（硬件配置、操作系统配置、已安装选项等）。
2. 确定工具可执行对象代码的版本，包括所有配置文件和参数文件。
3. 如有必要，确定外部组件。
4. 确定工具的运行方式（命令行、脚本文件等）；可能需要参考用户手册。

c)工具运行验证和确认案例及程序：

1. 审核与分析流程.：除工具验证计划中的描述外，需明确所采用审核或分析方法的范围与深度。
2. 测试用例：每个测试用例的目的、输入集、条件及预期结果，以满足通过/未通过标准。
3. 测试流程：包含各测试用例的设置与执行步骤、选定输入参数、测试结果评估方式及所用测试环境的详细说明。

d)工具运行验证和确认结果：

1.对于每次评审、分析和测试，说明在活动期间通过或未通过的每个过程以及最终的通过/未通过结果。

2.确定所评审、分析或测试的配置项或工具版本。

3.包括测试、评审和分析的结果。

4.记录并跟踪使用问题报告过程发现的任何差异。

参 考 文 献

1. HB\_Z 400-2013 民用飞机航空电子软件管理指南
2. HB\_Z 415-2014 民用飞机机载电子设备通用指南
3. HB\_Z 421-2014 民用飞机机载系统和设备软件合格审定保证指南
4. MHT 1039-2011 通用航空术语
5. HB 7233-1995 民用机载计算机软件质量保证大纲编写指南
6. HB 8390-2013 民用飞机可加载软件要求
7. HB 8429-2014 民用飞机标准化大纲编写要求
8. HB 8658-2022 民用飞机机载系统和设备软件设计要求
9. HB 9101-2012 航空质量管理体系审核要求
10. 蔡喁,郑征,蔡开元.机载软件适航标准DO-178B/C研究[M].上海交通大学出版社,2013.
11. 徐剑,刘建方,韩铖熹.基于RTCA/DO-330的机载电子硬件工具鉴定准则研究[J].航空标准化与质量, 2014(6):3.DOI:10.13237/j.cnki.asq.2014.06.012.
12. 黄铭媛,欧旭坡,宋智桃.民用飞机产品研制保证等级分配原则研究[J].航空科学技术, 2015, 000(007):6-10.