# 绪论

## 课题来源及研究背景

本课题来源于工信部的民用飞机专项科研《航空发动机电子控制系统适航审定关键技术》中的子课题《航空发动机电子控制器软件和硬件适航审定技术研究》，针对大型客机涡扇发动机电子控制系统软件的适航审定问题，开展相关的基础技术研究。

2017年5月，C919大客机的首次试飞顺利完成，我国民用航空事业进入新的历史阶段。2010年年底，民航局受理了C919飞机型号合格证申请，开展C919飞机型号合格审定，制定了66份的审定计划，规划4000余份用于证明符合特定适航标准的技术报告计划和近1500项的实验验证计划。截至2018年12月，已完成近300项验证试验的试验目击审查，批准500余份的验证试验大纲和80余份的试验报告。按照计划，审查组还要完成包括审定试飞在内的、证明飞机设计符合适航要求的大量工程技术审查工作，根据研制进度和取证的工作量，预计2020年底取得C919飞机型号合格证。

为确保航空飞行器的安全性，飞行器的设计维护、人员培训和飞机操作等各个方面都要达到足够的安全与完整性。航空电子设备属于安全关键项目，要使用航空电子设备实现功能，其设计与制造的安全与完整性要达到适航要求，否则有可能危害人员或设备的安全。为了确保飞机系统的设计和制造符合要求的目标，不同的国家成立了不同的机构负责安全关键系统的适航审查。飞行批准认证机构包括欧洲航空安全局(EASA)和美国联邦航空局(FAA)等。我国的则是民航管理局（）。当飞机系统首次开发或升级时，飞行认证机构必须在系统设计获准飞行前进行适航审查并认证。只有符合预先定义的设计和操作限制，该型号的飞行器才能用于飞行。航空电子设备由硬件和软件两部分组成，对于硬件来说其安全性和完整性通常有着定量要求。而对于软件来说，很难通过统计测试和操作经验等方法来定量地展示其安全和可靠性水平。因此，适航软件的标准通过确保控制开发过程的质量，来确保软件的可靠性。在民用航空领域，DO178C是机载软件批准的主要标准。DO178C文件的目的是“为机载系统和设备提供软件生产指南，使这些系统和设备在满足适航要求的安全可靠性水平上履行其预期功能”。DO178C定义了航空业界关于机载软件批准的共识。开发方通过提交生命周期数据来获得适航认证，这些生命周期数据表明软件开发是否按照DO178C标准指定的过程和要求执行。

软件配置管理（SCM——Software Configuration Mangement）（引用 Wayne Baich SCM Coordination for Team Productivity）是指“协调软开发使得混乱减到最小的技术，它是一种标识、组织和控制修改的技术，目的是使错误达到最小并最有效地提高生产效率。”,它在软件产品开发的生命周期中,提供了结构化、有序化、产品化的管理软件工程方法,是软件开发和维护的基础。对软件产品的安全性和可靠性发挥着至关重要的作用。在民机机载软件开发的过程中，实施软件配置管理可以规范软件开发流程，提高工作效率，使民机机载软件更加安全、可靠。Do178标准对配置管理过程的定义、作用以及其他过程间的关系都进行了明确的说明，并规定了配置管理过程必须要达到的6个过程目标，（引用）， 但是因为民用航空领域的特殊性，在实际的适航开发和审定中，目前配置管理方法无法满足适航软件研发和审定对配置管理的要求，难以达到Do178c标准的目标。

国外对适航软件研发和审定中配置管理的研究，主要包括证据管理方法研究和证据审查方法研究， 证据管理的方法包括文本型的模板、结构化的文本、建立开发的元模型或信息模型，基于图形符号论证等，证据审查的方法包括专家审查、审查单、定性分析、定量分析等，但这些方法大多未用于工业实践，或是仍主要依靠专家经验，很难适用于我国目前的适航开发与审定的情况；而当前我国对适航软件配置管理的研究，基本局限在对适航配置管理流程的思考和经验性分析，或是特定适航软件配置管理工具的的开发，相对国外具有较大差距。软件开发的工业实践中，缺乏一套成熟的系统，配置管理计划与实施取决于配置管理人员自身水平，成效难以保证，得到证据难以使用，审定难度大；审定工作中，也面临着各种问题：1.研发方提供的审定材料是庞大而混乱的，给局方进行审查造成很大的困难；2.由于开发过程是漫长与复杂的，而审查工作则只是参与其中的几个快照，局方耗费大量的时间、人力，却不一定能找到所有的问题；3.审查主要依靠审查人员对标准理解和及其经验和阅历，不同审查人员在进行评审活动时可能得出的结论有差异，导致评审结果有争议。

综上所述，对配置管理过程的关键技术进行研究，规范研发方提供证据的形式与格式要求，能够为适航安全关键软件的研发和审定提供有力支持；，提出审定方审定证据的具体方法，研发辅助审查人员半自动化审定的工具，能够提高审定人员工作的效率，保证审查工作的规范和可靠，对我国国产大飞机走出国门，获得国际社会认可具有重大意义。

## 国内外研究现状

### 安全证据管理技术概述

**目前国际上保证安全证据结构的技术，可以分为基于论证方法管理、基于模型管理、基于文本模板管理三种类型。**

1. 基于论证的证据结构管理。论证是一种向相关方证明系统安全性达到了目标要求的方法，结构化的论证需要有直接的证据支持，人们为此提出了构建证据的具体技术，技术又可以分为基于图形构建证据结构和使用基于文本构建证据结构。

基于图形技术构建证据结构，对论据和论证证据进行表示。在[PS3, PS5, PS8, PS9, PS10])中使用了GSN技术，它可以用来显式地记录参数的元素和结构以及参数与证据的关系。在GSN中，论证的声明被记录为目标，证据项被记录在解决方案中。[PS20, PS22, PS72, PS78]) 使用了CAE技术，它提倡一种类似于GSN的三层方法，包括在一个论证中断言的一个顶级声明、对支持一个声明的参数的描述，以及对支持一个声明或论证的证据的引用。 [PS23, PS38, PS58, PS, PS])使用了BBN,它在一个有向非循环图中引入一个结构来表示它们之间的条件依赖关系。SSG [PS]是一种线性图，它将安全规范表示为节点，将证据及其关系表示为边。

基于结构化文本对论据和论证证据进行表示的方法。（PS18540）使用 了Trust cases技术，Trust cases指的是为安全声明、参数和证据归纳出结构化的文本格式，并使用参考文档的方式进行展示说明。[PS185]中，使用了结构化HTML标签来链接和构造各种作为安全证据的人工制品。[PS80]，为普通文本规定某种结构，显式的表示论点的关键部分;规定大纲格式构造参数和证据，使用数学中的几何证明结构给出陈述和理由;使用LISP编程语言的语法结构的特点，使用简短的名称和括号作为证据和参数。

2. 基于模型的证据管理是指使用模型描述安全证据结构，确定证据的规格与要求。

[PS]使用模型驱动的方法，为do - 178标准构建了UML概要文件，用于软件产品的开发和审定。

-[PS99]使用的是数据建模，在大型安全案例中，使用实体关系图来构造数据内容。

-[PS]利用了过程模型，对于产生证据数据的活动与过程，进行研究和建模，并使用基于树的结构进行展示。

3.文本模板(3%): 使用模板来提供预定义的表，以及以预定义文本格式构造证据的约束。

- [PS50]，使用了ACRuDA模板，该模板被用来 根据预先定义的安全案例结构构造证据。

- [PS19]提出了 Template Add-ons，它为预定义的一组文档提供模板，这些文档将在不同的系统开发和安全保证阶段生成。它还就文档化、半形式化描述以及验证和验证过程所需的方法提供了建议。

对证据进行评估的技术概述：

**目前国际上对证据进行评估的技术，大致可以分成四种:**

1.定性评估(%):是指使用非数值方法评估证据的技术。 PS,PS,PS,PS,PS,PS使用了论证，论证是定性评价中最广泛使用的方法。可以基于不受限制的自然语言、(半)结构化的自然语言或图形化的论证结构，如GSN。图形论证结构通常具有易于理解、检查和导航的优点。

- [PS，研究基于活动的质量模型，它使用质量矩阵来评估是否符合IEC62304标准。

- [PS]提出了，Evidence-confidence conversion process ，该过程通过评审过程评估安全证据，从而确定对系统安全的信任程度。。

2，检查列表(%):我们将这类技术分类为“待办事项列表”，其中包含一组需要在检查证据时回答或检查的指导问题。

-[PS]设计核对表，核对表根据系统设计的具体情况来评估证据。

-[PS]，基于目标/问题/度量框架[](GQM)。定义了用于评估产品和过程证据的顶级目标、为实现这些目标而需要回答的问题，以及提供可度量参考的度量标准，以便执行分析。

-[PS]以论证为基础的核对表，将核对表与论证相结合来评估证据。

-[PS]则采用了基于分类的问卷调查，问卷包含了305个涉及到软件安全风险分类和评估过程中安全属性和人工制品的问题。

,3.定量评估(%):使用数值方法评估证据的技术。

- [PS41,PS,PS,PS,PS])BBNs技术，它利用条件概率分布来评估存在不确定性的证据。该技术与BBN证据结构结合使用(第4.2节)。这是文献中最常用的证据评估定量技术。

- [PS]，提出了结合了定量评估和正式论证结构Modus方法。该方法基于定量推理，使用目标模型、专家启发式和概率模拟来评估安全案例的总体目标。

-[PS96]使用了Evidence Volume Approach证据量法，该方法允许内部专家对描述每一证据相对重要性的证据分配加权因子。然后为加权证据选择一个聚合函数来计算一个称为证据量的卷，根据这个卷选择结果(接受或拒绝)。

4 基于逻辑的评估(6%):使用逻辑公式的技术，以阐明和验证证据项及其关系的属性。例如一阶逻辑语句。基于逻辑的技术最适合于检查证据信息的格式良好性和一致性约束。例如，。。。使用OCL[23] 确保特定系统生成的证据项之间存在一致的链接，

### 软件配置管理 安全保证技术相关研究。

目前对软件配置管理的研究主要关注软件文档制品中的配置项的标识和跟踪。通过对比软件开发文档制品，对配置项进行追溯与管理（包括不同的软件开发文档制品中相同需求的追溯性） ，通过比较软件开发文档制品和代码制品，验证软件开发文档工件的需求是否被正确的编写成了代码；通过对代码制品进行比较，对状态修改、软件版本和依赖性进行管理，对功能或物理特性的变化进行控制。

[10-12]使用跟踪矩阵管理软件配置管理的文档构件。跟踪矩阵中的每个需求都由惟一ID标识，并位于第一行。文档构件的类型位于列中，与需求映射的配置项的索引放在矩阵的内容字段中。可跟踪性矩阵可用于跟踪不同文档构件中具有相同需求的配置项。

[13-14]应用信息检索技术实现了可追溯矩阵的自动生成。通过处理软件文档构件中的简单链接，使用可跟踪矩阵的优势在于易于读取和使用。

,[15] 提出通过使用依赖关系,跟踪关系，和文档制品的结构化信息。推断新文档产品之间的追踪关系

【content-based】（）提出了基于内容的配置管理系统，该系统生成“关联链接”，用来管理软件研究文档构件中高度抽象的非功能需求。关联链接由相关的两个配置项和配置项间需要遵循的规则组成，用于对软件研发文档制品进行配置管理

我国也有许多对配置管理过程的研究

严晓光 （软件配置管理的问题\_目的\_层次和策略）结合软件配置管理可能出现的活动工作量大、效率低、配置失控等问题，深入分析了问题出现的原因，讨论了配置管理的层次目标及各层次的特点，提出了业务导向、分层次管理、注重实效等原则和策略。

姜文(软件配置管理中的基线问题研究)分别归纳并给出了五种不同的基线定义，叙述了基线的层级、软件项目的阶段基线、实现阶段基线的水平等级和粒度，集中介绍了基线相关的技术问题

赵海英(软件需求变更管理的研究与实践) 提出了一种oucm，oucm创建了一个隔绝开发工作的，潜在无序的需求变更的流程方法，避免了正常开发工作被扰乱，减少不当需求变更引起的成本和工作量。

以上研究主要针对于配置管理中的某一方面进行专门研究，这两个研究则面向配置管理的整个流程。张薇 (软件配置管理系统的优化研究与设计) 对配置管理的流程进行了优化设计，提出了比较普遍全面的流程 王槐彬(基于工作流技术的软件配置管理过程建模)基于工作流技术，使用工作交流网建模技术对配置管理进行过程建模，并析了配置管理建模的优点。

前面提到的研究是针对软件配置管理本身，没有特别的对适航软件中的SCM，或结合DO178标准进行研究。为设计到安全保证和证据管理的内容。这两个，则对我国机载软件开发 结合Do178标准，进行了流程、和各项活动具体方法与策略的研究。杨娟（民机机载软件配置管理技术研究）从结合民用机载软件的特点，提出了符合机载软件要求的配置管理流程，并与DO178B中的配置管理活动进行了符合性对照。毛佳、胡林平（基于DO-178B的软件配置管理技术研究 ）以DO － 178B 为指导，详细阐述了开展各项软件配置管理活动的具体方法与策略。为提高数据控制的有效性，以满足DO － 178B 的A 级软件研发过程为例，说明了如何按照DO － 178B 中的两种控制类型对软件生命周期数据进行分类，最后，提供了符合DO － 178B 要求的说明

# 相关理论与技术研究

本章主要介绍适航领域相关理论、软件配置管理相关理论和技术。首先介绍了适航领域中机载软件相关的标准DO-178B/C，以及配置管理过程与DO178C标准中其他过程的关系， 对变更管理的基本概念、主要活动和主要功能和活动件关系等进行了介绍，然后对信息检索技术的常用技术和模型进行介绍；接下来介绍了本论文中使用的word embedding；最后介绍了学习排序算法的基本概念，以及在软件工程领域中的使用情况和已有的方法。

## 适航领域相关理论

本节首先介绍适航领域软件相关的标准，以及标准中规定的与普通软件开发不同的软件生命周期和相应的软件生命周期数据，

### DO-178B/C标准介绍

在20世纪70年代，美国联邦航空管理局(FAA)、欧洲航空安全局(EASA)和其他全球航空安全机构首先援引了RTCA/DO-178文件。该文件的标题是“机载系统和设备认证中的软件考虑事项”。“该文件的目的是确保飞行中软件的安全，它通过实施结构化的开发过程来做到这一点。自20世纪70年代以来，负责该文件的RTCA委员会每十年就对其进行实质性修订，以跟上技术的发展。2013年7月，FAA通过AC20-115C启用了第四代标准DO-178C(欧洲称为ED-12C)。目前，适航标准RTCA DO-178是商业和军事航空航天项目事实上的软件标准RTCA

SC-205委员编写DO-178C时，故意使其具有“非规定性”。这意味着改标准提供了描述“必须做什么”的一般目标，但没有解释“如何做”的例子。虽然这种方法提供了很大的灵活性，但也带来了很多不确定性。此外，DO-178C还增加了四个新的补充(即单独但相关的文件)，处理软件中使用的特定技术：其中，DO-330用于工具资格认证，DO-331用于基于模型的开发，DO-332用于面向对象技术，DO-333用于形式化方法。

### 目标、过程和数据的关系

DO-178C提供的是一种结构化的机载软件设计保证指南，它是基于目标，面向过程和活动的。

系统安全评估过程中，软件的安全性是根据与软件组件相关的故障条件在系统级确定的。这些安全状况可分为五类:“即灾难性的，危害性的，严重的，不严重的，以及没有影响的[25,26]。然后DO178C指南定义了五个不同的保证级别，它们与上述故障条件的分类相关(级别A到E，其中级别A是最高的，因此需要最严格的流程)。每一层的软件保证都与一组目标相关联，这些目标大多与基本的生命周期过程有关，DO-178C标准针对9个软件过程提出了71个安全目标，以保证航空软件以规范的方式进行开发，并以规范的方式排除软件缺陷，修复代码漏洞，从过程上保证软件最终达到所需的安全性[1]。

软件生命周期是指软件产品从概念提出到产品“死亡”的整个周期。就机载软件而言，DO-178B/C规定的软件生命周期过程包括软件计划、软件开发以及软件综合过程三个过程,如图4所示。

计划阶段指导软件开发和集成过程的四个过程。计划阶段为每个阶段制定计划来指导其目标和活动。软件开发阶段包含四个过程：软件需求过程确定软件将在程序中实现的软件需求，软件设计开发软件架构和较低层次的需求。软件编码执行实际的软件编码，通过定义的体系结构将软件需求转化为设计，软件集成过程编译源代码，然后链接并将其加载到目标计算机系统。综合过程包含了四个子过程。软件验证过程执行评审、分析和测试，以确保软件设计实现其需求，配置管理过程提供了适当的数据控制和变更管理机制，质量保证确保项目的开发过程符合生命周期过程目标，审定联络过程由审定人员定期审查计划、过程和工件，以确保符合标准的目标。在整个软件生命周期中，软件计划过程为软件开发和集成过程提供指导，软件开发过程是主线，软件综合过程与软件开发过程同时执行，并且软件综合过程执行在软件开发过程的各个子过程上，具体的关系如图5所示。



图5 DO-178B/C软件生命周期方阵图



图4 DO-178B/C生命周期过程结构图

DO-178C准给出了22项软件生命周期数据（Software Life [cycle](https://cloud.tencent.com/developer/information/cycle) Data）。需要说明的是：这不是说生命周期数据就是指这22份文档，这些文档可以是独立的，也可以合并处理，实践中建议使用更细粒度的数据项进行管理，具体见第3章。

生命周期数据是软件研制整个生命过程中所有数据的记录。它应该完整地、真实地记录了软件的整个生命过程：计划过程的输出、高级别需求的捕获、设计描述以及每一次需求的变更、每一次代码的改动、每一个测试用例、每一次版本升级、每一次同行评审，都要真实地记录下来，纳入配置管理的控制之下。生命周期数据是软件产品的载体，其形式可以多种多样，但一定要便于高效地检索和审查，生命周期数据具有确定性、完整性、一致性、可更改性、可追溯性等特征。

软件生命周期的过程、数据和目标是DO178C的基本要素，它们的关系如图6所示：目标是在过程中实现的，目标的实现通过软件生命周期数据得到证明；过程使用生命周期数据，也生成生命周期数据。



图6 DO-178B/C中基本要素的相互关系

### 适航审定

适航符合性验证是为了保障民用航空活动安全，采取不同的说明和验证方法获得所需要的证据资料向审查方表明产品对于适航条款的符合性。在标准领域，为了实现DO-178C标准的认证，美国联邦航空管理局（FAA）提供了软件审批指南（Software Approval Guidelines），帮助适航审定部门审定适航软件是否符合DO-178C标准。该指南对DO-178C中规定的活动以及输出进行了更为详实的解释说明，指出了其中需要关注的重点，为建立目标满足性论证模式提供了重要的指导。由于软件生命周期数据完整地、真实地记录了软件的整个研制过程，它就成了软件适航审定的举证依据。审查方通过评审研制单位提供的软件生命周期数据，来判断机载软件的研制过程是否满足DO-178C相应软件等级软件的目标，从而判断该机载软件能否能够获得适航批准。

## 软件配置管理理论

### 软件配置管理的定义

软件配置管理是指在开发过程中各阶段,管理计算机程序演变的学科,它作为软件工程的关键元素,已经成为软件开发和维护的重要组成部分,它提供了结构化的,有序化的,产品化的管理软件工程的方法。它涵盖了软件生命周期的所有领域并影响所有数据和过程。ieee标准729-1983就配置管理的内容进行了规范的定义

标识 识别产品的结构、产品的构件及其类型,为其分配唯一的标识符,并以某种形式提供对它们的存取。

控制 通过建立产品基线,控制软件产品的发布和在整个软件生命周期中对软件产品的修改。例如,它将解决哪些修改会在该产品的最新版本中实现问题。

状态统计 记录并报告构件和修改请求的状态,并收集关于产品构件的重要统计信息。例如,它将解决修改这个错误会影响多少个文件的问题。

审计和审查 确认产品的完整性并维护构件间的一致性,即确保产品是一个严格定义的构件集合。例如,它将解决目前发布的产品所用的文件的版本是否正确的问题。

生产 对产品的生产进行优化管理。它将解决最新发布的产品应由哪些版本的文件和工具来生成的问题。

过程管理 确保软件组织的规程、方针和软件周期得以正确贯彻执行。它将解决要交付给用户的产品是否经过测试和质量检查的问题。

小组协作 控制开发统一产品的多个开发人员之间的协作。例如,它将解决是否所有本地程序员所做的修改都已被加入到新版本的产品中的问题。

软件配置管理是一套规范、高效的软件开发基础结构。作为管理软件开发过程有效的方法,早已被发达国家软件产业的发展和实践所证明。可以系统地管理软件系统中的多重版本全面记载系统开发的历史过程包括为什么修改,谁作了修改,修改了什么管理和追踪开发过程中危害软件质量以及影响开发周期的缺陷和变化。

\\\软件配置管理的定义

管理软件维护预算纪律的计算机方案变化管理进程所有阶段的发展,而且他们作为关键软件工程已经成为发展的重要组成部分和软件的维护和供应、有秩序的和结构化方法制定管理软件工程。它涵盖了软件生命周期的所有领域，影响了所有的数据和过程。ieee标准729-1983定义了配置管理的内容

识别产品的结构、产品的元素和类型，为产品分配一个唯一的标识符，并以某种形式访问它们。

控制是通过为产品建立一个参考基础，控制软件的发布和整个软件生命周期对产品的修改来实现的。例如，它将解决在产品的最新版本中执行哪些更改的问题。

国家统计数据记录和报告所要求的部件的状态和修改，并收集有关产品部件的重要统计信息。例如，它将解决与修改错误有关的案件数量的问题。

审计和检查确认产品的完整性，并保持组件之间的一致性，即确保产品是一组严格定义的组件。例如，它将解决目前发布的产品的文档版本是否正确的问题。

产品生产的最佳管理。它将解决需要生产最新产品的文档和工具版本的问题。

流程管理确保了软件组织的规则、指导方针和周期的正确实现。它将解决交付给用户的产品是否经过质量测试和验证的问题。

团队合作控制了几个开发统一产品的开发人员之间的协作。例如，它将解决当地程序员所做的所有修改是否已经添加到新版本的产品中。

软件管理是一个指定和有效的软件开发基础设施。发达国家软件产业的发展和实践证明了它们作为软件开发过程管理方法的有效性。可以系统地管理多种版本的软件系统的正式记载的历史进程,包括系统发展的原因,已修改,修改,从而改变了管理和监测变化的缺陷和质量已经危及软件在发展进程和循环,影响了发展。

\\\\

### 下面简要介绍几个软件配置管理的基本概念,

主要包括软件配置与配置项、基线、软件版本。

软件配置与配置项

软件配置是指一个软件产品在软件生存周期各个阶段所产生的各种形式机器可读或人工可读和各种版本的文档、程序及其数据的集合。该集合中的每一个元素称为该软件产品软件配置中的一个配置项。

软件配置项即软件配置管理的对象。软件开发过程中,产生的所有信息构成软件配置,它们是代码源代码、目标代码以及数据结构内部数据、外部数据、文档技术文档、管理文档、需求文档、报告,其中每一项称为软件配置项,软件配置项是配置管理的基本单位。同时,开发过程中使用的环境,如操作系统、各种支撑软件、工具等,也可纳入软件配置管理范围。图为常规状态下软件开发阶段的配置项定义。

基线和配置管理数据库在软件开发过程中,由于各种原因,可能需要变动需求、预算、进度和设计方案等,尽管这些变动请求中绝大部分是合理的,但在不同的时机作不同的变动,难易程度和造成影响差别甚大,为了有效地控制变动,软件配置管理引入基线的概念。

基线标志软件开发过程的各个里程碑,.任一个sci,例如,设计师说明书,一旦形成文档并复审通过,即形成一个基线,它标志开发过程中一个阶段的结束。对于已成为基线的,虽然可以修改,但必须按照一个特殊的、正式的过程进行评估,确认每一处修改。相反,对于未成为基线的,可以进行非正式修改。最常见的基线如图所示'。

某个一旦成为基线,随即被放入项目数据库。此后,若开发小组中某位成员想要改动,首先要将它拷贝到私有工作区并在项目数据库中锁住,不允许他人使用。在私有工作区中完成修改控制过程并复审通过之后,再把修改后的推出并回到项目数据库,同时解锁。

建立基线的主要原因是重现能力、可追踪性和报告能力。重现能力是指返回并重新生成软件系统给定发布版本的能力。可追踪性是建立项目各种类型工件需求、设计、实现、测试等之间的横行依赖关系,其目的在于确保设计满足需求、代码实施设计以及使用正确代码编译生成可执行文件。报告能力来源于一个基线内容同另一个基线内容的比较,基线比较有助于程序调试并生成发布说明。

不可变的源对象经质量检查合格后所形成的新的相对稳定的配置成为软件版本。版本记录了配置项的演化过程。通常说的版本指的是已经交付给用户使用的产品的版本。在交付给用户使用以前,同样也会产生各种各样的版本。软件配置项在不同的时期,由于不同的要求会出现不同的组合,为了表明不同的特性,也会标明不同的版本。因此,每个软件配置项有一个版本组。他们彼此之间有特定的关系,这种关系用以描述其演变情况,通常软件配置项的版本组成树形结构。

### 软件配置管理的关键活动。

在软件配置管理中,必须考虑如下一个问题,怎样的配置管理活动可以解决诸如采用何种方式标识和管理已存在程序的各种版本、在软件交付用户前后如何控制变更、利用什么办法来估计变更引起的其他问题等等类似问题,这些问题可归结到软件配置管理中的活动上面。配置管理主要包括十个活动访问和查找软件、在开发生命周期中实现各种变更、在开发生命周期中迁移变更、管理编译和构建过程、管理变更的分布、获得批准和签署、管理软件变更请求、协调组间通信、获得项目状态、追踪错误和定位等。而对于这十项活动所完成的功能进行

分析研究,可总结软件配置管理的主要功能流程示意如图所示。根据图,对于流程各项活动的归纳分析,软件配置管理流程中的活动可归划到配置标识、版本控制、变更控制、状态报告和配置审计五大功能当中去,我们称这五部分为软件配置管理的关键活动。

配置标识—在系统演化过程中表示其中的中间软件产品。

版本控制—记录每个配置项的发展历史,并控制基线的生成。

变更管理—包括在整个生命周期中控制中间软件产品的变化。

状态报告—记录和报告软件的变化过程。

配置审计—用于保证软件产品是依照需求、标准和合同开发出来的。

配置标识

配置标识是软件生存周期中产生的所有文档的总称。在软件开发过程中,随着软件生存期的向前推进,软件配置项不断增多。为了方便对新增的,以及对原进行变更而形成的新的版本进行管理和控制,使之管理方便且有序,因此而为其命名。

配置标识定义配置项的名称、与其他配置项的联系、版本信息等。在对软件配置项进行标识时,不但要考虑每个软件配置对象固有的名字、描述、资源列表和实际存在四个部分,还需注意软件配置项之间的关系以及变更控制方面的内容。

配置标识的主要活动由选择配置项、制定配置项标识方案以及存取方案组成,有效的配置标识是其他配置管理活动的前提。配置标识包括识别软件系统的结构、唯一表示独立的组建并使其可以某种方式存取。配置标识的目的是获得在软件整个生命周期中识别其组件的能力,配置标识告诉我们系统中的配置是什么,文件的版本是什么。

综上,配置标识是配置管理的基础,是唯一地标识软件配置项和各种文档,使它们可用某种方式访问。配置标识的目标是在整个系统生命周期中标识系统的组件,提供软件和软件相关产品之间的追踪能力。

版本控制

版本控制是软件配置管理中不可缺少的功能。所有置于配置库中的元素都应自动予以版本的标识,并保证版本命名的唯一性。版本在生成过程中,自动依照设定的使用模型自动分支、演进。除了系统自动记录的版本信息以外,为了配合软件开发流程的各个阶段,我们还需要定义、收集一些元数据来记录版本的辅助信息和规范开发流程,并为今后对软件过程的度量做好准备。当然如果选用的工具支持的话,这些辅助数据将能直接统计过程数据,从而方便我们软件过程改进,活动的进行。对于配置库中的各个基线控制项,应该根据其基线的位置和状态来设置相应的访问权限。一般来说,对于基线版本之前的各个版本都应处于被锁定的状态,如需要对它们进行变更,则应按照变更控制的流程来进行操作。

变更管理是指在软件开发的整个周期中对软件的变化进行控制和追踪。从对于基线的定义中我们可以发现,基线是和变更控制紧密相连的。也就是说在对各个做出了识别,并且利用工具对它们进行了版本管理之后,如何保证它们在复杂多变的开发过程中真正的处于受控的状态,并在任何情况下都能迅速的恢复到任一个历史状态就成为了软件配置管理的另一重要任务。因此,变更控制就是通过结合人的规程和自动化工具,以提供一个变化控制的机制。

变更控制是软件配置管理的核心,它通过创建产品基线,在产品的整个生存周期中控制它的发布和变更,是在整个软件生命周期中控制对软件的变化。变更控制记录每次变化的相关信息。查看这些记录信息,有助于追踪出现的各种问题。记录正在执行的变化信息,有助于做出正确的管理决策。

软件的变更通常有两种不同的类型`功能变更和错误修改变更。功能变更是为了增加或者删除某些功能、或者为了改变完成某个功能的方法而需要的变更。这类变更必须经过某种正式的变更的代价比较小且对软件系统其他部分没有影响,或者影响很小,通常应批准这个变更。反之,如果变更的代价比较高,或者影响比较大,则必须权衡利弊,以决定是否进行这种变更。如果同意这种变更,需要进一步确定由谁来支付变更所需要的费用。如果是用户要求的变更,则用户应支付这笔费用否则,必须完成某种成本效益分析,以确定是否值得做这种变更。因此,功能变更涉及到软件额外费用的问题。

错误修补变更是为了修复漏洞而要进行的变更。它是必须进行的,通常不需要从管理角度对这类变更进行审查和批准。但是,如果发现错误的阶段在造成错误的阶段的后面,例如,在实现阶段发现了设计错误,则必须遵照标准的变更控制过程,把这个变更正式记入文档,把所有受这个变更影响的文档都作相应的修改。

在软件开发过程中变更是不可避免的,但是,变更的内容、变更的幅度都会直接影响到各项目,我们时时刻刻需要考虑到变更的波及面。例如,在一个瀑布模型的生命周期中,如果需求发生变化后,软件生命周期中各个阶段都有可能受到影响,这些阶段的所有相关的软件配置项都或多或少的受到影响。受到影响的各部位必须进行调整才能够应对这些变化所带来的冲击。

配置审计

软件的完整性实质是指开发后期软件产品能正确满足用户所提出的对软件的要求。而软件配置审计的目的就是要证实整个软件生存期中各项产品在技术上和管理上的完整性,同时,还要确保所有文档的内容变动不超出当初确定的软件要求范围,使得软件配置具有良好的可跟踪性,这是软件变更控制人员掌握配置情况,进行审批的依据'。

软件配置审计一般使用两种方法来确定变更得正确完成,即正式技术评审和软件配置审计。正式的技术评审在软件交付用户前实施。着重检查已完成修改的软件配置对象的技术正确性,评审者评价软件配置项,决定它与其它软件配置项的一致性。一般对所有的变更都要进行正式技术评审。软件配置审计作为正式技术评审的补充,评价在评审期间通常没有被考虑的软件配置项的特性。

状态报告

配置状态报告通过支持创建和修改纪录、管理报告软件配置项的状态或需求变化并审核变化来实现,它提供用户需要的功能,跟踪任意模式的软件项,提供完整的各种变化的历史版本和汇总信息。配置状态统计涉及记录和报告变更过程的状态,其目的是为了持续地记录配置状态以及保持基线产品和其变更建议的历史,并使相关人员了解配置和基线状态。当一个软件配置项标识更改,或变更控制审核人批准一次变更,则生成一个配置状态报告。它包含在整个软件生命周期中对基线所有变更的可跟踪性报告,主要描述配置项的状态、变更的执行者、变更时间和有何影响。配置状态报告的结果存入数据库,管理者和开发者可以查询变化信息,也可以评价变更。软件工程师可以看到都作了哪些修改或者每个文件都包含在哪个基线中,项目经理可以跟踪详细的问题报告和各种其他维护活动最简单的报告,应包括事务日志、变更日志、配置项增量报告等。配置状态报告对于大型软件开发项目的成功起着至关重要的作用,它提高了所有开发人员之间的通信能力,避免了可能出现的不一致和冲突。

## 信息检索技术

信息检索技术是一个比较老的概念，指的是根据给定的查询项在一个数据集合上查找出相似的、相关的内容[23]，最后将结果按照相关度排列。对于信息检索，大部分的工作是基于文本检索，即查询项和待查询集合的内容都是以文本形式存储，主要方法可以分为两类：基于统计检索和基于语义检索。

### 基于统计检索

基于统计的方法是根据查询语句和待查询集合的内容的统计结果来确定查询结果。将查询和待查询集合放在同一个统计空间中，对文本进行分词，将所有的内容基于统计信息进行表示，然后计算得到与查询项相关的内容。当前常用的基于统计检索的方法包括向量空间模型和概率模型。

向量空间模型把文本信息用向量空间中的点表示，然后计算向量之间的距离来量化两个文本之间的相似度，最后通过对计算结果排序得到查询结果。文档向量构造最简单的方法是使用词袋模型（Bag of Words，BOW），即文档向量的长度是数据集中所有的词的数量，文档向量每一维的值为当前词在该文档中的出现频率或其他的一些特征，从而将一个文档表示为一个向量。在计算文本间距离时，可以使用余弦公式等方法，得到两个文本之间的相似度值。还有一些方法通过对关键词加权重的方式表示文本向量[24]。

潜在语义索引也是基于向量空间的检索方法。LSI使用one-hot方法表示文档，并将所有文档放在同一个矩阵中，通过奇异值分解（Singular Value Decomposition，SVD）的方法对矩阵降维，并且得到文本的主题和主题之间的关系。但是由于使用向量表示之后矩阵规模巨大、同时矩阵会过于稀疏，因此需要进行降维操作。LSI方法虽然表示和计算过程都比较简单，但是也有如下几个缺点：（1）通过SVD生成的新矩阵解释性较差；（2）无法解决“一词多义”的问题；（3）和词袋模型一样，忽略了文章中单词的先后顺序。

概率模型基于多个随机变量之间概率关系计算查询项和候选文档之间的关系，将查询项与文档集合中的文档同时出现的概率定义为二者的相似度，形式化表示如公式(2.1)所示。在计算完成后，对相似度进行排序，然后设置阈值过滤得到某个查询对应的相关内容，从而完成信息检索任务。

（2.1）

其中表示文档集合中第*i*个文档，表示查询语句，公式(2.1)使用贝叶斯公式对二者同时出现的概率进行求解。

### 基于语义检索

基于语义的方法是在一定程度上通过对查询句子进行句法和语义的分析，即通过其他的表现形式对文本的语义进行表示。由于当前大部分信息检索都是基于文本检索，因此基于语义的信息检索也和自然语言处理技术相关。使用句法分析的方法是借助文本的句式和一些词性信息对文本语义进行强化处理，如在文献[25]中提到的使用子树加速对文本相似度的计算，并将子树与文本特征向量进行比较匹配，从而得到更准确的文本相似度；使用语义分析的方法有基于本体[26]和基于外部语义词典（如Hownet[27]和Wordnet[28]）等方法，这些方法基本是将查询文本中的单词或词组抽离出来，然后用本体或者外部语义词典中的含义来表示，从而将整个文档集合中的内容统一表示，然后使用给定的算法计算相似度，从而确定检索列表。

## word embedding介绍

Word2vec是Google的Tomas Mikolov[29]等人提出的文本表示方法，将单词表示成word embedding的形式，是一种分布式表示模型。Word embedding的向量表示反应了词的共现关系。在给定的语料库下，通过算法模型将词表示成向量的形式，向量的维数可以由用户指定，通常为几十到几千不等，因此可以远远小于词典的大小，这样就避免了向量空间模型中的矩阵稀疏问题。同时，Mikolov等人还开发出了训练词向量的工具，训练效率非常高，非常适合在现在大数据集的应用中。

Word embedding基于分布式假说提出，即出现在相同上下文中的词的含义相近[30]。训练word embedding的流程类似前馈神经网络，但是仅包含输入层、隐藏层和输出层三层结构，并且由于Mikolov等人对隐藏层和输出层做了优化，使得模型的训练效率更高。训练模型主要包括skip-gram和CBOW两种，skip-gram模型的输入是特定词的词向量，输出是特定词上下文相关的词的词向量，而CBOW和skip-gram的输入输出相反。skip-gram和CBOW模型分别如图8和图9所示。



图8 skip-gram模型示意图



图9 CBOW模型示意图

## 软件工程领域学习排序算法

学习排序（Learning to Rank， LtR）算法作为一种监督或半监督的机器学习算法[31]，在信息检索、数据挖掘和自然语言处理等领域起着重要的作用。在软件工程领域，学习排序算法经常用在故障定位和数据重复检测等方面。对于每个查询与候选文档的组合，我们能够抽取出若干特征，比如词汇相似度、语义相似度等，作为机器学习模型的输入。学习排序的流程如图10所示，更加形式化的表示如公式(2.2)和公式(2.3)所示。

(2.2)

(2.3)

其中表示查询语句，表示相应的候选文档集合，函数将查询-文档对映射为特征向量，是表示每个特征向量的权重矩阵，是排序方法。排序方法有三种类型：单文档方法（Pointwise）、文档对方法（Pairwise）和文档列表方法（Listwise）。

单文档方法的处理对象是单独的一篇文档，将其转化为特征向量后，该问题转化为分类或回归问题，可以使用等级回归或分类算法来解决问题。该方法只考虑单个文档的绝对相关度，忽略了文档间顺序关系。文档对方法则考虑了文档间的关系，把任意两个文档组成的文档对（文档A，文档B）作为机器学习的输入，最后得到文档A是否应该排在文档B的前边，此时学习排序算法可以转化为二分类问题。该方法通过考虑两两文档对之间的关系进行排序，因此比单文档方法的效果有所提升。文档列表方法的处理对象是每个查询所对应的所有搜索结果的列表，该方法直接训练优化算法模型输出的整体序列，因此其结果能够更接近真实的文档序列。已有的且常用排序算法如表1所示。



图10 学习排序的流程

除了排序算法，选择合适的特征作为模型的输入对学习排序算法的性能也有很大的影响。在软件工程领域，大多数使用学习排序算法的应用都选择词或者文本的相似度作为其中一个特征，比如文献[32]中作者使用了词表面信息相似度、通过API增强的词相似度和类名的相似度，文献[33]中作者选择文本相似度和上下文相似度作为学习排序算法的特征。值得一提的是，在计算以上相似度时，作者们都是用了余弦相似度。除此之外，特征可以分为两类：依赖查询的特征和不依赖查询的特征[33]，其中依赖查询的特征即与查询文本相关的特征，如相似度特征；不依赖查询的特征则与查询文本无关，仅仅反应候选结果的特点，如候选结果的词频[33]、故障修复频率[32]等。同时，在选择特征时，特征数量不应过大或过小，过大会导致过拟合问题，过小则会降低结果精确度。

表1 排序方法的常用算法

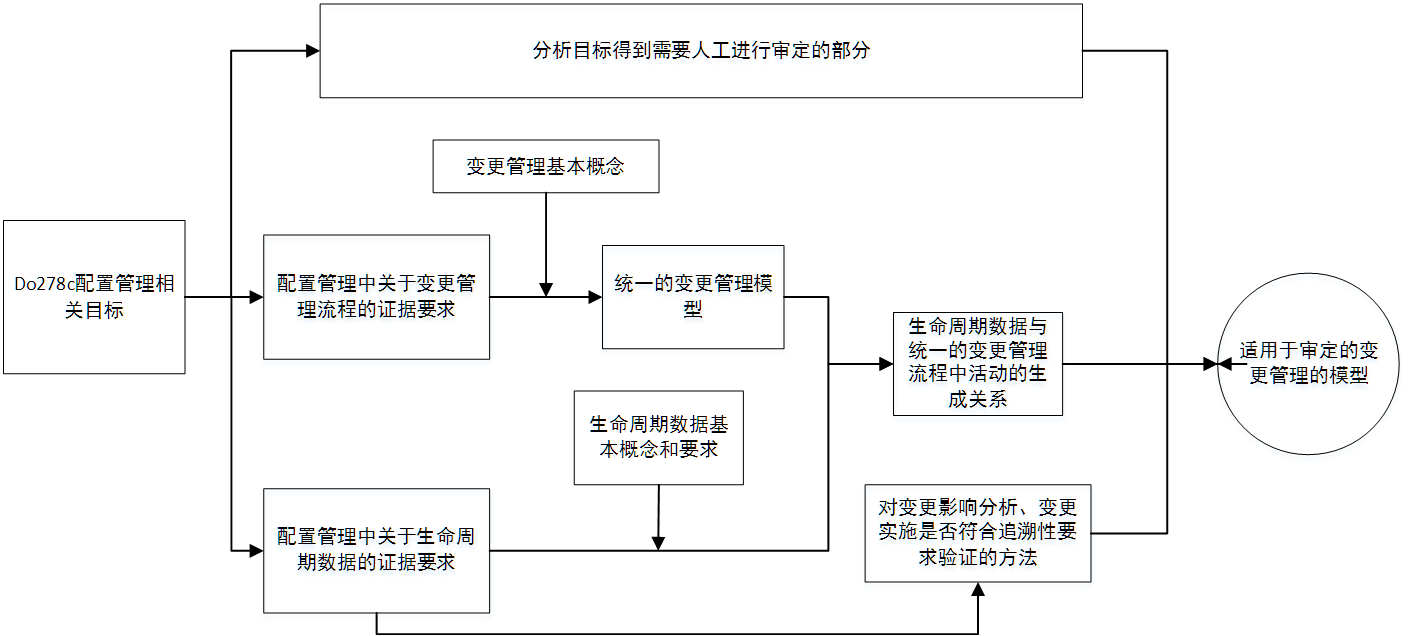
|  |  |
| --- | --- |
| **排序方法分类** | **常用算法** |
| 单文档方法 | 回归：Subset Ranking[34] 分类：SVM、McRank[35] |
| 文档对方法 | IR SVM[36]、Ranking SVM[37]、RankBoost[38] |
| 文档列表方法 | ListNet[39]、AdaRank[40]、SVM Map[41] |

# 基于DO-178C标准研究，建立适航领域配置管理过程审定模型

本章首先对使用GSN图符对D0178c中配置管理过程部分目标进行了分析，接下来提出了变更管理的审查中需要的生命周期数据及格式要求，然后提出了统一的变更流程，并确定了生命周期数据和变更流程的输入输出关系。并根据上述研究建立适航领域配置管理过程审定模型

## 建立适航领域配置管理过程审定模型的流程

本研究提出了通过对DO178c目标进行分析，结合变更管理的基本概念，以及DO178c中有关生命周期数据的概念和规定，建立了统一的变更管理流程，并 确定了生命周期数据和变更流程的输入输出关系 ，最终得到适航领域配置管理过程审定模型，对配置管理过程自动审定模型建模流程如下图所示。



## 使用GSN方法对目标进行分析

GSN是一种可以明确说明任一论证过程中存在的元素机器相互关系的图形化论证符号语言。运用GSN语言来表述论证过程可以有助于形象、具体、合理地说明系统、运营肌组织的安全性能。运用GSN语言完成的论证模型主要是用来表述安全子目标是如何支持顶层的安全目标的，下一层的安全子目标又是如何支持该层安全子目标的。

这与在Do178c标准中，目标、过程、与数据间的关系是相似的，通过适当的变换，所以可以利用GSN，对do178c标准进行具体分析，找出相关目标的过程要求、子过程及存在的证据间的对应关系。

将Do178c标准所要求的目标对应于GSN中的安全目标，生命周期数据或数据元对应GSN中的安全证据，而标准中对过程要求目标可以作为论证策略的一部分。其中将Do178c提供的过程指导作为策略或是论点的一部分在分析过程中得到证明，能够降低对整个论证过程进行分析的复杂性

DO-178C标准附录A中，DO-178C对配置管理过程中要满足目标，在DO-178C附录A的表7中有详细的说明，包括6个目标，分别是：

标识构型项（A-8.1）

建立基线和可追溯性（A-8.2）

建立问题报告，变更控制，变更评审和构型状态纪实机制（A-8.3）

建立归档、检索和发布机制（A-8.4）

建立软件加载控制机制（A-8.5）

建立软件生命周期环境控制（A-8.6）

本文选取a-8.1 ,a-8.2 ,A-8.3 三个目标作为研究对象，依据标准 进行了详细的解读，确定了各个目标的论证结构（包含证据）。进行详细解读的依据包括 标准中对过程目标的解读，审查指南，具有审查经验的人员的指导等。

在对目标使用GsN方法进行分析时，使用了两种大的策略：

策略1.第一次将大目标分解时，主要以“标准、流程、经验、审查指南”为依据，将Do178c要求的配置管理过程目标按子过程相关要求分解为支持目标论证的多个子过程目标。

策略2：为了构建符合Do178c标准的变更管理模型，在对子过程的目标进行自上而下的论证时，以“标准、流程、经验、审查指南”为依据，分析得出为满足子过程目标需求的证据需求，并分类：

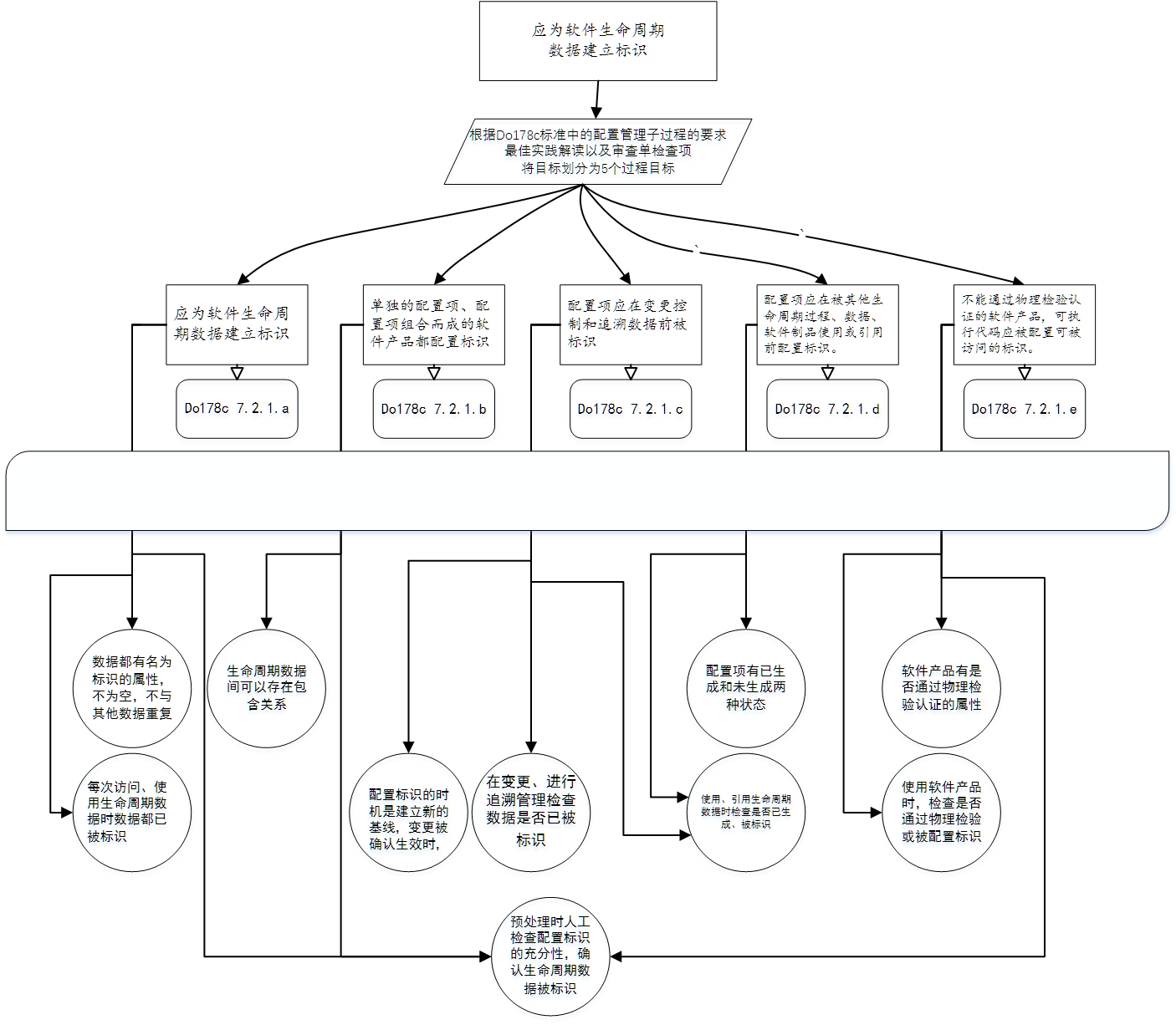
1在数据初始化和论证过程中需要人工辅助的证据要求。

2.主要对生命周期数据自身相关的证据要求。

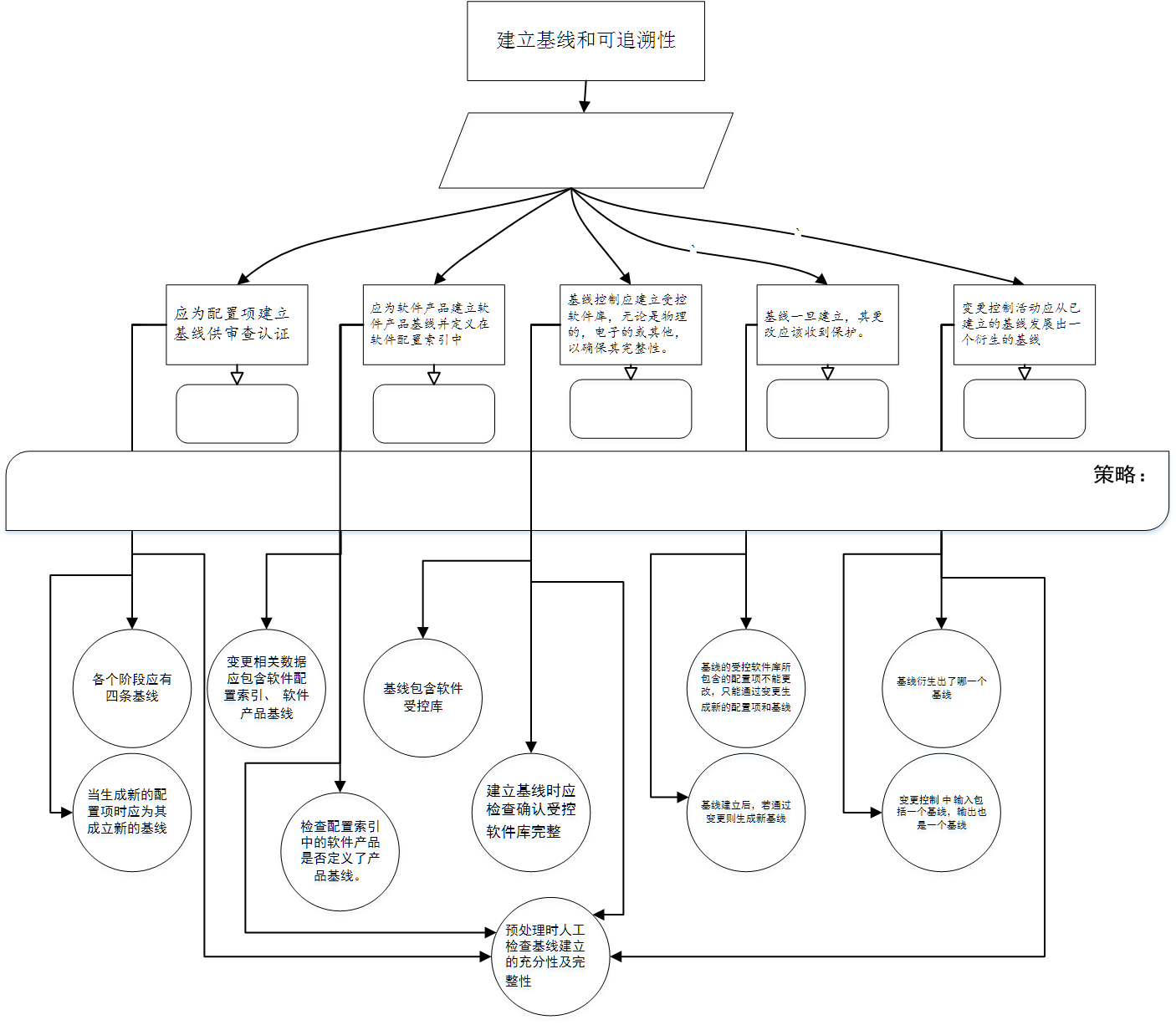
3与变更管理的流程要求相关的证据要求。

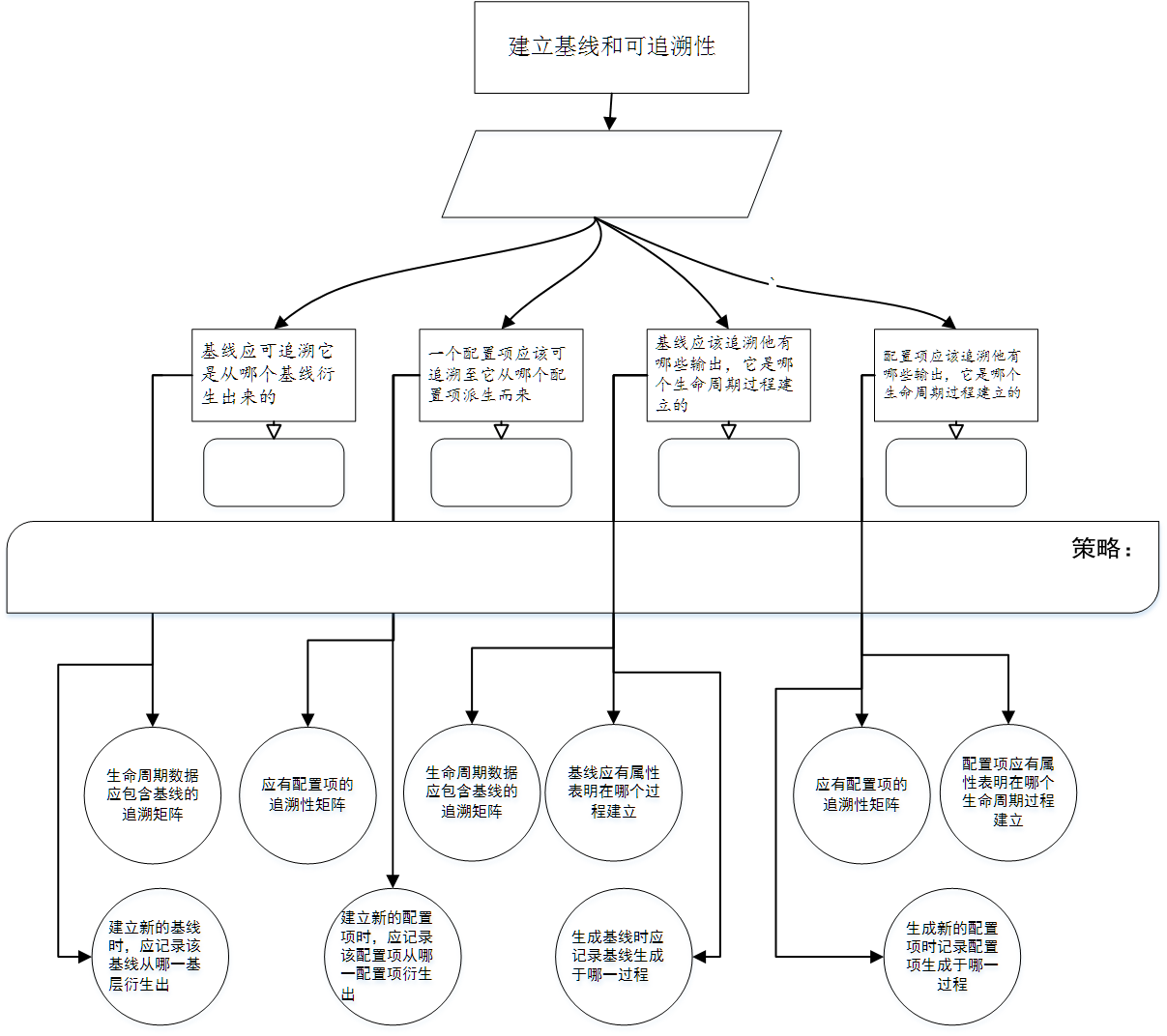
### 对目标 A-8.1使用GSN方法进行分析，

具体如图（）

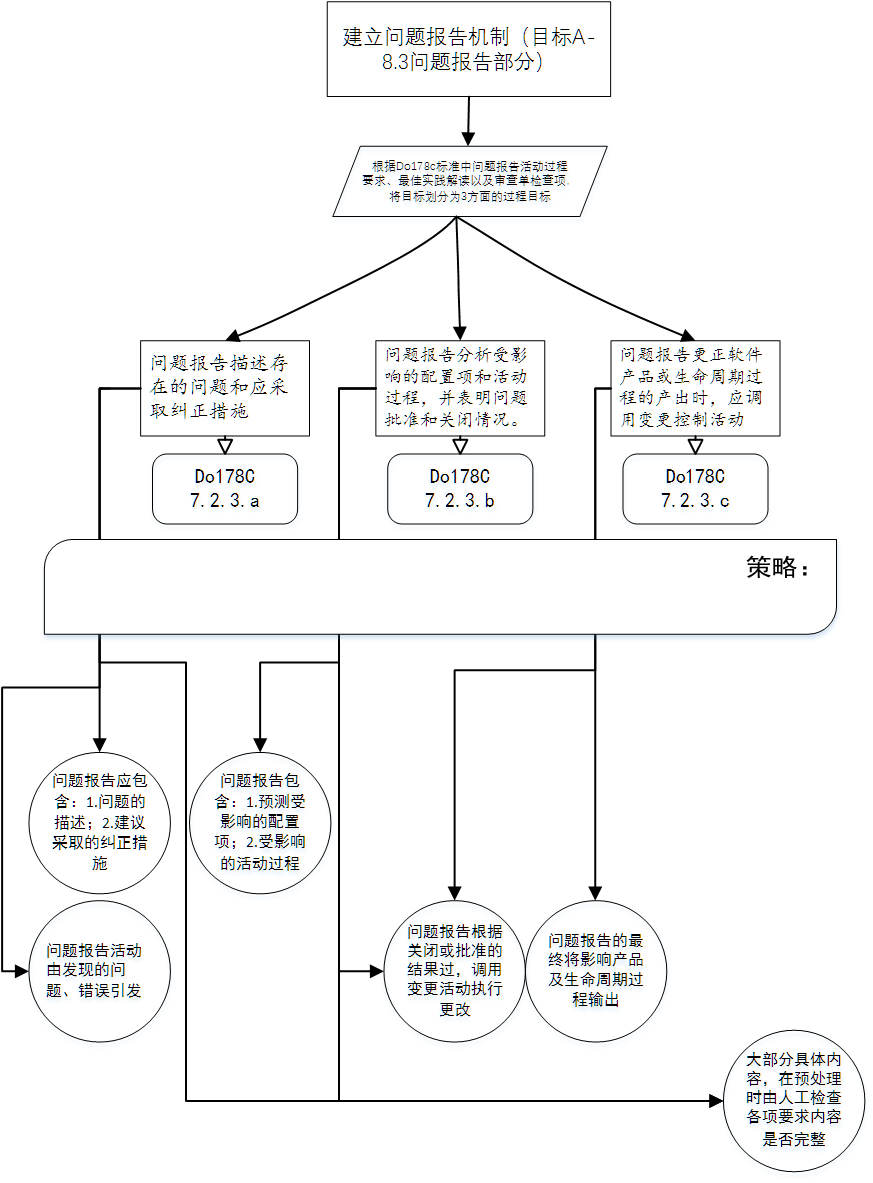


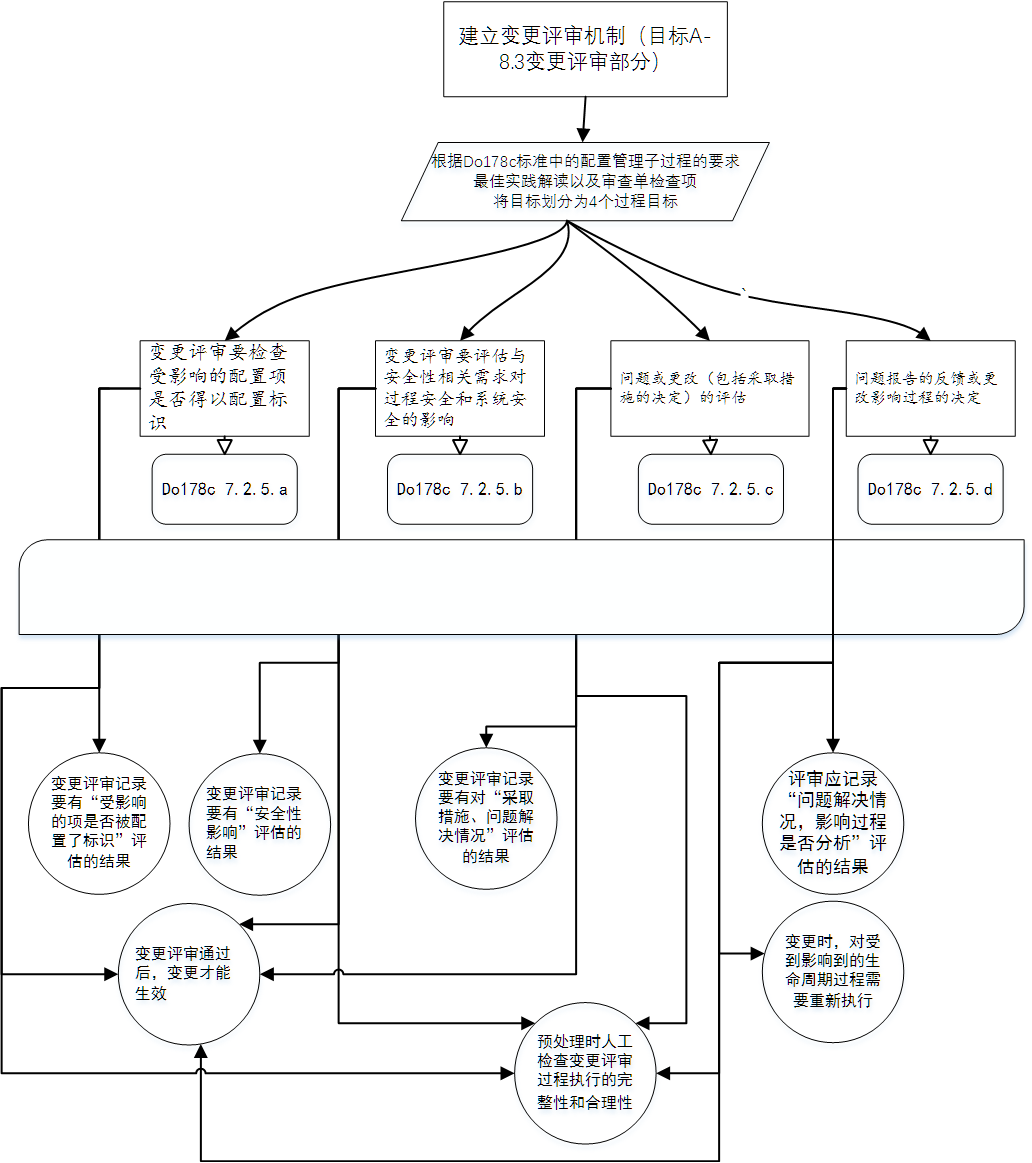
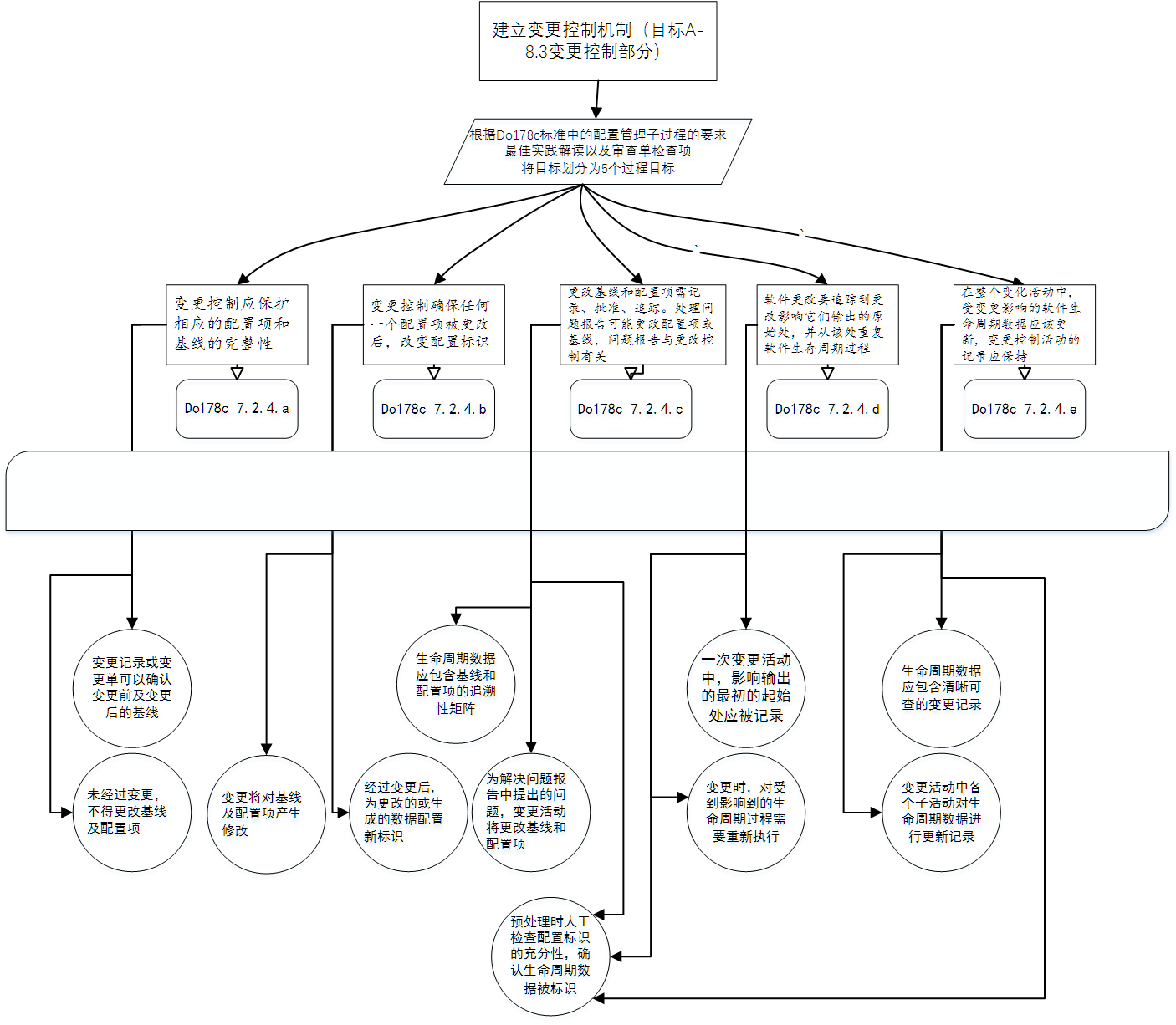
### 对目标 A-8.2使用GSN方法进行分析，





### 对目标 A-8.3使用GSN方法进行分析，





## 生命周期数据及格式要求

生命周期数据、最佳实践提出数据项与数据元

### 标准要求的生命周期数据

在Do178C标准中，列举了22种生命周期数据（错误!未找到引用源。），按照标准要求，所有标准要求的软件生命周期数据都应该被标识。也就是说开发过程中至少应该将这22种生命周期作为配置管理的标识项。同时Do178C标准也提出了其它需要作为配置项进行管理的要求，因此22种配置项不是配置项的全集。

在实际的适航软件开发中，目前大部分的生命周期数据材料都是以文本的方式提供的，这使得直接使用该生命周期数据材料进行自动审定的工作是难以开展的，即使是由专家进行审定，也必然由于数据庞杂且不规范，提高了审定的难度，也降低了审定效率和效果。

|  |  |
| --- | --- |
| **数据项名称** | **数据项名称** |
| 软件方面的验证计划 | 可执行目标代码 |
| 软件开发计划 | 软件验证例子和程序 |
| 软件验证计划 | 软件验证结果 |
| 软件配置管理计划 | 软件生命周期环境配置索引 |
| 软件质量保证计划 | 软件配置索引 |
| 软件需求标准 | 问题报告 |
| 软件设计标准 | 软件配置管理记录 |
| 软件代码标准 | 软件质量保证（SQA）记录 |
| 软件需求数据 | 软件完成综述 |
| 设计说明 | 数据追踪 |
| 源代码 | 参数数据项文件 |

### 最佳实践提出的数据项与数据元

为了解决这个问题，根据最佳实践提出的数据项管理的要求，将管理的配置项分为了数据元与数据项，数据项实质是生命周期数据被标识为配置项后，对配置项进行分类的结果，是为了方便对生命周期数据进行管理而进行的细化。数据元则是更为喜欢的生命周期数据元素。最佳实践识别的数据项见表（）。

|  |  |
| --- | --- |
| 数据项编号、名称 | 数据项编号名称 |
| D-01 项目进度计划 | D-29软件需求数据（条目化） |
| D-02 分配到软件的系统需求 | D-30高层需求与系统需求的追踪数据 |
| D-03软件等级 | D-31软件架构（非条目化） |
| D-04项目估算数据 | D-32底层需求层次结构 |
| D-05项目软件生命周期 | D-33底层需求（条目化） |
| D-06项目管理计划 | D-34软件设计说明（非条目化） |
| D-07软件合格审定计划 | D-35底层需求与高层需求的追踪数据 |
| D-08软件开发计划 | D-36源代码 |
| D-09软件验证计划 | D-37源代码与底层需求的追踪数据 |
| D-10软件配置管理计划 | D-38测试用例 |
| D-11软件质量保证计划 | D-39测试用例与高层需求的追踪数据 |
| D-12软件需求标准 | D-40测试用例与底层需求的追踪数据 |
| D-13软件设计标准 | D-41高层需求测试覆盖数据 |
| D-14软件编码标准 | D-42底层需求测试覆盖数据 |
| D-15核查检查单 | D-43测试规程 |
| D-16核查记录 | D-44测试规程与测试用例的追踪数据 |
| D-17问题报告 | D-45软件生命周期环境 |
| D-18评审检查单 | D-46目标代码 |
| D-19评审报告 | D-47可执行目标代码 |
| D-20软件配置索引 | D-48编译、链接、加载日志 |
| D-21基线 | D-49测试结果 |
| D-22变更请求 | D-50测试结果与测试规程的追踪数据 |
| D-23软件配置状态报告 | D-51测试结构覆盖数据 |
| D-24软件配置管理记录 | D-52目标代码与源代码的追踪分析数据 |
| D-25软件生命周期环境配置索引 | D-53加载控制记录 |
| D-26软件质量保证记录 | D-54软件完成综述 |
| D-27高层需求层次结构 | D-55软件符合性评审报告 |
| D-28高层需求（条目化） |  |

表 1 最佳实践识别的数据项

### 变更流程关键数据项及数据元

为了能够对变更流程中的适航符合性要求进行验证，本文根据上一小节的分析结果，对变更流程中需要使用到的数据项进行了筛选与分类，并对需要数据元级别数据信息的数据进行了数据元的确定与分类。

根据为了便于后期研究的使用（需求程度不同的生命周期数据，需要细化的数据元不同），

分类：1.配置管理记录相关等。确定用于多个变更执行时的相互顺序

2.变更管理流程审定相关的关键数据。

3.变更管理变更内容符合性进行审定的数据。

4.其他受控生命周期数据

5.配置管理纪实发布的信息等

具体数据项及数据元分类及需求见表（）。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 数据项 | 数据元内容 | 数据分类 |
| 1 | 软件配置管理记录 | 1.软件配置记录的标识;  2.条目化的变更记录，每此变更记录包含以下信息：a.变更申请时间;b.变更审批的时间;c.变更审批的结果（批准或驳回）;d.变更批准或驳回的时间;e.变更完成或终止时间;f.变更的最终状态;g.被变更基线的标识;h.变更完成后生成基线的标识 | 变更活动全流程相关关键数据 |
| 2 | 基 线 | 1.基线的标识;2.基线的受控配置项库;3.基线建立的时间;4.与其他基线间的衍生关系。 | 变更流程相关关键数据 |
| 3 | 问题报告 | 1.数据项标识;2.创建时间;3.问题描述  4.变更影响分析;5.受影响的配置项或数据元的标识;6.软件发生错误的起始配置项（若是需求变更则是被变更的需求的标识）;7.建议的变更和行动 ;8.问题报告提出者 |
| 4 | 变更请求 | 1.数据项标识;2.变更请求提出时间;3.变更影响分析;4.受影响的配置项或数据元标识;5.软件发生错误的起始配置项（若是需求变更则是被变更的需求的标识）6.建议采取的变更和行动;7.变更请求提出者 |
| 5 | 变更审批单 | 1.数据项标识;2.对变更的批准或拒绝意见;3.变更审批或拒绝的时间;4.负责审批的人员 |
| 6 | 变更评审单 | 1.变更评审单标识;2.评审的最终结果;3.评审生效时间;4.评审负责人员 |
| 7 | 软件配置状态纪实 | 已软件配置状态纪实报告的形式，应包括：  1.配置项标识;2.基线列表;3.配置项列表;4.问题报告列表;5.变更申请、审核、评审记录列表 |
| 10 | 高级需求 | 1.配置项标识;  2.需求数据具体类型;  3.条目化的数据内容 | 需求或需求派生  配置项数据 |
| 11 | 低级需求 |
| 13 | 源代码 |
| 14 | 测试用例 | 1.配置项标识;  2.需求派生数据具体类型;  3.条目化的数据内容 | 需求或需求派生  配置项数据 |
| 15 | 测试规程 |
| 16 | 目标代码 |
| 17 | 可执行目标代码 |
| 18 | 测试结果 |
| 19 | 测试用例与高层需求的追踪数据 | 1.配置项标识;  2.所属追溯性数据的类型;  3.条目化的数据内容 | 追溯性相关  配置项数据 |
| 20 | 底层需求与高层需求的追踪数据 |
| 23 | 源代码与底层需求的追踪数据 |
| 24 | 测试用例与底层需求的追踪数据 |
| 26 | 测试规程与测试用例的追踪数据 |
| 27 | 测试结果与测试规程的追踪数据 |
| 29 | 目标代码与源代码的追踪分析数据 |
| 30 | 其他受控生命周期数据 | 1.配置项标识;2.生成数据项时所处的生命周期过程;3.配置项间的追溯关系 | 其他配置项数据 |

确定配置管理过程生命周期数据配置项或数据元级别的需求，能够为建模审定的模型提供对生命周期数据形式和内容的需求，而且对于形式随意，难以使用的研发方提供的生命周期数据，既不符合标准关于提供易于使用的生命周期数据的要求，更加难以应用于审定的工作，因此，研发方也可参照此数据项及数据元要求，对生命周期数据进行记录与归档整理时，尽量做好分类与标注工作，为审定提供便于使用的生命周期数据。

## 统一的变更流程

本节主要根据GSN分析目标所得出底层目标中对变更流程相关的目标要求，提出统一变更流程。

根据3.1节对Do178c的分析结果，特别是对变更流程相关证据的需求作为依据，结合2.n节变更管理的基本概念和一般过程，提出一种用于适航审定的变更管理的流程，在确定该流程的过程中，考虑了两方面的因素。第一，该流程必须包含Do178c目标及过程中所涉及到的变更管理的流程要素及生命周期数据的需求。否则不符合标准要求。第二，该流程应该尽量简洁。作为审定用流程，应面向实际开发工的，应在要求其符合标准目标的基础上，不对开发流程提出与标准中目标无关的要求和活动。审定用流程过于复杂，会导致对符合do178c标准要求，但与该复杂流程不兼容的研发流程无法进行审定；第三，该流程强调的是过程而非方法，与使用何种变更管理工具无关。在开发过程中，配置管理或变更管理使用的工具种类较多，各有特点，过多的工具要求必然给开发造成更多困难且不是必要的，只要流程相符即可。

统一的变更模型的流程图如图（）所示：

一个图

按照变更子流程的特点，本文将变更流程分成了5个阶段，a.变更申请阶段 b.变更评估阶段 c.变更实施阶段 d.变更评审阶段 e.变更完成阶段 f.状态统计阶段。整个流程共9种子活动，分11步进行。

1．问题报告或需求变更报告。按照变更原因的不同，可以将变更分为两类，即问题引起的变更和需求变化引起的变更。若变更因问题而提出的，则由相关负责人展开问题报告活动，填写问题报告单；若变更时因为需求改变引起的，则由相关负责人进行需求变更的分析。

2．变更申请。变更申请人根据问题报告或需求变更报告，提出变更申请。。

3．变更审批。由审批负责人员对变更申请的内容进行审核，若审核通过，则开始执行变更；若审核不通过时，则重新申请，或中止变更，关闭申请和问题报告。

4．基线管理。根据变更审批明确的要被变更的原基线的标识，获得该基线的受控配置项的标识，该基线状态根据配置管理计划是否需要被checkout，取决于是否允许同时对同一基线开始两次以上的变更，一般来说应是不允许的。

5．配置项：通过基线受控配置项库中配置项的标识，获得配置项数据。并将此配置项checkout。

6．数据开发过程。由开发人员执行数据的更改。需要说明的是，根据最佳实践关于建立阶段基线的要求文献（），在开发阶段，完成一个子阶段后，需要建立核查基线、评审基线以及审查基线，因此开发活动过程中也会建立基线，但这些基线是在处理人角色发生变化时建立的，不是由变更建立，与变更生成的基线不同，此处不对其生成过程进行具体说明。

7.配置标识。为新生成的数据配置标识，包括开发过程生成的阶段基线和基线的受控数据项。对于阶段基线，虽与变更生成的基线不同，但由于可能作为未来变更活动的初始基线，因此要将这些基线及基线中新生成的数据项作为基线类型的数据。因此实际在第6步数据开发过程中，也需进行配置标识和基线管理活动。

8.变更评审。再由变更委员会进行变更评审，若评审通过，则建立新的基线，若不通过，则返回第6步数据开发过程重新执行。

9.基线管理：变更评审通过后，确定了基线的受控数据项库，建立新的基线。

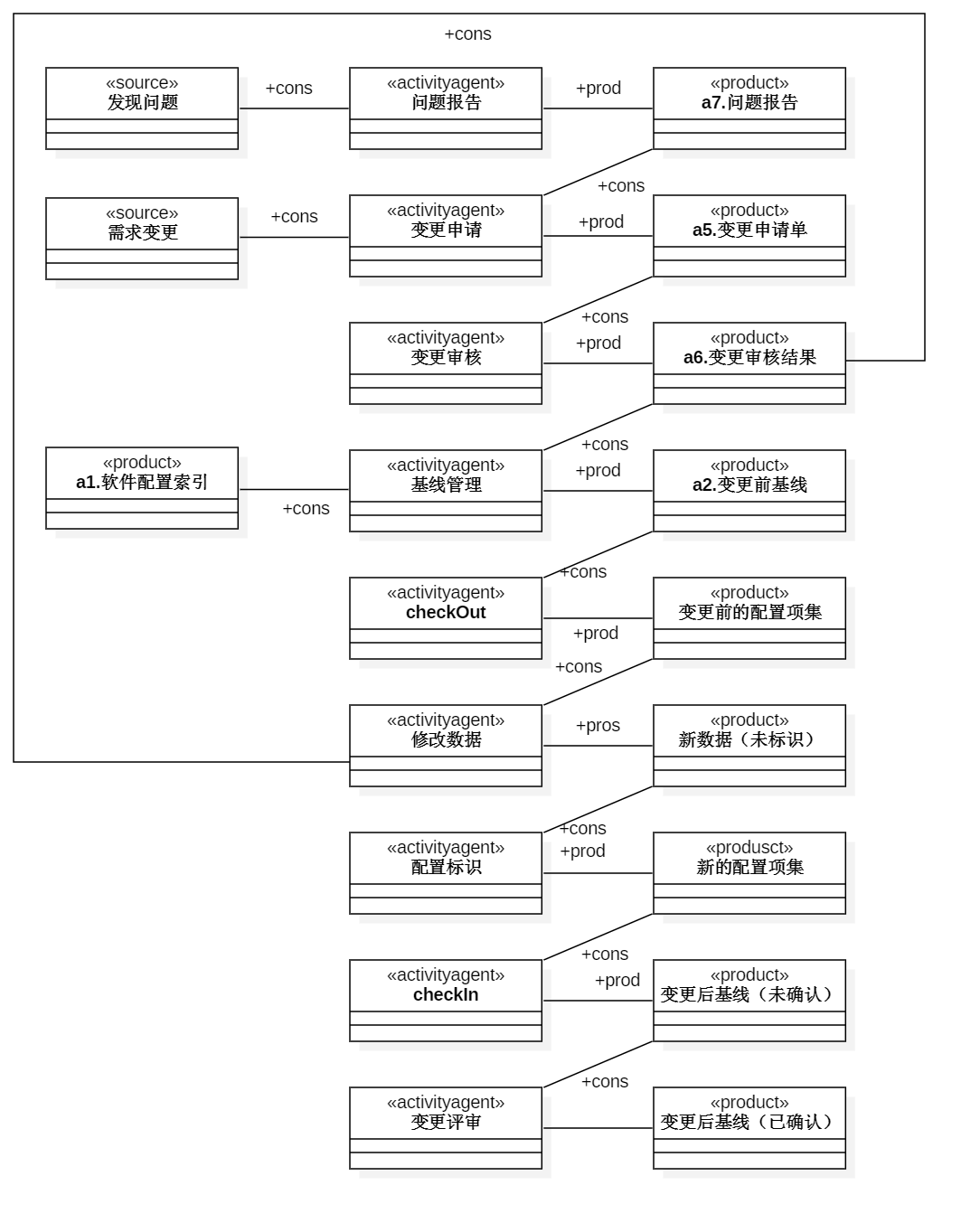
10.配置标识：为建立的新基线配置标识。同时基线已建立成功并配置了标识，将第4、5步检出(checkout)的基线和配置项进行检入(checkIn)。

11.配置状态纪实统计：将新基线加入基线列表，新配置项加入配置项列表，问题报告加入问题报告列表，变更申请、审核、评审记录加入变更报告列表。

## 生命周期数据和变更流程间的输入输出关系

在章节3.3、3.4中，我们确定了变更管理中生命周期数据项、数据元，提出了统一的适用于审定的统一的变更管理模型。适航审定的基本思想，是通过对研发阶段的生命周期数据进行审查，判断研发过程是否满足了Do178C标准的要求，因此，本节确定变更流程中生命周期数据与变更子活动间的输入输出关系。

图() 展示的是各项生命周期数据和统一的变更流程间的输入输出关系。Cons代表consume，表明活动的输入，prod代表produce，表明活动输出。中间一列为活动代理类，代表了活动的子流程，source类 引起变更的原因，product代表了生命周期数据，子活动流程自上而下依次执行。



数据与流程关系图

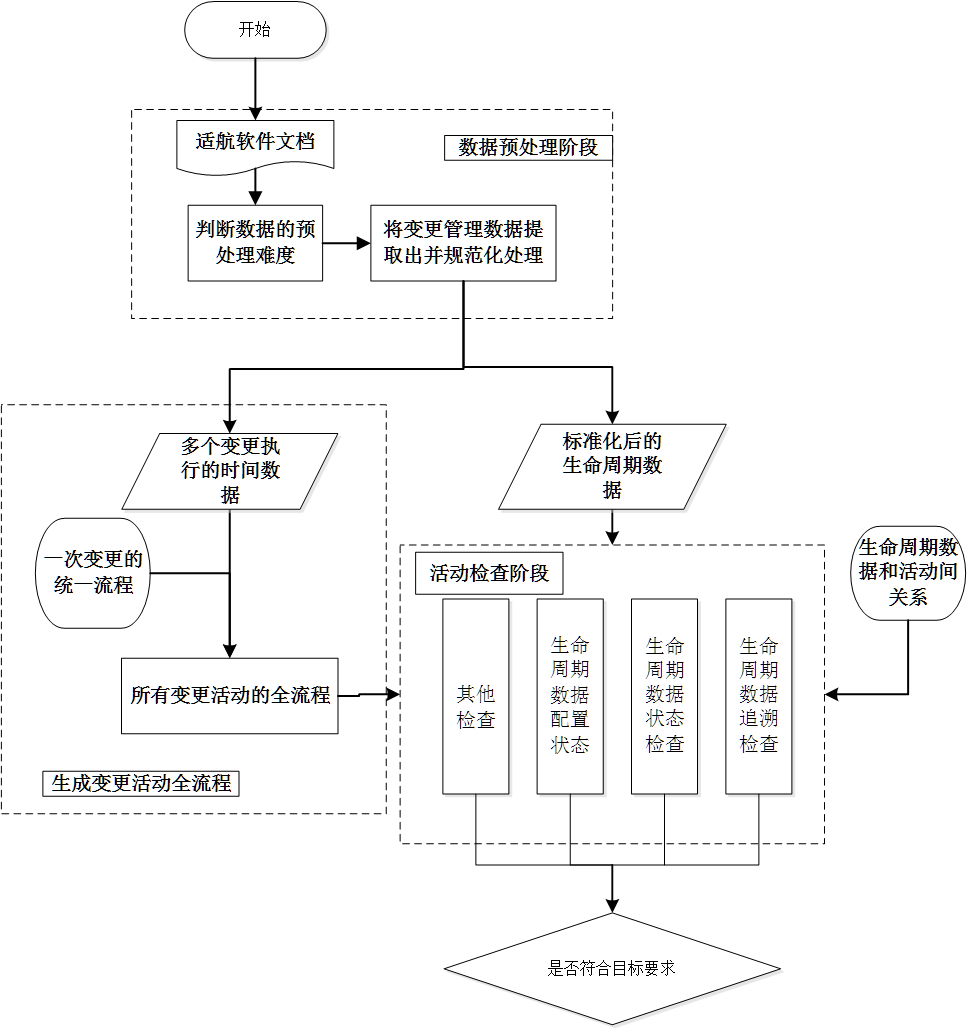
# 第四章 适航领域变更管理审定模型

本章以第三章的研究成果为基础，提出了适航领域变更管理审定模型。首先介绍了变更管理审定模型的整体框架SSSR，然后对使用审定模型三大部分及其应考虑的问题和具体实施方法步骤进行了详细描述。

## 审定模型

本文的第三章研究了变更管理生命周期数据项数据元要求、适用于审定的统一的变更管理流程以及生命周期数据与变更流程子活动间的输入输出关系，我们以此为基础，提出了适航领域变更管理审定模型。按照审定将审定分为了几个部分。分别是人工审查，流程审查和变更内容审查。人工审查是对需要人工参与进行审查部分的界定与说明；流程审查部分通过将生命周期数据代入变更活动全流程，按照统一变更流程要求、生命周期数据格式要求、输入输出关系要求，对流程相关的数据符合性要求进行审查；变更内容审查则是通过重建需求相关数据项追溯链接，对变更影响分析的准确性和变更实施时正确定进行审查。

审定模型的框架如图（）所示。



适航领域变更管理审查模型

## 人工审查

人工审查部分包含预处理时审查与监督审查两部分。

在预处理时，审查人员根据研发方提供的生命周期数据形式、精细程度、处理难度等，对生命周期数据进行初步判断，确定此生命周期数据的基本格式和要求是否符合适航要求，若符合，则对该原始生命周期数据进行预处理，得到标准化后的数据，同时也对预处理时需考虑部分进行人工审定。该部分审查的内容包括各项生命周期数据的充分性、完整性等资料预处理之后无法进行审定的内容。

另外，由于适航审定是一个极为复杂的工作，对于能够由机器审查的部分，机器审查的同时也需要人工参与进行监督审查。

## 流程审查

### 变更活动全流程

在上一章中，我们确定了统一的变更流程，将单次变更中的生命周期数据代入到统一的流程中，即可重现此次变更的具体流程。通过重现每次的变更流程，我们可以对变更过程中的先后关系、协作关系、数据的传递关系及资源的共享关系等是否符合要求进行验证。但如果仅进行此项验证，则忽略了多个变更间存在的这些关系，因此还需构建整个变更活动的全流程。

构建全流程的基础是配置管理记录，在生命周期数据中我们已经确定配置管理记录至少应提供变更的时间信息、关键标识项的标识以及变更的最终结果。因此，各次变更相应的问题报告、变更申请和变更审核结果、变更前及变更后的基线标识等可以由配置管理记录取得。Do178c并没有明确规定配置管理记录的具体内容及形式，在制定配置管理计划时，可以自行确定易于记录和使用的管理方式。

基于这些信息，我们可以将每个变更按照变更执行的顺序，放置于统一的时间线上，从而对所有的变更活动按照研发工作中变更的时间节点排序，得到了变更活动的全流程。代入相应的变更相关生命周期数据，多个变更的相互关系即可通过流程重现得到验证。

### 变更子活动审查

由上一节的方法，我们已经得到了变更活动的全流程。本章介绍的是通过将生命周期数据代入到各个子流程中，模拟研发过程对变更流程进行重现，对流程相关的生命周期数据是否符合DO178c的相关目标进行审查的方法。

在重现变更流程时，我们以开发时各项步骤的具体活动为依据，规定相应步骤的子活动在重现过程时的活动模拟和数据验证。

活动模拟的任务是模拟研制过程，并据此改变数据项相应状态，为数据验证提供依据。我们规定，审查时为数据项设置相应的状态属性，通过检查状态属性应能够确定相应问题报告或变更申请单在此时间点上是否已提出，变更审核单是否已审核完毕，各数据项是否已生成、已标识或已检出。

数据检查验证通过对相应数据项的状态属性进行检查，判定此过程中的生命周期数据是否存在矛盾、冲突、缺失等情况。例如，标识为C1变更的批准生成了配置项n1 ,变更完成时，会将配置项n2状态改变为已生成、已标识，而标识为C2的变更，对基线 B1 进行变更，基线B1的被控库中包括配置项n1，此时变更C2会检查配置项n1是否已生成、已标识。若变更C1的完成时间早于变更C2的批准时间，则此项检查无问题；若变更C1的完成时间晚于变更C2的批准时间，变更C2检查n1的状态时为未生成、未标识，无法使用该配置项，表明通过此生命周期数据重现的流程是有问题的，也就是说研发过程中的变更管理流程存在问题。审查流程时各步骤应对生命周期数据进行的活动模拟和数据验证见表（）。

一次变更管理活动代理中各个活动过程应进行的操作和检查：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 子活动名称 | 子活动的阶段 | 检查 | 状态操作 |
| 1 | 问题报告 | 申请阶段 | 1. 该变更应是变更的起始，无上一子活动 2. 2.输入的数据项应为问题或需求变更； 3. 输出的数据项应为问题报告；   3.问题报告的要素（上面某章节已确定）应齐全4.该问题报告状态应为尚未被提出；  5.问题报告提出者要有相应权限 6.问题报告的提出时间应与时间顺序相符 | 问题报告状态改变为已提出 |
| 2 | 变更申请 | 申请阶段 | 1.上一子活动应该是问题报告或变更审批未通过 2.输入的数据项应为问题报告（或审批未通过意见）； 输出的数据项应为变更申请单； 3.该变更申请单的要素（上面某章节已确定）应齐全 4.该变更申请单状态应为尚未被提出； 5.变更申请的提出者要有相应权限  6.变更申请的提交时间应与时间顺序相符 | 变更申请单状态改变为已提出 |
| 3 | 变更审批 | 评估阶段 | 1.该变更的上一子活动应该是变更申请 2.输入的数据项应为变更申请单  输出的数据项应为变更审批单（或变更审批结果） 3.变更审批单要素（上面某章节已确定）应齐全  4.该变更审批单状态为审批未完成 6.有明确的变更审批结果 7.变更审批人员要有相应权限。  5.变更审批时间应在时间顺序相符 | 变更审批单状态改变为审批完成 记录该变更审批结果（通过或不通过） |
| 4 | 基线管理 | 执行阶段 | 1.上一子活动应该是变更审批且审批通过，或变更评审且评审未通过;2.输入的数据项应为变更审批通过的意见和变更审批确定的准备变更基线的标识;3.输出的数据项应为准备变更的基线配置项；4.该基线的要素（上面某章节已确定）应齐全 5.该基线状态应为已建立状态；6.该基线应未处于检出状态 | 准备变更基线的状态改变为检出 |
| 5 | 配置项检出（checkout) | 执行阶段 | 1.上一子活动应该是基线管理;2.输入的数据项应为准备变更的基线受控数据项的标识;3.输出的数据项应为准备变更的基线的受控配置项的全集；4.该基线受控库的所有配置项的要素（上面某章节已确定）应齐全 5.受控配置项的状态应未处于检出状态 | 从基线受控库得到的受控配置项状态改变为检出 |
| 6 | 开发过程，执行数据的更改 | 执行阶段 | 1.该变更的上一子活动应该是配置项检出;2.输入的数据项应为准备变更的基线受控数据项的全集 3.输出的数据项为变更后的基线生成的数据项全集，同时还有此过程开发中建立的阶段基线及其生成的数据项全集 4.基线的状态应为尚未建立，新生成数据项的状态为未生成,未标识， 5.执行变更的开发人员符合权限要求 | 变更后的基线、阶段基线的新生成数据项的状态改变为已生成，未标识 |
| 7 | 配置标识 | 执行阶段 | 1.上一子活动应该是开发过程;2.输入的数据项应为开发中建立的阶段基线及其新生成数据项的全集，以及变更后基线新生成数据项的全集（均未标识） 3.输出的数据项为已经标识过的输入数据项 4.这些数据项状态应为尚未被标识 5.执行变更的开发人员符合权限要求 | 阶段基线及其新生成数据项的全集，以及变更后基线新生成数据项的全集状态改变为已标识 |
| 8 | 变更评审 | 评审阶段 | 1.上一子活动应该是配置标识;2.输入的数据项应为开发后基线的受控配置项内容 3.输出的数据项为变更评审单 3.该变更评审单的要素（上面某章节已确定）应齐全 4.变更评审单的状态应为尚未完成； 5.变更评审单的提出者要有相应权限  6.变更评审单提交时间应与时间顺序相符 | 1.变更评审单的状态改变为已完成； 2.记录该变更评审的意见（通过或不通过） |
| 9 | 基线管理 | 完成阶段 | 1.该变更的上一子活动应该变更评审且评审通过;2.输入的数据项应为开发后建立的新基线的受控配置项的全集 3.输出的数据项为开发后建立的新基线 3.该基线的要素（上面某章节已确定）应齐全 4.该基线的状态应为尚未建立 ，未标识；5.建立基线的人员要有相应权限 | 变更后的基线状态改变为已建立，未标识 |
| 10 | 配置标识 | 完成阶段 | 1.上一子活动应该为基线管理;2.输入的数据项应为开发后建立的新基线（为标识） 3.输出的数据项为开发后建立的新基线（已标识） 3.该基线的要素（上面某章节已确定）应齐全 4.该基线的状态应为已建立 ，未标识；5.配置标识的人员要有相应权限 6.基线标识配置时间应与时间顺序相符 | 变更后的基线状态改变为已建立，已标识 |
| 11 | 配置状态纪实 | 记录阶段 | 1.上一子活动应该为配置标识;2.输入的数据项应为变更过程中生成的配置项。基线、问题报告、变更申请、审核、评审记录 3.输出的数据项为配置状态纪实报告 4.配置状态纪实报告要素应齐全 5.本次变更配置状态纪实状态为未完成 5.配置状态纪实的人员要有相应权限 6.配置状态纪实报告的内容应与配置项库中数据项生成状态相符 | 本次变更配置状态纪实的状态改变为已完成 |

## 内容审查

在一次变更中，除了流程相关证据要求外，变更影响分析、变更新生成的配置项是否符合追溯性要求也是需进行审定的重要方面。对此要求的验证需要对相关配置项的具体内容，即3.4节中需求或需求派生配置项数据与追溯性相关配置项数据进行分析。

### 验证方法

内容审查主要是对变更影响分析和变更执行新生成的数据项追溯性进行验证审查，对内容进行该项检查的相关生命周期数据为 3.4节中需求或需求派生配置项数据与追溯性相关配置项数据。具体验证逻辑如下：

**1.变更前，判断变更影响分析的正确性。**

变更执行前，通过文本相似度计算的方法计算需求或需求派生配置项之间的相似度，并进行选择排序，确定各个层次需求间的追溯关系，将确定追溯关系与实际变更中被更改项、受影响项进行比较，初步判断变更影响的正确程度。若是提供了追溯矩阵，则可以用于对追溯矩阵进行检查。

**2.变更后，判断新生成配置项追溯性的正确性。**

变更执行后，计算被变更的需求与受变更影响新生成的配置项之间的相似度，并进行选择排序，以此为依据判断生成的配置项是否满足追溯性关系，来确定变更执行是否正确。

通过图+说明 表述清楚

下节介绍使用文本相似度加选择排序重建追溯矩阵的方法。

### 需求追溯矩阵建立方法

获得生命周期数据项之间的追溯性的流程如图（）所示，包括3个阶段：

流程图

（1）数据预处理阶段。包括对数据项文本的预处理和词向量的训练。具体在第五章实验部分进行说明。

（2）文本相似度计算阶段。预处理完成之后，将待建立跟踪关系的追溯相关数据项文本映射为词向量，，然后把每个数据项文本看作信息检索中的一个查询语句。计算每一个查询语句与其他数据项的文本相似度，根据相似度排序，得到了该查询的排序列表

（3）最后，将排序列表引入学习排序算法模型中，利用机器学习分类算法将排序后的列表进行二次处理，最终得到相关跟踪链接。

接下来主要介绍文本语义相似度计算和学习排序模型中的改进部分。

下面主要详细介绍对文本相似度计算算法的改进：

### 文本语义相似度计算方法

本节基于TF-IDF方法权重计算与缺失词替代，提出一种提高文本相似度计算效果的方法。

###### 1．基于TF-idf策略计算权重

在计算文件相似度时，是先计算词与词之间的相似度，再计算出词与句子的相似度，最后再计算文本与文本间的相似度。但是不同的词能够代表文档信息的程度是不同的，如果将所有的词一视同仁，就忽略了关键词或词组对文档相似度影响更明显这一情况。因此，要对文档中的词计算特征权重，通过计算权重选出更能代表文档信息的词或词组。此处使用TF-IDF算法找出权重足够大的词汇。TF（Term Frequency）是词频，如公式（）所示，指的是目标词在文档中出现次数的频率，tf与目标词能够表示文档信息的程度成正比，；IDF（Inverse Document Frequency）是逆文档频率，如公式（）所示，IDF与含有目标单词或词组的文档数量有关，用来表示目标词在其他文档中出现的频率，IDF值越大，代表目标单词在其他文档中出现的频率越小，则目标词或词组能更大程度的表示文档信息；最后将求TF与idf的乘积，如公式（）所示，得到该目标词的tf-idf值。

(3.1)

(3.2)

其中指目标词在文档中出现的频率，指单词在文档中的词频，表示文档中目标词出现的次数。指目标词在文档集合中的逆文档频率，表示文档总数，表示出现过目标单词的文档数，为了防止出现过目标单词的文档为零而导致分母为零，将该文档数加一处理。最后取对数处理。

###### 2．对关键信息词进行加权计算调整后相似度

使用tf-idf选出能够更大程度反映目标文档信息的关键词后，需要使用专门加权方法计算相似度，以合理使用关键词所携带信息。

当计算关键词与其他词的相似度时，若两词的相似度大于阈值，则进一步按照一定比例提高相似度，应满足 ，具体数值通过实验确定，若两词的相似度小于阈值，则按照一定比例降低相似度，应满足 ,具体数值通过实验确定。计算方法如公式（）所示。

代表经过加权计算后得到的词与词之间的相似度， 代表直接计算得出的词与词之间的相似度，为取值大于1的常数，的值决定了对相似度进行调整的幅度。当或时，应用此公式。

当时，不对相似度进行调整，直接使用作为词与词间最终相似度。

###### 3．改进的文本语义相似度算法

最后基于以上权重调整策略与缺失词处理方法，提出优化的文本相似度计算方法：

Ye X等人在**错误!未找到引用源。**提出了用来计算文本与文本间相似度的算法W2V，该方法计算文本语义相似度的步骤为：

1）通过计算两个词向量的cosine相似度得到词与词之间相似度，如公式（）所示；

(3.4)

其中和分别表示词和的词向量，和分别表示两个词的词向量的长度。

2）计算单个词与文本之间的相似度，如公式 （）所示，计算目标词与文本中所有词的相似度，取最大值即为目标词w与文本T的相似度。

(3.6)

3）计算文本与文本间的相似度。如公式(3.7)所示，算出文本中的每个词与文本的相似度，并求和，最后进行归一化处理，即得到文本与的相似度。

(3.7)

(3.8)

其中表示文本中与文本的相似度不为0的词的集合。

下面是对以上算法的改进部分

（1）首先计算TFIDF权重，然后根据TFIDF进行排序，选择前的单词或词组，组成集合，其中参数设置为经验值0.3。

（2）改写计算词与词之间的相似度的计算方法，当两个词都不属于集合中时，算法不变，仍然按照公式（）计算，得到词与词之间的最终相似度，当两个词中有任一词属于集合时，计算出两词的相似度，并将与设置好的阈值 、比较，

若，则

；

若或，则

其中设置为经验值0.3，设置为取经验值0.7，r设置为取经验值3。得出的作为最终的词与词间相似度，继续参与下一步的计算。

两词之间计算相似度的公式改写为公式（）。其中 a,b,c分别取经验值。。。。

（3）对原算法的第二步进行改进。在w2v算法中，算法得出的相似度与两个文本的输入顺序是相关的，两个文本在算法中并不完全对称，差异出现在计算词与文本间相似度时，不同的文本顺序会得出不同的结果；另外，当短文本是长文本的子集时，若拆分的文本是短文本，长文本中不属于短文本部分的词将会被忽略，计算出的相似度为1，与实际情况明显不符。为了保证计算文本相似度时计算结果的一致性，同时为了避免特殊情况下的错误，在新的算法中，作出如下规定：

计算两个文本间的相似度时，设较短的文本为，较长的文本为，则必须拆分长文本,通过计算中每个词与短文本的相似度，获得文本间的相似度。

### 学习排序算法

通过上述的文本语义相似度计算之后，对于每一个查询都可以排序生成一个排序列表，即候选列表，设置相应的阈值，即可产生相应的需求跟踪关系。但是，此时仅仅使用了文本语义相似度这一个特征，在数据集中还有一些没有使用的特征。综合使用其他的特征可以提升结果的准确率，因此本文使用学习排序算法来进一步对已有的排序列表进行处理。以下将从两个方面描述学习排序算法：特征选择和排序算法。

#### 特征选择

本文选用了五个特征作为学习排序模型的输入，五个特征可以被分为两组：依赖查询的特征和不依赖查询的特征，如表3所示。前三个特征为依赖查询的特征，这几个特征强调查询与候选结果之间的关联关系；后两个为不依赖查询的特征，更强调候选结果文本自身的特点。

（1）语义相似度，通过提出的语义相似度算法WQI计算；

（2）广义Jaccard系数，通过计算文档向量的关系表示两个文档上下文相似度，在本文应用中即为查询语句和一个候选文档之间的上下文相似度，如公式(3.10)所示。

(3.10)

其中，向量和向量均为文档向量，此处文档向量使用词向量的平均值表示。和分别表示向量和向量的数值大小。

（3）IDF之和，在候选结果文本上中计算查询中的词或短语的IDF值，并求和，用来表示当前查询在某一候选结果上的重要程度，如公式(3.11)所示。

(3.11)

其中为查询中的单词或短语，表示在查询结果中的idf值。

（4）关键词数量，表示查询候选结果中关键词的数量，关键词的界定使用TFIDF的值，计算过程与加权策略中介绍的过程相同。

（5）文本长度，表示查询候选结果的长度，在一定程度上体现该结果的有效程度。

表3 学习排序算法选择的特征

|  |  |  |
| --- | --- | --- |
| **特征** | **描述** | **是否依赖查询** |
| 语义相似度 | 使用WQI方法计算的文本语义相似度 | 是 |
| 广义Jaccard系数 | 文本的上下文相似度，文本向量由词向量取平均值表示 | 是 |
| IDF之和 | 候选结果文本上的当前查询中所包含的词或短语的IDF值之和 | 是 |
| 关键字数量 | 候选结果中关键字数量，通过计算TFIDF然后排序和取阈值得到 | 否 |
| 文本长度 | 候选结果文本的长度 | 否 |

#### 排序算法

正如第二章中所提到了，单文档方法仅考虑了单个文档与查询的绝对相关度，忽略了文档间的顺序关系；文档对方法考虑了任意两个文档之间的相对前后关系，相比单文档方法的效果更好；文档列表方法需要考虑每次对查询候选结果列表，当文档数量较大时，需要考虑的数量较大，相对而言没有文档对方法的效率高。综合以上原因，Tr-WELR方法模型中使用了文档对方法，并改进了当前成熟的IR SVM算法。

IR SVM的基本思想和文档对方法处理思想一致，都是将排序问题转化为分类问题。IR SVM模型将两个样本和表示成一个训练样本，然后使用SVM模型训练和预测分类。由于IR SVM是对Ranking SVM的改进，下面将首先介绍Ranking SVM，然后再详述IR SVM对Ranking SVM的改进，最后提出在需求跟踪任务中对IR SVM的改进。

##### Ranking SVM

Ranking SVM算法的思想是将排序问题转化为pairwise的分类问题，然后使用SVM分类模型进行学习并求解[43]。在排序应用中，假定有两组查询对应的文档集合，每组查询结果集合中有三个等级，分别是等级1、等级2和等级3。举例说明，在第一组查询结果中有三个对象、和，分别属于等级1、等级2和等级3，如图13所示。图中的向量是第二章公式(2.2)中中的权重向量。但是，此时对于每一个查询得到的候选文档集合，都有一个各自的权重向量，在应用中非常麻烦。因此，对同一组查询结果集合中的不同等级的对象的特征向量进行组合，形成新的特征向量，即把对象映射到另一个向量空间，然后根据另一个向量空间的向量进行排序。比如将上述的三个对象重新组合为：、和，并且给这些对象重新打标签，如将、和标记为负相关，相应的、和标记为正相关，如图14所示，即可将以上的排序问题转化为二值分类问题，在该例子中就可以通过训练线性SVM分类器对上述新的向量空间的向量进行分类，进而可以计算得到同一组查询结果集中的不同等级的向量的前后词序。

训练数据表示为，其中为，每一个训练实例由两个特征向量表示，标签由表示，并且的取值为+1和-1，分别表示正相关和负相关。Ranking SVM的最优化问题可以形式化的表示为公式(3.12)。

(3.12)

*s.t.(使得)*

其中为松弛变量，是训练实例的数量，2-范数：https://img-blog.csdn.net/20130624111745156?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvbGVmdF9sYQ==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center，Euclid范数（欧几里得范数，常用计算**向量**长度），即**向量**元素绝对值的平方和再开方，matlab调用函数norm(x, 2)是第二范式，把松弛变量代入公式(3.12)中，可以得到：

(3.13)

其中，加和的第一项是合页损失函数, 第二项为正则项，防止过拟合。

##### IR SVM

IR SVM是Ranking SVM在信息检索领域的一个改进[36]。Ranking SVM在信息检索领域的不足之处包括两个方面：

（1）Ranking SVM是将排序问题转化为分类问题，在学习过程中使用了0-1分类损失函数。在信息检索任务中，最终结果列表中排在前面的对检索效果的影响更大，而Ranking SVM对每两个文档的相对顺序都一视同仁。举例说明，有三个等级，其正确的排序顺序应该为等级1、等级2、等级3，则等级3>等级2和等级3>等级1都是错误的相对顺序，但是二者对Ranking SVM的训练过程造成的影响是相同的，显然，这与实际的排序过程有一定的误差。

（2）另一方面，Ranking SVM同等对待不同查询下的结果对。举例说明：每个文



图13 排序问题示意图



图14 将排序问题转化为分类问题示意图

档使用相应等级来表示，两个查询的结果如表4所示。对于查询一，在转换向量空间时可以有2个“等级3-等级2”的文档对，4个“等级2-等级1”的文档对，2个“等级3-等级1”的文档对，共8个正相关文档对可供训练；对于查询二，则有两个“等级3-等级2”的文档对，八个“等级2等级1”的文档对，四个“等级3-等级1”的文档对，共有14个正相关文档对可供训练。虽然两个查询得到的结果中等级结构相同，但是由于数量不同，查询二对Ranking SVM模型的影响会比查询一大，因此最后的结果会有偏差。同时，这也与信息检索任务中所要求的“每个查询的重要性等同”[23]是不相符的。

表4 排序列表举例

|  |  |
| --- | --- |
| **查询** | **结果** |
| 查询一 | 五个文档：等级3、等级2、等级2、等级1、等级1 |
| 查询二 | 七个文档：等级3、等级2、等级2、等级1、等级1、等级1、等级1 |

由于Ranking SVM以上两个方面的不足之处，IR SVM将二值分类问题改进为代价敏感的分类问题，即对来自不同查询的文档对，或者不同等级的文档对设置不同的损失权重。由于在排序结果列表前面的结果比后面的结果的重要程度高，在计算时会对在前边出错的文档对加大损失函数的权重，即出错的文档在列表中顺序越靠前，所需要付出的代价越高；IR SVM对上述Ranking SVM第二点不足的改进是，对排序出错的文档，如果某条查询的结果列表文档数量较少，则加大损失函数的权重，反之，如果结果列表中文档数量较多，则减轻权重，从而减少因为查询本身对查询效果的影响。IR SVM的最优化问题可以表示成如公式(3.14)所示。

(3.14)

其中表示第*i*个文档对的等级，表示等级下的权重值，表示第*i*个文档对对应的查询文档，表示查询对应的相关文档对的参数。和作为惩罚因子，分别用来弥补上述Ranking SVM两个方面的不足。参数值的确定是使用一个启发式算法：选定一个评价指标，对每个查询，找到能够使得评价指标最优的排序序列，然后随机交换任意两文档的位置，查看评价指标的降低值，重复该过程，最后对降低值取平均值作为分类下对查询文档对所加的权重。参数值是对查询项对应的相关文档的数量取倒数，当相关文档数目较少时，该值相对较大，进而弥补了在文档数较少时Ranking SVM优化函数对其重视程度低的问题。

##### 改进的IR SVM

在IR SVM中考虑到了查询结果列表中排在前面的内容对结果影响较大，因此提出了使用公式(3.14)中的权重参数来解决这一问题，即对不同等级的查询文档设置不同的权重。如上一小节所说，IR SVM使用启发式算法计算值，但是该方法仅仅考虑了由于顺序不同导致的权重值的不同。

经过3.3节的计算，查询文档已经与一个文档序列建立了关系，并且序列中的文档与查询文档通过文本语义相似度值相关联。和查询项相关的文本语义相似度中不仅包含了文档相对顺序的信息，同时还包含了不同位置的文档与查询项的具体关系和一个可以量化的值，因此可以将上述权重参数使用文本语义相似度来表示，具体表示如公式(3.15)所示。

(3.15)

其中和分别表示第*i*个文档对（pair）中两个文档与查询项的文本语义相似度值，的取值为二者中的较大值与二者差值绝对值的乘积。用来表示文档对中排序更靠前的文档在排序列表中的重要程度；指文档对中两个文档与查询项相关度的差值，用来区分不同等级（不同位置）的文档的重视程度。改进之后的参数省去了启发式取值的繁琐过程，同时包含了查询结果的相对顺序和每个查询结果与查询项的相关程度。此时，学习排序算法的最优化问题可以表示成公式（3.16）所示。

（3.16）

其中为公式（3.15）表示的内容，的意义不变，依然表示查询项对应的查询结果数量的倒数，用来避免某一个查询项有过多的查询结果，从而对训练结果造成偏差。改进的IR SVM算法使用SMO（Sequential Minimal Optimization，序列最小最优化）算法实现，同时由于实例集合非线性可分，因此引入核函数，这里使用高斯核函数，如公式（3.17）所示。

(3.17)

其中，为映射函数，在高斯核函数中，并没有指定的具体函数实现，写出该映射函数旨在方便下面对问题的描述。

易知，公式（3.18）是公式（3.16）的等价形式。

（3.18）

引入拉格朗日乘子并对其求极大极小问题后，可以得到对偶问题，如公式（3.19）所示。

(3.19)

使用SMO对上述对偶问题求解的步骤如算法1所示，计算得到所有的值和*b*值，然后可得到SVM分类器，分类决策函数如公式（3.20）所示。

(3.20)

其中是符号函数，当*z*大于等于0时，函数值为+1，当*z*小于0时，函数值为-1。

|  |
| --- |
| **算法1** 改进的IR SVM中的SMO算法部分 |
| **输入：**训练数据集，其中和均属于，*n*为特征数量，，，核函数中参数，精度值，常数*C*；查询项与查询结果的相似度集合，为查询项和文档的语义相似度值；每个查询项对应的结果数量集合；最大迭代数*maxIter*  **输出：**向量，*b*向量  **初始化工作：**创建*alpha*向量并初始化为0，计算所有的值和值  **while** 当前迭代次数 < maxIter **do**  **for all** **do**  计算当前alpha和b下分类器的值和的误差值  **if** Ei超过的范围 **and** 当前alpha[i] 满足限制条件 **then**  随机选择另一个alpha[j]  同时优化alpha[i]和alpha[j]  **if** 通过当前alpha计算得到的误差值都满足要求精度值 **then**  结束本次循环  **end if**  设置b向量  **end if**  **end for**  **if** alpha在上次循环中没有变动 **then**  当前迭代数加一  **end if**  **end while**  **return** alpha, b |

完成上述步骤后，即可通过分类器预测查询结果文档的两两相对顺序关系，然后对所有的相对顺序关系进行汇总，即可得到整体的文档排序关系。本文在第四章中将改进的方法与未使用学习排序模型的算法、原IR SVM方法分别进行了对比，实验结果表明在本文任务总，改进的IR SVM是有效的，并且比原有的IR SVM方法的精确率更高。

## 本章小结

本章首先给出了基于word embedding和学习排序算法的适航领域需求跟踪算法模型框架，并对框架中包含的几个部分进行了简单介绍。接下来详细介绍了算法模型中改进的部分，其中包括：在计算文本语义相似度时提出的在word embedding的基础上添加加权策略和查询扩展，并以此为基础改进了文本语义相似度的计算算法。由于使用单独的信息检索技术在软件需求跟踪关系恢复任务中对数据的特征使用较少，因此提出了使用学习排序算法对结果的精度做进一步的提升，并改进了IR SVM算法。

## 4.5 本章小结

本章首先介绍了面向发动机适航的软件需求追踪算法的模型框架，然后介绍了基于词向量的文本相似度算法，算法包括了语义相似度的计算和缺失词语的动态处理，最后，介绍了机器学习排序将排序问题转化为0-1分类问题的原理，并给出了损失函数的优化和推导过程，结合当下的流行的机器学习技术，通过组合文本中的多个特征从而提升结果精度。

# 适航领域软件需求跟踪算法模型验证

**本章通过实验对第四章提出的算法效果进行检验，并将该算法与其他经典算法对比，评估算法的性能。**

5.1实验设置：

5.1.1数据来源。

5.1.1.1 适航开发方提供的的生命周期数据  
 进行预处理方法

5.1.1.2 公开的可用数据集

5.1.2对比算法，

5.1.3指标介绍

5.2实验结果：

## 数据准备

取得适航开发中相关追溯性数据的方法：

1.来源于开发方，追溯性矩阵无误的需求数据及其链接。由于目前国内生命周期数据提供方提供的生命周期数据格式和类型一般无法达到Do178c规定的格式要求，所以为取得可用于实验的需求数据及其链接关系，需要专业人员将数据标注，并由机器或人工取得生命周期书记及其追溯矩阵，还需要领域专家对追溯矩阵进行检查，确定追溯矩阵正确性，获得正确性追溯矩阵与实验取得的追溯情况进行对比。并由专门人员将其录入数据库中。

2.以上是从已完成的适航项目中获取实验数据的方法，由于适航项目保密性强难以获得，特别是经过专家验证的追溯性准确的数据更难以获取，因此，我们选取了CoEST网站提供的几个公开数据集作为实验数据。CoEST是个为研究者进行软件和系统可追溯性研究提供数据集的网站，该网站提供了多个从实际的项目中整理而来的软件数据集，我们选择MODIS、CM1、eTOUR、EasyClinic和iTrust作为实验数据，这些数据集不但提供各层次的需求，而且提供了经过专业人员验证的追踪链接，能够显著减少我们进行数据标注以及人工录入、邀请专家进行链接标注与验证的工作量，同时由于这些数据集被研究者们广泛使用来评估算法的有效性，给我们对实验性能的对比提供了极大便利。

MODIS由NASA提供，包含其公开项目的部分需求数据，该数据集由Hayes[14]等人整理，被广泛应用在需求可追溯性的研究中；CM1来源于NASA基础项目的子项目CM1，它包含软件的高层需求和低层需求。eTOUR来源于一个面向旅游的导航软件，由Salerno大学开发； EasyClinic来源于一个医疗系统管理软件，同样由Salerno大学开发；iTrust来源于一个登记医用药品数据的开源项目。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **公开数据集** | **高级需求** | **低级需求** | **用例** | **UML交互图** | **测试用例** | **类描述信息** | **代码类** | **追溯链接数** | **词条数** |
| CM1-NASA | 22 | 53 | × | × | × | × | × | 45 | 5767 |
| eTOUR | × | × | 58 | × | × | × | 116 | 308 | 97452 |
| MODIS | 19 | 49 |  | × | × | × | × | 41 | 4534 |
| EasyClinic | × | × | 30 | 20 | 63 | 47 | × | 1257 | 21882 |
| iTrust | × | × | 131 | × | × | × | 367 | 534 | 123501 |

图15 数据获取过程示意图

对数据进行预处理

需求数据经过整理后，根据类型可以分为两类：（1）文本文件（2）源代码文件。

在使用词向量相似算法对需求数据进行计算之前，由于在文本或代码中与文本内容无关的数字、词汇、符号等，为降低无关词汇对相似度计算的干扰。还需要对文本或代码进行以下处理：

1. 分词，通过空格，将文本或代码分割为单个单词。
2. 去除其中除了下划线和连字符之外所有的非数字、非字母符号，
3. 将所有单词转化为小写
4. 去除停用词。对于文本文件，去除的停用词时使用Python编写的NLTK[48]（自然语言工具包），NLTK 提供了大多数语言的停止词表；对于代码文件，去除的停用词是其代码语言的关键字

最后得到最终用于实验的需求数据文本。

这个图要重画



Siwei L.等人[45]指出在训练词向量时，语料的领域相关性比语料库的大小更重要，并且语料的领域性越强，词向量的表示效果越好。

获取了软件开发中产生的具有追溯关系的数据集后，还需要获取用来训练词向量的语料库，本实验中获取语料库的过程如图 （）所示，首先下载源数据，可选源数据包括百度百科、中文百科、维基百科等，其中维基百科提供数据库的下载，此处选择维基百科英文数据库作为源数据；第二，对源数据进行过滤处理，由于维基百科数据库特别庞大，数据库达到13.9G，同时，为了使获得的语料库领域性更强，对源数据进行过滤处理，此处选择《信息技术软件工程术语》作为词库，生成术语表，并使用术语表筛选维基百科源数据，得到用来训练词向量的语料库，最终生成的语料库大小为372M。

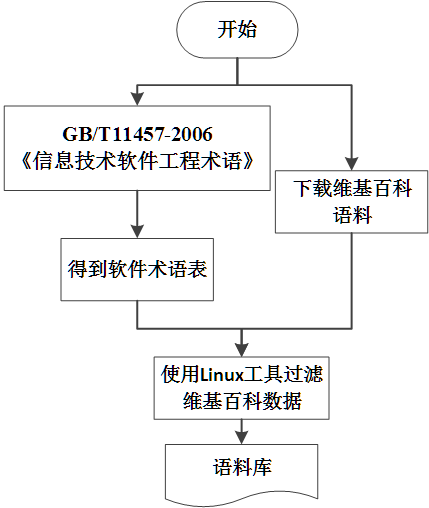


图16 获取训练词向量语料库流程

训练词向量：

本实验中使用开源项目Gensim来训练词向量，Gensim由python实现， Gensim是Radim Řehůře在其论文《**SCALABILITY OF SEMANTIC ANALYSIS IN NATURAL LANGUAGE PROCESSING**》中，使用python实现，集成了词向量模型，是一款很强大的工具。

在训练设置时，词向量训练模型选择Skip-gram，预测窗口大小设置为5，生成的向量的维度设置为200；同时选择softmax选项来提高训练速度。

## 实验设置

在实验中，使用本文提出的。。。相似度算法与LSI算法、W2V算法作对比，检验本文提出的改进的相似度算法的提升效果。LSI算法指的是使用潜在语义索引方法恢复软件跟踪链接的方法，该方法已经在跟踪恢复领域广泛使用，是一个比较成熟的方法。使用LSI方法进行文本相似度计算的流程如下：（I）将文本使用one-hot方法表示为矩阵（II）使用奇异值分解方法对矩阵降维到潜在语义空间；（III）通过对降维后的文本向量计算内积或者余弦相似度确定文本相似度。使用LSI作为对比方法是为了说明使用文本语义相似度计算比使用基于统计的检索方式效果更好。

表7 对比方法

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验** | **对比方法** | **文本词语顺序** | **基本方法** | **计算方法** | **文本/词向量 维数** | **算法改进** |
|  | LSI | 不考虑顺序 | 基于词频统计与文本向量空间 | 将文本表示为向量，并进行降维处理后计算文本相似度 | 通过SVD降维 | 无 |
|  | W2V | 考虑顺序 | 基于词向量空间 | 将词表示为向量，通过计算词相似度计算文本相似度 | 通过Word2vec确定词向量的维数 | 未使用IDF |
|  | 改进的文本相似度算法 | 考虑顺序 | 基于词向量空间 | 将词表示为向量，通过计算词相似度计算文本相似度 | 通过Word2vec确定词向量的维数 | 基于idf计算权重并对权重单词，对缺失词间相似度算法进行规定，进行加权处理 |

W2V方法是使用论文[16]中的方法来解决需求跟踪任务，算法核心在于对文本语义相似度的计算，如公式（3.7）所示。与W2V方法对比是为了通过实验验本文本相似度算法改进部分对性能的提升。

本论文中实验所用计算机使用八核英特尔i7处理器，内存为8G。

## 算法评估标准

为了精确、有效的对算法的性能进行比较。首先给出评估文本搜索领域算法性能的3个指标，分别是精确率、召回率和以及F测度。在描述模型性能评估指标之前，记：

为被检索文本列表中与查询的文本确实有追溯关系的文本数量。

为被检索文本列表中通过算法与被查询的文本建立了追溯关系的文本的数量。

为被检索文本列表中与查询的文本无追溯关系，却被算法错误的建立了追溯关系的文本数量。

为被检索文本列表中与查询的文本有追溯关系，却未通过算法得到追溯关系的文本数量

**精确率（Precision）：被检索的文本列表中通过算法确定了正确追溯关系的文本数量与通过算法建立了追溯关系的文本数量的比，可以被表示为**

**召回率（Recall）：被检索的文本列表中通过算法确定了正确追溯关系的文本数量与列表中确实存在追溯关系的文本数量之比**，可被表示为：

**F测度（F-measure）:**精确率和召回率越高，说明算法的效果越好，但是精确率和召回率之间是存在着相互影响的，为了提高精确率，召回率可能会降低；当召回率变高时，精确率可能会变低。F测度能够综合考虑精确率和召回率，用来表示精确率与召回率的调和平均值，F测度值越高，代表算法综合考虑精确率和召回率后的效果越好。

(4.3)

## 实验结果与分析

本节将详细描述实验结果，并使用前面提出的性能指标精对实验结果进行分析评估。

本实验分别使用LSI方法、W2V方法和WQI算法对数据进行相似度计算，根据计算出的相似度建立追溯性链接，并使用精确率、召回率和F测度对方法进行评估，实验结果见表（）。

表中，PRE代表方法结果的精确率，REC则代表方法结果的召回率，HReq、LReq、UC、CC、UMP、TC分别代表高级需求、低级需求、用例、类文件、UML交互图和测试用例。

比较表（）中展示的精确率和召回率，可以得出结论，基于词向量的W2V算法和WQI算法的效果在总体上要优于LSI方法。具体来说，与LSI方法相比，WQI算法的精确率相对高出了约33.3%，而召回率则高出了约24.5%；与W2V方法相比，WQI算法的精确率相对高出了约6.6%，而召回率则高出了约17.2%。

表8 三种方法的实验结果

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **数据集** | | **LSI** | | **W2V** | | **WQI** | |
| **PRE** | **REC** | **PRE** | **REC** | **PRE** | **REC** |
| CM1-NASA | HL→LL | 0.127 | 0.41 | 0.262 | 0.217 | 0.371 | 0.329 |
| GANTT | HL→LL | 0.286 | 0.332 | 0.278 | 0.418 | 0.255 | 0.563 |
| eTOUR | UC→CC | 0.077 | 0.221 | 0.098 | 0.332 | 0.088 | 0.415 |
| iTrust | UC→CC | 0.009 | 0.45 | 0.192 | 0.363 | 0.198 | 0.322 |
| EasyClinic | UC→ID | 0.259 | 0.833 | 0.338 | 0.75 | 0.342 | 0.806 |
| UC→TC | 0.45 | 0.755 | 0.522 | 0.867 | 0.499 | 0.867 |
| UC→CC | 0.317 | 0.503 | 0.215 | 0.677 | 0.232 | 0.76 |

实验使用的3种方法在五个数据集上，共有7组有追溯关系的数据，图（）展示的是这3种方法在这7组数据上的F测度，通过图（），我们可以直观地看出，XXX算法在6组数据上的效果都要优于LSI方法和W2V方法，对比W2V和发发发算法，发发发算法中对于IDF-df高的词的处理是合理的，能够使占权重高的词语更大程度代表两个互相比较的文本之间的关系，从而更清晰的区分出清晰度。

可以看WE、CWI算法在各个指标上对比LSI均有所提高。产生这种结果的一个主要原因是LSI算法在计算文本相似度时，使用奇异值分解的方法将文本映射到一个较低维的空间，具有相同主题的词语被映射到同一维度，而这种变幻是通过数学方法得到，并没有考虑词语的语义特性，而且LSI算法没有考虑词语之间的先后关系，相对于其他两个算法，忽略了词语之间的关系，因此在性能上要明显低于W2V和 。算法。

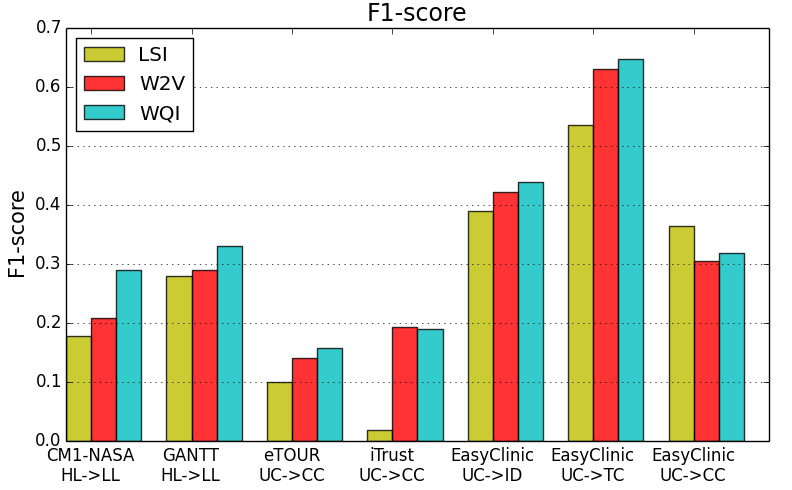


图18 LSI、W2V和WQI三种方法在七组数据上的F测度

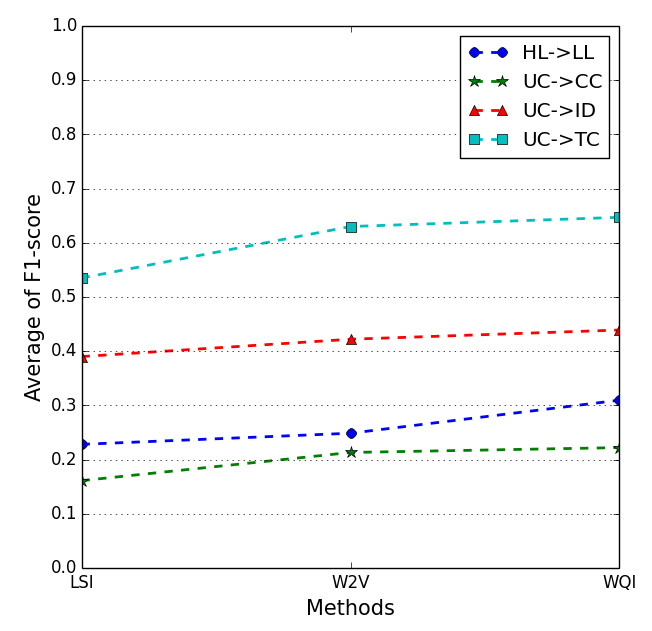


图19 LSI、W2V和WQI三种方法在不同类型的跟踪链接上的平均F测度

相比W2V方法，WQI算法引入了TFIDF加权策略和缺失词词向量处理办法，并且改进了计算的过程。从结果上来看，WQI算法所做的改进提升了结果的精确率和召回率，即提升了需求跟踪任务的性能。对于idf值较高的词，与文本含义的相关性更强，在算法中相对的提高这些词对文本相似度算法的影响，能够得到更好的效果；而对于缺失词利用相关词向量进行替代的方法，把缺失词的信息也有效利用，取得较好的效果。

综上所述，相比LSI方法和W2V方法，文本语义相似度计算算法WQI能够在软件需求跟踪任务中得到更好的效果。

## 本章小结

本章是本文的实验部分，目的是对本文所提出的软件需求跟踪模型Tr-WELR进行验证。首先介绍了数据准备流程和实验所用到的数据集的特征和来源，然后根据文本种类的不同分情况介绍了文档预处理过程，接下来对实验中的对比方法、训练词向量的语料库和实验环境做了介绍，并给出了模型的评估标准，最后对实验的结果和现象进行分析，给出结论。实验证明，在多组数据集和多个不同类型的跟踪链接上，本文提出的WELR模型能够较好的完成恢复跟踪链接的任务，）在软件文上使用提出的基于word embedding的文本语义相似度算法准确率有所提高；