其他研究提出了许多对其进行开发或审定的方法，但由于软件的规模、特点、要求等的不同，此项工作涉及到的管理软件，开发计划、配置计划等，在开发中都会有很大的差异， 针对Do178c的针对于复杂的开发环境、开发工具等， 对于工具的使用和过程的控制方法等，过大的差异和过于复杂的过程，使得 对配置管理过程的审定只能依靠专家人工的方法。

本研究为了解决此项问题，只专注于 审定标准中 规定好的 目标的满足和审定，对于具体研发过程中的工具、流程规定等， 部分可由专家直接通过 经过验证的 工具进行审定，部分经过预处理后进行验证， 而研发方在制定开发计划时，也可在于审定人员沟通时，参照在此模型的基础上 制定软件配置管理计划的总思路，对于其他需要增加或者考虑的部分，再另行规定，单独审定。

有的只专注于研究开发过程中如何符合标准，有的对于审定方法的研究主要还是比较理论的层次， 如 具体项目配置管理软件的设计， 针对项目无关的目标验证模型（也是针对于开发的）等

#### 3.2.4 对分析后的目标进行总结归纳

对3.1节中对目标分析进行分析的结果，进行统一的分析和归纳，仍然按照分类，得出以下的证据性要求。

1. 人工进行审查的部分
2. 根据流程进行审查的部分
3. 根据配置项数据审查的部分

|  |  |
| --- | --- |
| 人工进行审查的部分 |  |
| 流程审查 |  |
| 配置项具体数据 |  |

### 3.3 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

#### 3.3.1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（**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 |

#### 3.3.2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

#### 3.3.3 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 | 数据类型 | 数据项内容 | 原子性数据项 | 数据元内容 | 备注 |
| 1 | 配置管理记录 | 变更的配置管理记录 |  | 配置管理记录应能以某种便捷的方式确定每次变更的批准时间，完成时间。变更的基线，生成的基线。相对应的其他生命周期数据的标识。 |  |
|  | 流程相关 | 需要变更的基线 | a.2.1基线的受控配置项  a.2.2基线.pre  a.2.3基线.next |  |  |
|  | 变更内容相关 | 变更完成后的基线 | a.3.1基线的受控软件库  a.3.2基线.pre |  |  |
|  | 其他受控生命周期数据 | 配置项库，软件受控库 |  |  |  |
|  | 配置管理 | 变更申请单 | a.5.1变更影响分析（包含受变化影响的生命周期数据）  a.5.2受影响的配置项或数据元列表（变更对象）  a.5.3软件发生错误的起始处（如果是需求变更则需求处）  a.5.4变更和行动 ） |  |  |
|  |  | 变更审批单 | a.6.1变更影响分析 （包含受变化影响的生命周期数据）  a.6.2受影响的配置项或数据元列表（变更对象）  a.6.3软件发生错误的起始处（如果是需求变更则需求处）  a.6.4变更和行动 ） |  |  |
|  |  | 问题报告 | a.7.1变更影响分析 （包含受变化影响的生命周期数据）  a.7.2受影响的配置项或数据元列表（变更对象）  a.7.3软件发生错误的起始处（如果是需求变更则需求处）  a.7.4变更和行动 ）  a.7.5问题描述  a.7.6问题报告的批准、关闭情况。 |  |  |
|  |  | 追溯矩阵 | 包括配置项追溯矩阵、基线追溯矩阵 |  |  |
|  |  | 基线库 |  |  |  |
|  |  | 配置项库 |  |  |  |
|  |  |  | 软件配置索引 | 软件配置索引列表 |  |

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。

### 3.4 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，提出该统一变更流程。

根据对Do178c的分析结果，特别是对证据需求的作为依据，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，主要考虑以下几点：

1. 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。
2. 该流程应该尽量简洁。作为审定使用流程，理想情况应是对于实际开发的流程，既能使开发中实际使用的流程能够符合关键要素，又不对开发中提出太多于标准目标无关的流程。在开发过程中，存在的配置管理或变更管理的软件都不相同，过多的要求必然给开发造成更多困难，也可能导致无法对一些符合do178c要求，但由于审定用流程过于复杂而无法对其进行审定。

软件需求变更管理的研究与实践 赵海英 第三章 优化的统一变更管理

优化的统一变更管理 Rational的统一变更管理为基础（UCM）,同时考虑Do178C标准的相关目标和要求，加上其他促进变更管理的措施，建立的“优化的统一变更管理”

变更处理流程：1.变更评估处理流程 2.软件配置管理流程（变更实施流程）（分为两个部分，1.变更评估 问题报告、变更申请、变更审核 2. 新的基线的建立、配置项改变，、软件受控库的改变，更改的配置项的识别等，还涉及到可追溯性的判断、）

强调的是过程而非方法，不需要借助复杂的变更管理工具。

一、问题报告或需求变更，变更可以是由于出现了问题提出，也可能是有需求的变更引起的。

二、变更申请：

三、变更评估：

四、变更实施（配置管理）：

1.首先根据确定 需要重新执行哪个开发过程，每个开发过程，又要进行验证和核查，分别建立核查基线和评审基线。

2.先Check out，开始第一个开发过程，开发过程结束后，建立核查基线，进行核查，执行软件的验证及质量保证，核查通过后，建立评审基线，由软件的评审团和QA进行评审，评审通过后，建立本开发过程的审查基线，并进入下一个开发过程。

3.当所有开发过程完成，并且经过核查和评审合格后。变更实施阶段结束

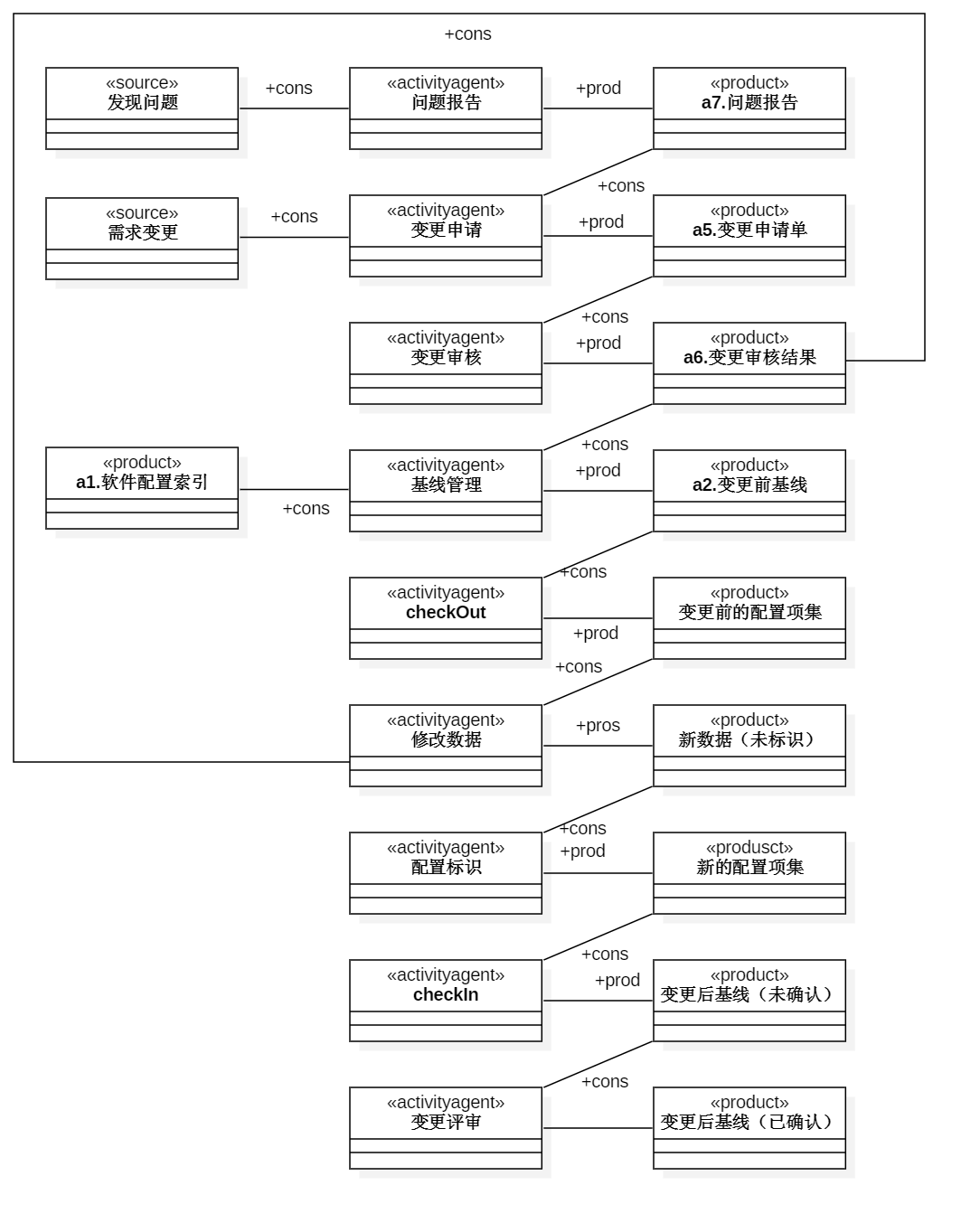
五、变更核查：XX人员根据变更核查和评审报告，记忆是否达到变更目的，对此次变更进行评估。若通过，建立新的产品基线，变更完成，建立新的审查基线。若不通过，重新进入变更实施阶段。

统一的变更模型的流程图如图（）所示：

一个图

### 生命周期数据和变更流程间是输入输出关系

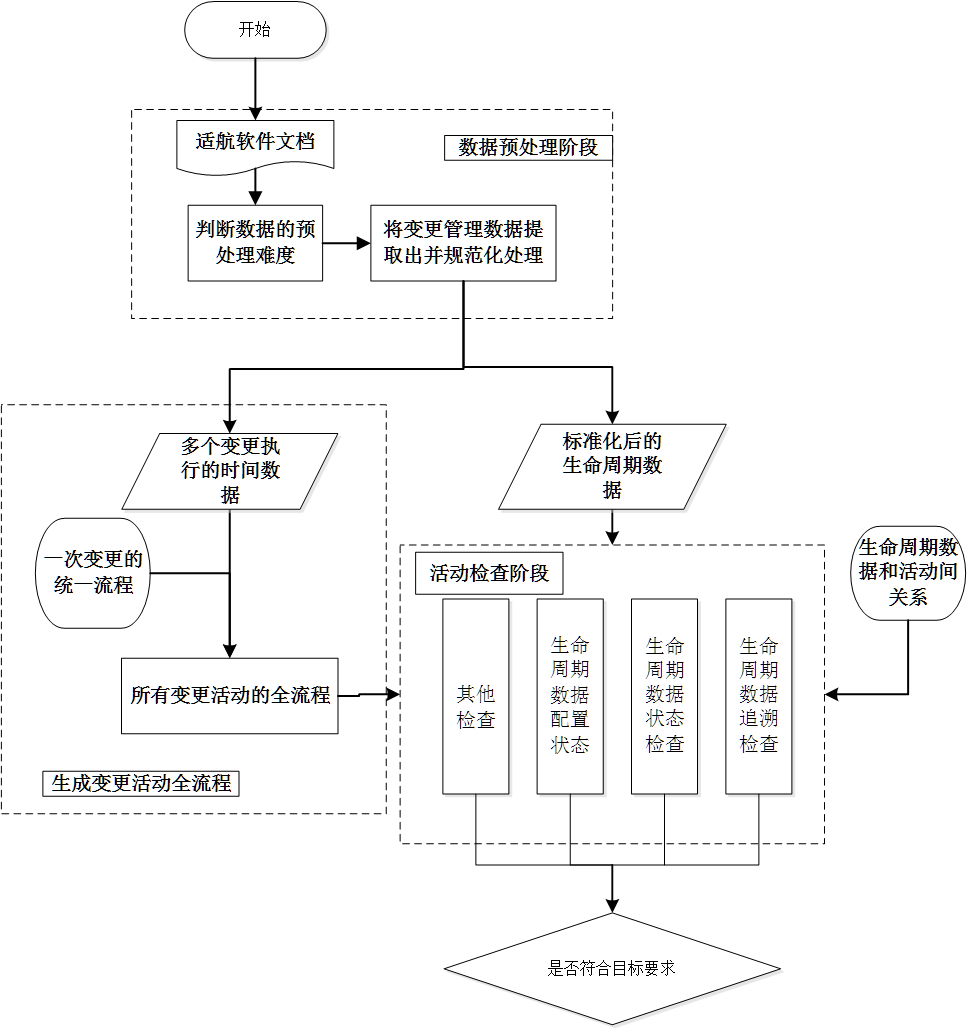
图() 展示的是各项生命周期数据和统一的变更流程间的输入输出关系



#### 数据与流程关系图

# 第四章 适航领域变更流程审定模型介绍说明

## 4.1 审定模型图



将需要达到的目标分为几类。1.追溯性。2。需满足的时间、人员、归属、已被验证、顺序关系、并发关系、 一致性等关系。先后关系、协作关系、数据的传递关系乃至资源的共享关系等。2.X还有多次变更间的关系，3.对于充分性的要求，无法进行判断，只能在初审（根据审定指南相关部分，由审查人员人工提前审查）。

## 人工审查（包含预处理部分和辅助审查部分）

 在预处理时，由审查人员根据研发方提供的生命周期数据的形式、精细程度、处理难度等情况，对生命周期数据进行初步审查（充分性等），确定此次审定的精细化层次，然后按照审查模型要求的生命周期的格式将原始生命周期数据进行处理，标准化该数据。

## 流程审查

### 4.3.1 变更活动全流程

在上一章中，我们确定了统一的变更管理的流程，将一次变更中生成的生命周期数据 代入到此统一的历程中，即可重现此次变更的具体流程。通过重现变更流程，我们可以对变更过程中的 **各种关系** 是否符合要求进行验证。但如果只进行这项工作，则忽略了多了变更间存在的顺序关系 及 资源的共享关系、生成的先后关系等。因此需要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。

需要考虑到的因素：1.各个变更中关键的时间节点是哪个。

2.涉及到了哪些关系

3.对于变更的最终结果是取消或是完成，应该也有所体现

因此 基于各个变更各个节点的时间信息，可以将每个变更按照变更执行的顺序 放在了统一的时间线上，这样就确立了多次变更的执行顺序，所有的变更活动按照研发工作中变更的时间节点排序，得到了所有变更活动的全流程。代入相应的变更相关生命周期数据信息，多个变更间的 **各种关系**  也可以通过重现变更过程而得到验证。

### 4.3.2 变更子活动工作（检查）

由上一节的方法，我们已经得到了全部变更的全流程。本章介绍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对DO178c的证据目标进行检查的方法。

DO-178C没有明确定义和解释活动之间的各种关联关系，例如，先后关系、协作关系、数据的传递关系及资源的共享关系等。

在研发过程中，变更过程的各项活动的任务包括生成数据、进行审核、进行更改等具体性的工作，

在重现变更流程时，我们依据开发中各项子活动的任务和具体操作，规定相应活动在重现过程中的操作，操作分为了 数据状态改变 和 数据检查验证 两个大类。数据状态改变指的是。。。。，包括。。。。。 ，是为数据检查验证提供依据的步骤；数据检查验证则是通过 对相应数据项的存在性，状态进行检查，判定在此过程中存在 互相矛盾、冲突、数据缺失等情况，

例如，变更 序号1 批准时间为 ，生成了配置项 A1,该变更完成的时间是，变更完成时，会将A1生成状态改变为已生成，而变更序号为 2的变更，变更审批的时间是 ，对基线 S1 进行 变更，而基线的被控库中包括配置项A1，此时变更2会发现使用配置项A1时词配置项已生成，说明此项检查无问题，而 若是变更1的完成时间晚于变更2的开始时间，变更2检查A1的状态时为未生成，无法使用该配置项，

、出现此情况 表明了 通过生命周期数据 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。表（）指明了变更管理的各项子活动应对相关生命周期数据进行的具体配置和验证操作。

本文变更的顺序以及变更前后基线，假设在配置管理汇总进行了记录，Do178c并没有对配置管理记录的具体内容进行明确的规定，在制定配置管理计划时，也可能以其他方式体现

配置管理记录确定变更的顺序，

问题报告、变更申请和变更审核结果 可以由配置管理记录取得， 并可以得到 变更 前 变更后的基线及基线的受控配置项索引。

*子活动过程开始*

*1.首先对变更所处的状态进行检查*

*2.然后（生命周期数据的类型由统一的变更流程来确定，具体哪一条记录则根据变更管理记录确定）确定了哪些生命周期数据是要检入该过程的，通过配置管理记录从大量的生命周期数据库中获得此子活动应输入的生命周期数据（获得的同时也对该数据的存在性进行检查）；*

*3.然后对数据项的状态进行检查；*

*活动实施*

*4.接着根据活动的任务，对输入的生命周期数据的状态进行相应的修改，对于会生成新的配置项或基线的开发活动，生成的新的配置项通过读取变更管理记录得到，修改的同时也对此项子活动生成的数据的存在性进行检查，并确定该活动生成的新的生命周期数据的状态。*

*5.根据子活动的类型和活动的结果重新修改变更所处的状态，并确定进入哪个下一子活动流程*

*进入下一子活动，从第1步重新开始。*

1.2.3步属于活动执行前的检查阶段。

4.5.6属于活动执行后的状态设置阶段。

全过程，对 执行人进行检查

对 数据项内容的合理性进行检查

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：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 检查哪些存在性（检查了流程顺序和生命周期数据存在与否） | 检查状态  （查看了流程的顺序） | 配置哪些状态（为检查流程顺序做准备） | 检查追溯性要求 | 检查时间性要求，若数据中有时间 | 检查其他特点 |  |
| 问题报告（需求变更分析） | 问题报告（各个要素是否齐全），问题报告的提出人是否符合权限 | | | | | | |
| 变更申请 | 变更申请各个要素是否齐全。包括变更申请人，（追溯性要求） | | | | | | |
| 变更审批 | 变更审批 各个要素是否齐全（要变更影响到哪条基线（得到准备进行变更的基线），变更的起始项，影响到哪些其他配置项），审批的结果。（追溯性要求）， | | | | | | |
| 基线管理 基线checkout | 检查需变更的基线状态是否为存在且已生成 | | | | | | |
| 配置项checkout | 从基线中获得配置项，改变基线状态为锁定，检查配置项是否存在，是否为已生成，改变配置项标识状态为已锁定。 | | | | | | |
| 开发过程，执行数据的更改 | 在生命周期数据中定位，此次变更建立的新基线（若变更成果），开发新已生成的配置项及其受控配置项库。 | | | | | | |
| 配置标识 | 新的配置项标识是否已存在，是否已为未生成状态，标识是否与其他标识重复，并将其标记为已生成状态。 | | | | | | |
| 新数据在开发库 | 对新生成的配置项进行追溯检查。 | | | | | | |
| 变更审核 | 变更审查的各个要素，变更审查的结果，审查人 | | | | | | |
| Check in |  | | | | | | |
| 建立新基线 | 检查生成的新基线是否在提供的生命周期数据中已存在，是否标记为未生成，并将其标记为已生成，）新基线的受控配置项库 | | | | | | |
| 基线管理 | 检查该基线是否已在配置项库中，将新生成的基线放入基线库，并改变状态。 | | | | | | |

## 变更内容审查

对变更影响项分析和新生成基线的审查方法

### 4.4.1方法说明

变更管理的关键因素中包括变更影响分析，应该属于变更理中的核心内容，然而对流程的分析无法对具体的配置项中的内容进行审定，来确定其是否符合相关目标要求。因此为了对需求变更影响分析、需求变更后新生成的配置项是否符合追溯性要求进行检查，本位提出了额使用文本相似度加选择排序重建追溯数据进行审定的方法，

该方法适用于对追溯性体现的较为明显的生命周期数据的变更进行验证审查，具体为 系统需求、高级需求、低级需求、代码等。

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，使用文本相似度计算的方法判断系统需求、高级需求、低级。。。间的相似度，并根据计算出的相似度，确定各个层次需求间的追溯关系，将该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的相似度，得到新生成的配置项是否满足追溯性关系，来确定变更实施是否正确。

通过图+说明 表述清楚