目录

介绍

本文件的开发旨在帮助读者确定采用基于风险的方法验证医疗器械质量系统中使用的过程软件的适当活动，该方法应用批判性思维。

这包括 ISO 13485：2016 要求的质量管理系统中使用的软件，生产和服务提供中使用的软件以及用于监视和测量要求的软件：4.1.6,7.5.6 和 7.6。

本文件是努力汇集来自处理执行此类软件验证的医疗器械行业人员以及负责建立可审计文档的经验的结果。本文档针对的是某些问题和难点，在面对医疗设备质量体系中使用的验证过程软件时，我们都会经历如下情况：需要做些什么？要做多少就够了？风险分析如何涉及？经过多次讨论，得出的结论是，在任何情况下，都确定了一系列活动（即来自工具箱的工具），以提供一定程度的信心，以使软件的性能根据其预期用途执行。但是，活动列表取决于多种因素，其中包括软件的复杂性，涉及的危害风险以及供应商提供软件的系统（例如质量，稳定性）。

本文旨在帮助包括制造商，审核员和监管机构在内的利益相关方了解和应用 ISO 13485：2016,4.1.6,7.5.6 和 7.6 中所包含软件验证的要求。

1 范围

本文件适用于在设备设计，测试，部件接受，制造，标签，包装，分发和投诉处理中使用的任何软件，或者用于自动化 ISO 13485 中描述的医疗设备质量体系的任何其他方面。

本文适用于

用于质量管理体系的软件，

用于生产和服务提供的软件，以及

软件用于监测和测量需求。

它不适用于

用作医疗设备的组件，部件或附件的软件，或

软件本身就是一种医疗设备。

2 规范性引用文件

本文档中没有规范性参考。

3 术语和定义

就本文件而言，ISO 9000 和 ISO 13485 中给出的术语和定义适用。 ISO 和 IEC 维护用于标准化的术语数据库，地址如下：

IEC Electropedia：请访问 http://www.electropedia.org/

ISO 在线浏览平台：http：// www，iso，org / obp.

4 软件验证讨论

4.1 定义

术语“软件验证”已被广泛而狭义地解释，从简单的测试到广泛的活动，包括测试。 本文档使用术语“软件验证”来表示确定软件适用于其预期用途的可靠程度的所有活动，并且它是值得信赖和可靠的。 所选择的活动，无论它们是什么，都应确保软件满足其要求和预期目的。

4.2 建立信任活动：工具箱中的工具

在工具箱中的工具（参阅表 A.1 到表 A.5）包括软件生命周期中完成的活动，降低风险和建立信任。

4.3 批判性思维

该文件促进了批判性思维的使用，以确定应该执行哪些活动来充分验证特定的软件。批判性思维是分析和评估软件各个方面及其使用环境的过程，以确定在验证过程中应用的最有意义的一系列建立信任活动。批判性思维避免了一种方法，即不采用全面评估解决方案以确定它是否确实产生了预期结果的情况下，应用一种适用于所有人的验证解决方案。批判性思维认识到验证解决方案在软件和软件之间可能会有很大差异，并且也允许在类似情况下将不同的验证解决方案应用于相同的软件。批判性思维挑战提出的验证解决方案，确保它们符合

质量管理体系要求的意图，并考虑所有关键利益相关者及其需求。当软件的特性发生变化，软件的预期用途发生变化或新信息可用时，批判性思维还用于重新评估验证解决方案。

批判性思维产生了一个验证解决方案，该方案建立了制造商的合规性，确保软件可以安全使用，并产生由审核人员认为合适且充分的书面证据，并导致执行验证工作的个人感受到该努力增加了价值并代表了达到预期结果的最有效方式。

附件 C：介绍了一些示例研究，演示了在各种情况下（包括不同的复杂性，系列和风险水平），如何将批判性思维应用于医疗器械质量系统中使用的软件的验证。

5 软件验证和批判性思维

5.1 概述

在整个医疗器械质量系统软件的生命周期中，需要采取适当的控制措施以确保软件按预期运行。 纳入批判性思维和应用选定的建立信任活动，可以建立和维持软件的有效状态。 Figllre 1 描述了典型活动和控制的概念视图，这些典型活动和控制是从决策到自动化过程直到软件退役或不再用于医疗设备质量系统为止的生命周期的一部分。 尽管图 1 描述了一个顺序模型，但实际上，这个过程具有迭代性质，因为元素被定义，识别风险并应用批判性思维。

在开发用于医疗器械质量体系的软件时，从工具箱中选择一项基本的建立信任活动是软件开发生命周期模型的选择。选择的模型应包括批判性思维活动，以便在各种生命周期活动中选择其他适当的工具。所使用的分析和评估结果推动了最有意义的一系列建立信任活动的选择，以确保软件按预期执行。本文档并不意味着暗示或规定使用任何特定的软件开发模型。然而，为了简单起见，本文档的其余部分在瀑布式开发模型的上下文中使用通用名称解释了批判性思维的概念。只要将批判性思维和适当工具的应用纳入模型中，其他软件开发模型（例如迭代，螺旋）当然可以使用。

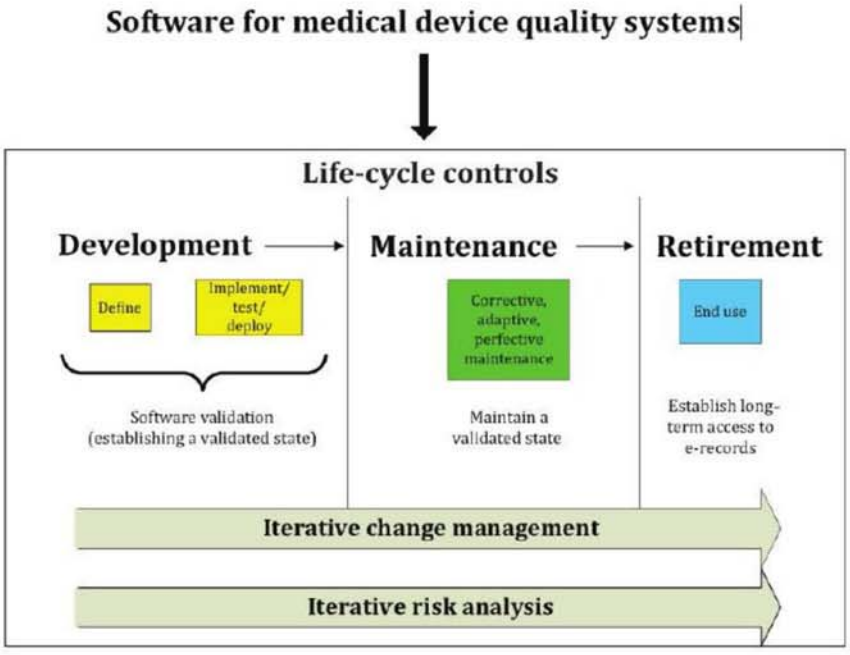


Figure 1 — Life-cycle controls

当考虑在流程中使用软件时，应该通过调查其预期用途来确定拟议软件是否用作医疗器械质量体系流程的一部分。如果是这样，那么应该验证软件的预期用途。虽然本文档描述了

一种验证医疗器械质量系统软件的方法，但同样的方法也是软件评估其是否满足明确要求的良好实践。软件验证的最关键部分是开发/购买正确的软件工具，以便能够按照制造商的预期支持流程。这意味着应准确确定要求以评估开发/购买的软件是否适合满足预期用途的要求。适用于验证的技术要求以及适合验证的过程要求同样重要。当考虑在一个过程中使用软件时，软件可以与其他软件进行交互或可以与其他软件进行接口。

在生命周期的开发阶段，执行风险管理和验证计划任务以收集信息并推动以下四个方面的决策：

对文件和交付品的审查重要程度;

文档和交付内容的范围;

从工具箱中选择工具和应用工具的方法;

应用这些工具的能力水平。

这四个领域的主要决策因素是过程风险和软件风险。 但是，其他驱动因素可以影响决策，包括软件和流程的复杂性，软件类型和软件版本。

验证计划过程由两个不同的元素组成。 第一个确认计划要素涉及确定文件中的严格程度以及适用于审查交付成果的审查。 这一部分的决定主要由过程风险分析的结果驱动。 第二个验证规划元素驱动从工具箱中选择工具来实施，测试和部署软件。 工具的选择主要由软件风险分析驱动。 此类计划步骤源自不同类型的风险分析，并在本文档中作为单独的活动进行描述。 然而，这些步骤多次被合并为一项活动，其中包括风险分析的不同方面以及进行验证的最终选择。

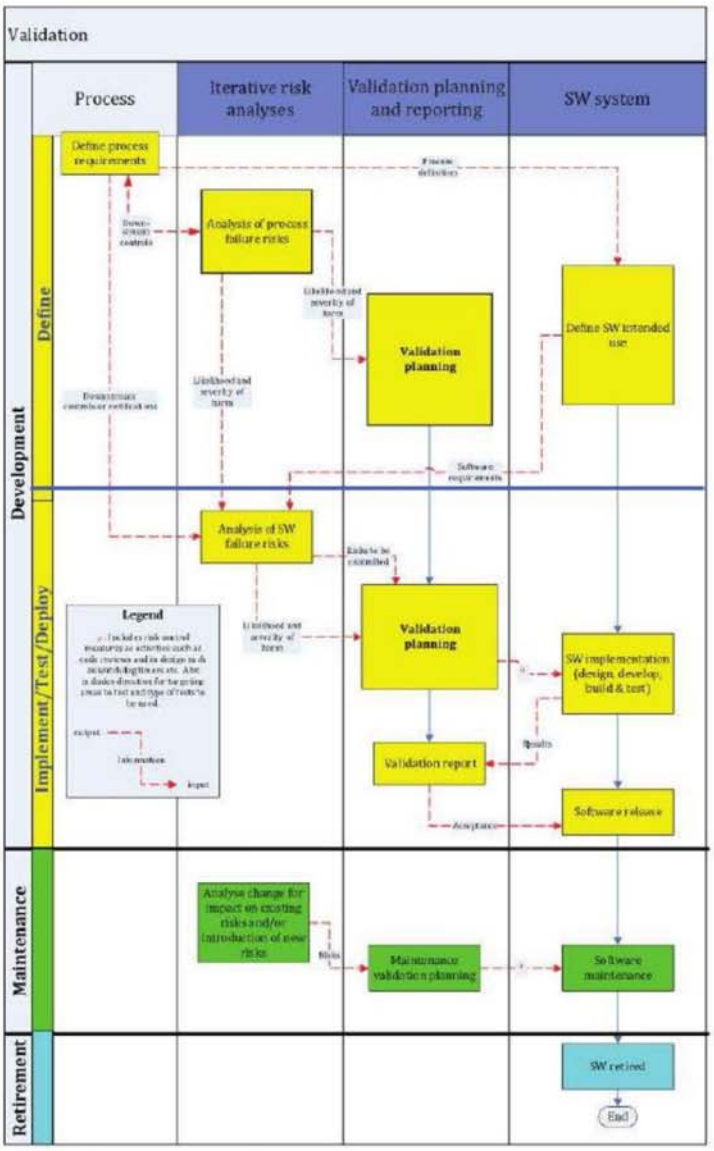
在生命周期的开发阶段，风险管理和验证计划任务用于定义适用于软件的适当努力水平，并确定应用哪些可靠性建模工具。 这种方法导致完成适当的增值活动和验证任务，这是建立验证状态的基础。 一旦执行了这些活动和任务，就会在验证报告中引用工具及其相关结果，以支持软件得到验证的结论。

一旦部署，软件就会进入软件生命周期的维护阶段。 在此期间，根据业务需求或监管要求的变化监控，增强和更新软件。 变更控制活动使用与生命周期开发阶段应用的初始方法相同的概念。 然而，现在对变更进行评估，以了解它们对预期用途，失败风险，初始开发期间应用的风险控制措施以及软件本身的任何功能的影响。

退休阶段是通过删除或替换过程中使用的软件来删除软件的行为。

图 1 所示的活动反映了主要的软件生命周期控制活动。 其他工作流包括项目管理，流程开发，供应商管理（如适用）以及可能的其他工作流程，具体取决于正在实施的软件。

图 2 描述了软件生命周期控制活动和其他工作流活动中的批判性思维。 批判性思维活动出现在迭代风险分析和验证工作流中。 在组织业务模型中对这些工作流进行明确和正式的定义非常重要，以确保从商业和监管两个角度正确管理软件。



NOTE When the term "develop" or "development" is used, it is about the development of a validated state of the software.

Figure 2 — Life-cycle controls work stream

图 2 中描述的各种颜色对应于图 1 中总体方法流程图中所示的生命周期部分。 红色虚线表示从一项活动输出的信息，该信息提供输入或帮助推动另一项活动的决策。 该图演示了在完成需要输入的活动之前，需要输入信息才能驱动活动的顺序。 重要的是要注意，无论正在实施的软件的大小和复杂程度如何，所有活动都已完成。 但是，对于更大或更复杂的软件，这些活动很可能是离散的; 对于更小或更简单的软件，许多这些活动将被同时组合或完成。

总之，批判性思维方法描述了一种系统化的方法，用于在各种工作流中识别并纳入适当的建立信任活动或工具，以支持软件在发布时进行验证的结论，并且验证状态将保持到软件退役。

下面的小节为图 1 中描述的生命周期控制中的每个块提供附加细节。小节使用图 2 中所示的迭代风险分析，验证和软件活动的工作流描述。 提供包含思想的各种决策点和决策驱动因素的视角。

5.2 确定软件是否在范围内

5.2.1 记录软件过程和使用的高级定义

确定软件是否被视为用于医疗设备质量系统的第一步是记录流程的高级定义和软件的使用。 如果知道该软件的范围，并且已经开始定义软件的全部预期用途，则此活动似乎具有很小的价值。 但是，对于这样的假设不太明确的情况，记录过程和使用可以明确确定软件是否在范围内。 另外，对于已识别的范围外软件，这种活动可能会导致软件为什么超出范围。

5.2.2 管理使用评估

监管使用评估可用于确定软件是否为“医疗器械质量体系软件”，因此属于本文档的范围。 首先确定适用于使用软件的流程和由软件管理的数据记录的特定法规要求。 一系列问题可以用来帮助充分理解软件在支持这些规定方面的作用。 应考虑以下类型的问题。

a）软件的故障或潜在缺陷是否会影响医疗设备的安全性或质量？

b）软件是否自动执行或执行法规要求所要求的活动（特别是医疗器械质量管理体系的要求）？ 示例可能包括捕获电子签名和/或记录，维护产品可追溯性，执行和捕获测试结果，维护数据日志（如 CAPA），不合格，投诉，校准等。

对任何问题的“是”答案都表示需要验证并且在本文档范围内的软件。

有时可能很难确定一个过程和相应的软件是否属于质量体系的一部分。 某些工具可能与实际的医疗设备有很多程度的分离。 因此，每个组织应仔细考虑这些边界软件的情况，并应完全理解软件故障对流程的影响，并最终考虑到任何制造的医疗设备的安全性和功效。如果答案不确定，最好的方法是将软件视为范围内并应用本文档中定义的方法。

5.2.3 与医疗器械监管要求无关的过程和软件

当流程或软件包含的功能不符合医疗设备监管要求时，应执行分析以确定软件的哪些部分被认为在范围内，哪些部分不在范围内。 这些决策应根据软件的各个组件，模块和数据结构之间的集成程度以及组织的合规需求进行合理化。 这种合理化对于用于支持质量体系的软件尤其重要，例如大型复杂的企业资源计划（ERP）软件。 ERP 软件可以包含非医疗设备监管流程的功能，如会计和财务。 尽管这些功能对于商业运作至关重要，并且必须满足某些政府要求（例如萨班斯 - 奥克斯利法案）。

5.3 开发阶段

5.3.1 验证计划

验证计划活动的第一部分在应用批判性思维时获得，使用过程风险分析的输入（参见附录 B），以确定应该应用于文档的工作量的基础并推动工具的选择 从工具箱的定义部分（参见表 A.1 到表 A.5）。 第二部分使用来自软件风险分析的输入来驱动工具箱中工具的选择，测试和部署工具。 一旦执行，软件的活动和验证状态就建立起来，并且验证报告中记录了验证的证据。

许多开发生命周期模型可以在开发阶段应用。 本文档不提倡或推荐任何内容; 然而，预计应用受控制的方法。 这种受控制的方法将基于定义要求（包括预期用途）的概念

实施，测试和部署，这对于确定软件的预期用途是至关重要的。

5.3.2 定义

5.3.2.1 定义块要求

在定义块中完成的活动包括过程的定义，该过程中打算使用的软件的定义，以及基于过程中确定的固有风险的验证工作等级的规划。 图 3 描述了所选瀑布模型示例中的这部分开发阶段

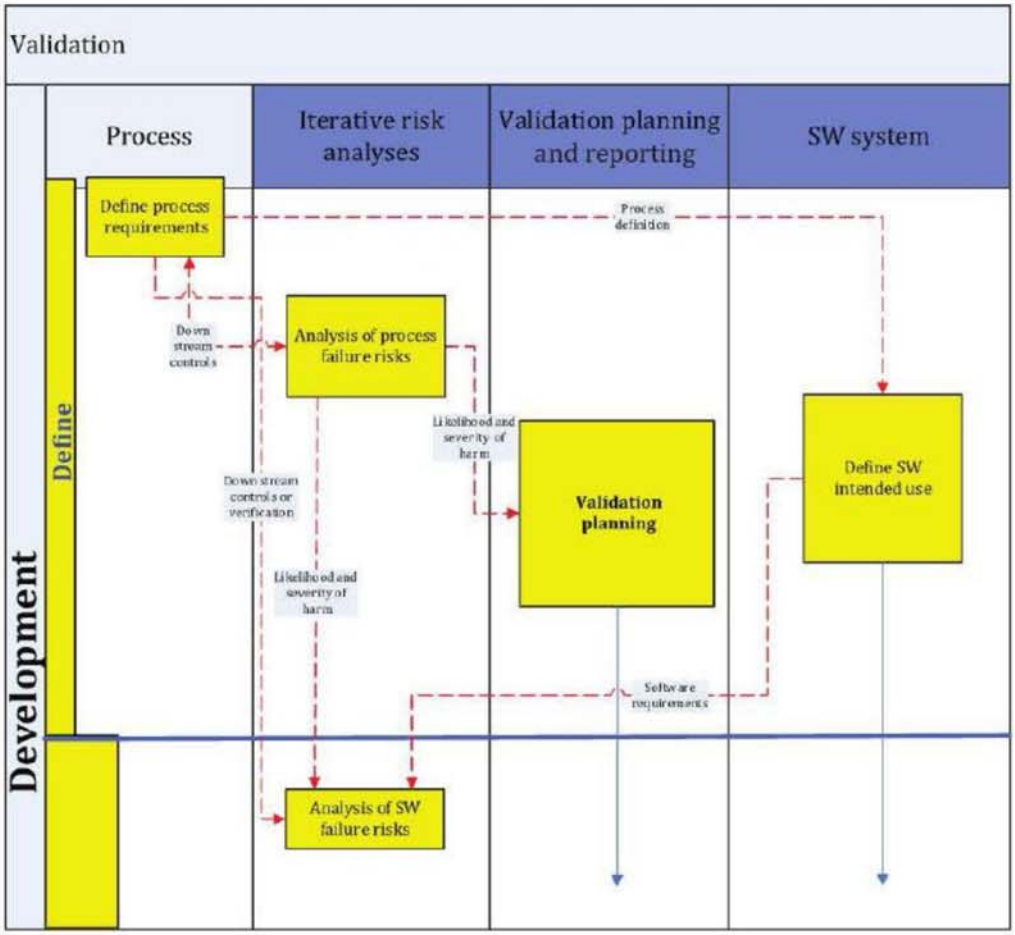


Figure 3 — Life-cycle phase: Define block work streams

图 3 - 生命周期阶段：块定义工作流程

5.3.2.2 过程要求

生命周期控制应用的第一步是定义整个过程的目的和功能，特别是要由软件控制的部分。最好的做法是让适当的主题专家参与进来，包括所有相关的方面和活动，而不管所有方面是否都由软件控制。 好处解释如下：

- 监管要求可以清楚地看出;
- 在过程的上下文中对特定软件的预期用途可以清楚地看出;
- 过程方面和不受特定软件控制的活动都可以清楚地识别并通过程序或其他方式解决;
- 确定软件上游和下游的过程活动，并在评估软件故障风险和设计软件故障风险控制时考虑。

流程定义活动为生命周期后期做出的决策奠定了基础，对于针对增值型，基于风险的活动的努力至关重要。

5.3.2.3 分析过程失败风险

在风险分析过程中将考虑软件与医疗产品的最终安全性和功效之间的关系。 以下也应该考虑。

- 危害人类的风险：这包括直接伤害患者和用户，以及当控制制造或软件质量的软件发生故障时造成间接损害，导致设备故障，从而导致危害。
- 监督风险：如果软件故障可能导致监管机构要求的记录丢失（例如，CAPA，投诉，

设备主记录或设备历史记录文件记录）或偏离质量体系，则需要考虑不符合监管要求的风险 和制造程序。

- 环境风险：对软件运行环境的风险。 物理和虚拟。

其他类型的风险可以纳入该模型。 但是，本文档的范围和讨论的降低风险的工具无法解决这些问题。 本文件着重于确定在过程失败情况下与软件故障相关的人身安全风险，监管风险和环境风险。

应该明确记录风险分析的结果，因为这些结果是从工具箱中选择工具的有价值的决策驱动因素，也是验证应用于验证活动的有效程度的合理性因素。

5.3.2.4 验证计划

确保软件的要求能够一致地实现所需的确认程度和客观证据取决于整个过程中软件的关键价值。 因此，第一次关于所应用努力水平和可交付成分审查的验证计划活动完全基于过程失败风险分析的输入。

此验证计划活动导致验证计划文档的第一次审核。规划包括选择“努力程度”（即决策）和这些选择的基本原理（即决策驱动因素）。 理由应该基于过程失败造成的危害风险。 验证计划应提供批判性思维应用于验证计划过程的客观证据。

5.3.2.5 软件预期用途

5.3.2.5.1 预期用途的要素

预期用途旨在提供过程中软件功能及其用途的完整画面。 具体而言，它是为了描述和解释软件如何适应自动化的整个过程，软件的功能，软件的期望以及软件的设计，生产和维护的安全程度 医疗设备。 预期用途是用于了解与使用该软件相关的潜在风险的关键工具。预期用途的三个主要要素是：

- 目的和意图有关
  - 软件的使用（例如，谁，什么，何时，为什么，在哪里以及如何），
  - 软件的监管使用，以及
  - 过程中软件或其他软件和/或用户的边界;
- 软件使用要求。 随着复杂性以及风险的增加，此要素将增加关于软件使用的更详细信息（例如用例，用户需求）;
- 软件要求。 随着复杂性和风险的增加，软件的实施者应该有明确的方向，这个要素提供了关于软件期望的更具体和更详细的信息。

预期用途应由正规的技术和经验丰富的人员在法规，质量体系和被控制的过程中正式控制和批准。

鉴于我们应该对“预期用途”进行验证，除非对软件的预期用途进行了充分定义，否则无法进行验证。

以下小节提供了有关软件预期用途的更多细节。

5.3.2.5.2 软件的目的和意图

它包含三个要素的信息：软件使用，监管使用和边界定义.

a） 软件使用

- 在定义软件的使用时，应该考虑以下问题：什么，为什么，如何，谁，何时何地。 答案探讨了如何使用软件来满足过程要求。 这种探索有助于确定软件定义的基本信息，如表 1 所示。
- 对软件描述有意义的答案应包含在既定的预期用途定义中

表 1 - 示例问题和答案

问题	回答
软件解决了什么问题？	有效和准确地汇集产品缺陷数据以达到趋势目的存在问题。
为什么该软件有用？	该软件支持来自全球各地的数据汇集和趋势分析
软件如何解决问题？	该软件驱动数据收集过程，并自动汇集和计算趋势信息，或者该软件不驱动该过程，但提供用于汇集和计算趋势信息的被动收集数据
谁使用该软件？	质量保证和运营部门使用该软件。
软件在哪里使用？。	该软件由美国，欧洲和日本的地区访问。
何时使用该软件？	该软件在正常工作时间内访问全球位置（即每天，周一至周五）。
注：这些样本问题并不详尽。	

b）监管使用

- 在评估监管使用时，可以进一步探讨回答的问题，以确定软件是否在范围内（见 5.2）。扩展所有“是”的答案以包括这些结论的原因。现在软件已被确定为范围 ，任何对人类（医疗设备的使用者除外）或对环境的潜在危害都需要确定，以下所有问题都将引导用户考虑作为法规要求的要素，例如公共卫生，安全和 电子记录和签名的有效性或真实性。
  - 软件的失败或潜在缺陷如何影响医疗设备的安全性或质量？
  - 软件如何自动化或执行法规要求的活动，特别是医疗设备质量管理体系的要求？
  - 软件如何对人员（医疗设备用户除外）或环境造成伤害？

c）软件边界

- 通过软件（过程中的界限）和软件接口存在的地方（与其他软件的界限）来定义要控制的过程部分有助于验证工作的有效性和效率。 例如，将多个软件产品作为一个组进行验证通常会更高效，而不是执行单个验证。 还应该考虑各种分组策略如何影响正在进行的维护活动的效率。
  - 过程中的边界
  - 在流程中识别软件的边界清楚地确定了要包含在预期用途中的方面。 软件可以自动执行整个过程，也可以自动执行一部分活动，也可以用作过程数据的存储库。 了解软件在该过程中扮演的角色有助于确定与软件潜在故障相关的风险。
  - 与其他软件的界限
  - 当与其他医疗设备质量系统软件或医疗设备软件进行外部接口时，识别应用程序之间的所有接口非常重要。 验证工作通常包括将内部接口作为方法的固有部分，但不应忽视软件的外部接口。 软件应用程序之间的所有界面都应该纳入批判性思维过程。

5.3.2.5.3 软件使用要求

软件使用要求包括记录良好且可追溯的元素，为软件的目的和意图提供额外的细节层次。从用户角度或产品需求角度来看，这些要求可以深入了解系统的使用情况。 用户的观点可以以用户需求，用例或其他以用户为中心的需求定义的形式来捕获。 产品需求视角捕捉受系统影响的医疗设备的需求，并且在某些情况下，可以包括对特定设备要求的参考或软件可能影响的产品系列的摘要。

5.3.2.5.4 软件要求

由定义要素的活动组成，有详细记录和可追溯的内容，明确软件需要做什么才能达到其目的，意图和使用要求。 软件需求包括系统设计的输入，系统的配置以及测试活动的输入。

5.3.3 实施，测试和部署

5.3.3.1 所需的活动

实现，测试和部署块内完成的活动包括

a）规划设计中的验证严密程度，

b）开发和配置，

c）建立软件，并且

d）根据所识别的风险对软件进行测试。

5.3.3.2 分析软件故障风险

软件故障风险分析的关键在于确定和记录与软件故障相关的固有风险，并确定任何控制措施（包括分析中软件之外的过程和软件控制）。 然后使用分析来确定一个现实和有效的验证方法。

在审查可归因于软件故障的风险时，我们考虑构成风险控制措施的分析软件之外的任何过程控制。 这种风险控制措施可以减少软件故障的影响，从而减少对软件的依赖，从而减少对测试（检查）和文档（收集客观证据）的依赖性，从而确保软件的安全运行。 包括这些考虑因素将有助于确保在整个过程中查看软件。

附件 B 中提出的模型并不代表一个包罗万象的公式。 由此产生的分析为从工具箱中选择用于软件验证的工具提供了输入。

5.3.3.3 验证计划

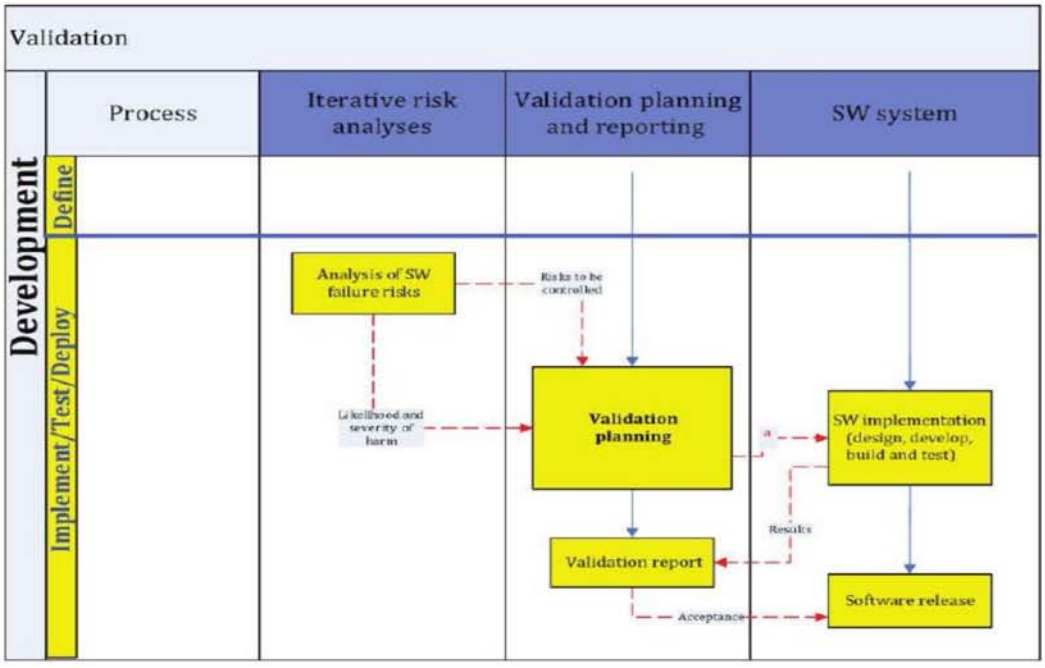
本活动使用预期用途定义和软件风险分析结果作为识别风险控制措施的输入，以及从工具箱中选择将用于验证软件的工具。

工具选择过程中必须包括合格的人员，他们了解失败对流程的影响以及软件失败的固有风险，以使流程实现自动化，但个人不必是软件专家。 任何来自不同学科（监管，质量，临床等）的个人都应参与规划过程，以处理任何高度复杂或与其失败相关的高风险软件。

验证计划活动产生一个文档化的计划，描述做出的选择（决策）和选择的原因（决策驱动程序）。 验证计划提供了用于选择增值建立信任活动的基本原理的书面证据，用于确保软件按照预期执行

5.3.3.4 软件实施（设计，开发，构建和测试）

该块包含工具箱中许多工具的实际应用。 工具（验证计划中要求的活动）在设计，开发，构建和测试步骤中进行（见图 4）。



a 包括风险控制措施，如代码审查等活动，以及诸如看门狗定时器等设计。还包括指定测试区域的方向和要使用的测试类型。

图 4 - 生命周期阶段：实施，测试和部署块工作流程

5.3.3.5 验证报告

一旦完成了包括从工具箱中选择的工具在内的充足的建立信任活动，以确保软件按预期执行，活动和（可能）活动结果应在最终验证报告中引用。 报告的正式审核和批准提供了对所有已形成文件的客观证据的参考摘要，该证据支持软件已被验证用于其预期用途的结论。

5.3.3.6 软件版本

验证结束后，应该有正式的受控方法来释放软件。 所定义的控件应确保并确认投入使用的软件与通过验证报告中引用的建立信任活动的评估软件相匹配。 否则，基本原理和控制应确保并确认结果充分代表了发布软件在其预期环境中的性能。

5.4 维护阶段

5.4.1 进入维护阶段

阶段入口标准：在软件发布使用后，软件维护阶段开始。 活动范围：维护阶段活动包括确保软件保持在验证状态，同时适应，管理和控制各种类型的更改。 某些类型只能涉及对软件使用过程的更改。

根据政策和程序，应以受控方式对任何经过验证的系统进行更改。

理想情况下，建议在测试环境中进行更改并在将系统提升为生产使用之前进行验证。在验证测试环境中的更改不可行时，应在生产环境中测试更改，应采取适当的控制措施，以尽量减少对生产环境或直接影响产品的不利影响。

工具箱中选择哪些工具用于验证变更应该通过引入新风险对现有风险控制措施进行软件变更的影响分析或两者兼而有之。

由于软件或其配置的实际使用可能会随着时间的推移而迁移，尽管需要对其进行控制，但使用维护阶段专用工具(例如实际使用的定期监视或软件配置的实时监视)可能是合适的。如果预期使用的变化导致更高的风险水平，则即使没有对软件进行更改，更改也会引发比最初执行的更广泛的验证活动。

关于执行更广泛的验证活动的选择和证据的决定应作为验证计划的一部分进行记录，以提供证据证明软件仍处于验证状态。

5.4.2 规划维护

维护计划证据应在开始维护阶段之前记录。

理想情况下，维护计划在开发阶段开始。 人们应该正确理解变化如何影响软件验证，检查变化对风险的影响，并规划适当的活动以维持验证。

大型复杂的软件可能必须适应日常维护和性能调整活动，而不会影响软件按预期执行的能力。 在开发阶段对维护进行规划可以确定哪些操作活动可以完成，而不会影响验证，哪些更改需要验证工作。 在软件到达维护阶段之前，应规划和讨论确定何时对软件执行进一步验证活动的方法，包括底层软件（例如操作系统，数据库管理系统）的变化如何影响已验证的软件。 培训软件操作员识别这些边界并识别正常操作活动与需要验证的任何更改之间的区别是有帮助的。

可追溯性分析是管理维护活动的有用工具。可追溯性分析通常是初始验证的基石，并且通常通过可追溯性矩阵来促进。矩阵映射测试或其他验证活动的要求，风险控制措施等等。如果在初始实施过程中表现良好，可追溯性分析将成为维护过程中的重要工具，方便识别变更影响和确认变更的适当活动。在简单的软件中，这种分析可以包含对实现和验证的单一要求。但是，复杂的软件可能需要一个多层矩阵，将顶层功能分解为较低层需求，然后再分解为实现和验证。还可以嵌入其他信息，例如，可以在追踪矩阵内指定被认为特别高风险的软

件部分，可能还会显示附加的验证活动。

5.4.3 维护阶段的维护类型

软件在发布后会发生变化的原因有很多。 一些更常见的维护更改类型包括以下内容：

- 纠正维护更改以纠正软件中的错误和故障;
- 完善的维护更改以提高性能，可维护性或其他软件属性;
- 进行适应性维护以更新软件操作环境（例如，对操作系统，系统硬件或软件所连接的其他应用程序的更改）。

5.4.4 流程变更：改变风险控制措施

当整个或部分受软件控制的过程发生变化时，应该进行影响分析以重新评估风险控制措施。

完全或部分由软件控制的过程可以独立于软件进行更改。 当发生过程更改时，了解该更改如何影响软件的已验证状态很重要。 过程更改可能会影响软件的预期用途或有关软件的其他支持信息。

流程变更也会影响作为验证原理一部分的软件的风险控制措施。 由于软件是过程的一部分，下游控制可能是软件的重要风险控制措施。 如果将下游控件正确识别为软件验证基本原理和过程定义的一部分，则对提议的过程更改的影响分析将更容易执行。 影响分析对于以对软件和软件运行过程建立信心的方式进行维护至关重要。

5.4.5 紧急变更

应急更改应由批准的流程进行管理。 这些过程应该要求制定和实施的理由，获得和记录授权部署变更的机制，确保风险得到适当评估和控制的规定，以及援引紧急变更所需的任何活动（例如培训， 沟通，产品评论和处置）。 在这种情况下，执行适当评估和控制风险的规定代表了满足在发布之前验证变化的监管要求所需的最小活动集。

软件更改可能需要在紧急情况下执行。 通常，如果软件，操作系统或数据的完整性受到损害或有助于缓解潜在的有害情况，则需要进行此类更改。

另外，可能需要紧急情况后的变更活动来充分评估变更的所有影响。 根据流程失败造成的整体风险，流程输出（数据或产品）可能需要额外的控制措施，直到所有紧急情况后的变更活动完成。

中断过程的软件问题通常很明显。 检测细微的潜在问题可能会更困难。 对错误日志，帮助中心请求，客户反馈和其他缺陷报告进行定期评估可能会指出潜在的问题。 这种监测技术可以挑出不够明显的错误报告，但可以指出可纠正的软件问题。 维护活动可能需要通过在未来版本中实施更正来处理已发现的问题。 此外，可以主动管理发布的软件中归因于这些类型的软件问题的问题。

在维护活动纠正未来版本的问题后，应审查已发布软件中发现的缺陷的历史影响，并对其后果进行管理。

如果软件验证依赖于通过培训确保软件的正确使用，则定期评估用户评估的有效性是另一种有助于维持已验证状态的监控技术。

5.4.6 维护用途

如果软件的预期用途发生变化，则应针对新的预期用途进行验证，否则应停止使用新的用途。 在后一种情况下，进行评估是为了确保在未经授权的使用期间不会引入风险。

预期用途的变化是一个需要特别关注的类别，因为变化可能很微妙，很难察觉，或者很明显。 在细微的情况下，目的和意图或软件使用要求发生变化，并不一定会导致详细的软件要求元素发生变化（见 5.3.2.5）。 这种改变可能是故意发生的，也可能是因为仅仅使用现有的软件而没有意识到预期的使用受到影响。 预期的使用可能会随时间推移而迁移，或者用户可以以最初并非打算使用的方式开始使用该软件。 由于这种转变，部署的软件不再处

于验证状态。

每次对已验证的软件进行更改时，都应审查预期用途，以确保其与软件的实际使用情况保持一致。

5.5 退休阶段

在退休阶段，应该记录软件的退役情况，并建立在任何需要的记录保存期内访问任何相关电子记录的方法。

软件退休活动高度依赖于退休软件的类型。 某些软件只是执行一项活动而不存储任何数据。 其他软件可能与不可追踪性或文档控制系统一样复杂，其中包含大量与产品相关的和合规性相关的数据。 在存储数据的软件的实例中，应该存在用于处理数据的计划。 一些需要考虑的问题包括以下几点：

- 那里有软件取代退役的软件吗？
- 数据可以迁移到新软件吗？
- 数据是否应该迁移到便携式格式以便长期保留？
- 数据类型的数据保留要求是什么？
- 数据会存储在持久媒体上吗？
- 如果是这样，那么存储说明或程序是什么，并且可以检索包含所有相关数据要求的数据？
- 维护可以读取它的持久性媒体和软件的过程是什么？
- 将存储归档的硬件平台以便使用和检索退役的应用程序？
- 存储的硬件将如何维护？
- 作为投诉或 CAPA 调查的一部分，是否需要访问退役的软件？
- 是否需要平台和应用程序来重新创建软件程序？

6 文档

应确保所有与软件生命周期控制活动相关的信息都得到适当的记录和控制。

拥有高质量和高效率的文档有两大好处。

a）文档中明确阐述的完整软件定义能够全面了解软件的预期用途和预期性能，并且可以帮助您理解对软件所做的任何和所有更改的全部影响。

b）验证计划和执行记录提供了作为批判性思维结果而作出的决定的书面证据。 围绕所执行的评估或分析以及针对基于风险和有意义的建立信任活动的结果工具选择聚焦本文档，可以简明了解所执行的验证。 通过总结如何满足验收标准，文档提供的证据表明，已完成的活动确保软件按预期执行，并为其自动执行的过程带来可接受的风险级别。

所产生的文件的范围直接关系到软件验证的努力水平。 努力水平应该与风险相称。 因此，本文档中讨论的软件验证方法基于文档对流程失败影响的程度。 该过程对人员或环境造成的危害越大，文件的预期程度就越大。 此外，更高的危害风险应该促使多个跨职能同行对文档进行更高层次的审查，或者公司内部管理层级更高或两者兼而有之。

将生命周期控制信息组织成文档可能会因许多因素而异，例如所使用的技术以及软件的大小或复杂程度

信息的组织方式应该有利于审核信息，并且能够在软件生命周期的维护阶段保持有效状态的证据

如何捕获和记录生命周期控制信息取决于执行验证的各方的偏好和既定政策。 谨慎对待验证软件的各方如何将生命周期控制的客观证据打包并呈现在文档中。 从合规审查的角度来看，应制定验证计划和报告文档，以汇编所有计划和执行的增值建立信任活动，以确保软件按预期执行。 从本质上讲，本文档是基于投入（决策驱动因素）做出选择（决策）的关键记录，体现了批判性思维过程，这些过程用于确认已制定完整的软件解决方案，符合法

规的意图并考虑 所有关键利益相关者及其需求。

注：术语“文档”用于指被记录的信息主体，无论它是记录在实际文档中还是记录在捕获信息的工具中，例如需求管理工具。

7 先决条件流程

本文件中介绍的方法旨在充分运用有效的质量管理体系，以提高效率。

质量体系对批评性思维方法的成功产生最积极影响的方面包括资产和基础设施管理(人力和硬件)，变更管理（包括配置管理）和供应商管理。 详细说明这些方面不在本文的范围之内; 在行业内的其他标准和文件中涉及每个方面（参见参考书目）。 此外，本文件不打算将特定角色或职能（例如质量保证，管理和制造）与本文档中的活动相关联。 每家公司的理念和人力资源基础设施将决定执行验证活动的可接受角色。

附录 A（资料）
工具箱

A.1 总则

这个工具箱提供了一系列建立信任活动，可以用来满足验证要求的意图。 这并不意味着为此目的而详尽列出了可用的活动，但它提供了一个基于当前软件工程知识体系的初学者集合。 其中一些活动重叠或一起工作，例如，正常情况下的测试通常是软件系统测试的一部分，但重点在于活动的价值。 这些活动将被用作验证计划和执行的基础。

选择和开展活动应该适合与软件相关的风险。 为了支持这一选择，工具箱中的活动按照以下方案进行分类和标记。

- 全程：在任何情况下进行此项活动。
- 裁剪：选择并执行此活动的适当部分。
- 选择：在适当的地方选择并执行活动。.

可以定制工具箱来定义组织中使用的活动，并随着技术的变化和学习经验而随时间发展，从而整合新的软件工程最佳实践。 在适用的情况下，一些活动也将在标准程序中以程序方式提出。

A.2 工具箱结构

为了方便起见，这些活动被组织为五个主要的软件生命周期过程活动。 根据软件的范围和性质，应该在软件生命周期的不同阶段应用批判性思维，以确定和选择最适合软件的活动。

对于列表中出现的每个指定活动，都有一个简短的定义和活动对验证工作做出贡献的值的描述。 定义块还包含可用来完成指定活动的方法示例。

表 A.1 - 发展阶段：定义

活动	定义
工艺要求定义（全程）	针对软件，制造过程或质量系统过程部分或全部自动化考虑的过程定义的活动。 活动还描述了在执行流程或软件风险分析时可以考虑的流程内的任何验证或预防措施。 此活动的输出可以记录在定义业务，制造或质量系统过程中执行的活动的流程示意图或需求声明中。
过程失败风险分析（全程）	确定过程失效对设备安全性和有效性，制造人员，环境或质量体系的影响的活动
预期用途（裁剪）	简单软件的活动可以由几个句子或段落组成。 对于大型复杂软件，活动可以包含多个文档的大量文档，并可能包含详细的软件要求。风险也是决定预期用途定义深度的重要因素。 预期用途的要素： • 软件的目的和意图; • 软件使用要求; • 软件要求。
验证计划（全程）	验证计划分两个阶段进行： 在确定验证文件中预期的详细程度和努力水平的开发确定阶段，审查的水平和选择将包括在确定阶段的活动; 之后在实施阶段，根据定义阶段和相关风险分析活动所做的决定选择适当的验证活动。 验证计划的输出是一个计划，描述将执行的活动，以确定软件始终如一地满足其预期用途的要求。
正式软件需求评审（选择）	活动（过程，会议等），利益相关方根据预期用途审核并达成软件要求。
软件开发生命周期模型选择（选择）	在整个软件生命周期的开发部分中用于定义生命周期方法和控制的活动。 通常只需要复杂或有风险的软件。 IEC 62304：2006 / AMDl：2015 可能特别适合作为某些软件的过程标准。
风险管理计划（全程）	与计划如何执行软件风险管理相关的活动。 风险管理计划的输出是一个计划，用于分析软件相关风险领域的分析方法，以及分析风险的方法选择，例如故障模式和影响分析（FMEA），故障树分析或 其他工具。
在制造或业务流程中识别风险控制措施（全部范围）	该活动是确定控制风险或危害的措施（例如程序控制）的机制。 它包括持续监控以确保控制装置正常工作并正常工作。

表 A.2 - 开发阶段：实施

活动	定义
软件故障分析（风险分析）（全程）	软件故障分析是指确定软件故障相对于过程的影响以及过程故障分析中确定的关注领域。
软件体系结构文档和评论（选择）	软件体系结构定义了软件的高级结构以及它们之间的关系，记录体系结构并检查软件功能的正确性，完整性和能力。
设计规格（选择）	设计规范是如何实施软件要求的精确声明。 它通常包括软件或组件结构，算法，控制逻辑，数据结构，数据集使用信息，输入和输出格式，接口描述等。
开发和设计审查（选择）	开发和设计评审是为评估所选设计方法在一个或多个配置项目上的进度，技术适宜性和风险解决方案而进行的评估。
在软件设计中识别风险控制措施（全程）	本活动确定了控制风险评估过程中发现的风险或危害的措施。 风险控制措施的识别应该是一个反复的过程，以允许持续监控并确保控制措施到位并正常工作（例如程序控制，硬件冗余）。
代码审查或代码验证（选择）	代码审查或代码验证包括对软件源代码的同行评审，旨在发现并消除缺陷并提高整体代码质量。 通过建立和遵循一套通用的编码标准，可以增强代码评审和整体代码质量。
可追溯性分析（选择）	可追溯性分析是指设计，编码，测试，风险或危害分析以及风险控制措施的可追溯性。 它也可能包括对工艺要求的可追溯性。
供应商审核（选择）	供应商审核意味着对软件供应商系统的评估达到必要的水平，以确保购买者供应商能够充分提供安全可用的软件。 各种供应商审计方法都是可能的。

表 A.3 - 开发阶段：测试

活动	定义
测试计划（选择）	测试计划应定义测试活动的整体方法，以帮助建立软件满足其预期用途的信心。 但是，软件测试本身可能不足以确定该软件适合其预期用途的信心。 其他验证技术可能需要与测试结合以确保全面的验证方法。 测试水平应基于风险驱动因素和因素，并应提供适当的可信度来证明软件符合相应测试方法的要求和设计规范。 这种测试可以包括开发人员测试，测试，集成测试，用户测试，负载测试，操作测试等。
单元测试（选择）	进行测试以验证一个软件元素（例如单元或模块）或一组软件元素的设计的实现。
数据验证（选择）	数据验证是指为确认数据的正确性而完成的活动。 它可以作为数据迁移，转换或测试工作的一部分或独立完成，并且可以包括适当的统计抽样。
集成测试（选择）	集成测试是一种有序的测试过程，其中将软件元素，硬件元素或两者结合起来并进行测试，以评估它们之间的交互，直到软件被集成为止。
使用测试（选择）	用例测试是一种功能测试形式，忽略了系统或组件的内部机制或结构，并侧重于响应所选输入和执行条件而生成的输出。 每个用例可以具有与其相关的输入参数，每个参数可以具有一组用于模拟实际使用条件的值。 可以使用预定流程来连接一系列用例，这些流程描述了实现某个目标的顺序。
接口测试（选择）	接口测试是指确认软件应用程序之间的接口，并考虑从输出到输入的整个数据传输路径。 接口测试可以通过直接测试或 100％数据验证来完

	成。 测试活动应包括确保界面在规范限制或边界条件下正常和异常情况下按要求执行的策略。
回归测试（选择）	重新运行程序以前正确执行的测试用例，以检测软件开发和维护期间所做更改或更正所产生的错误。
供应商提供的测试套件（选择）	供应商提供的测试套件可以测试软件解决方案的全部功能，并且可以为最终使用环境中的软件性能提供充足的信心。 然而，应评估这些套件是否适合所定义的预期用途和测试的完整性，包括测试是否存在任何风险控制措施。 使用这样的套件可能需要合同协议，要求供应商在软件的使用期限内维护测试套件。
软件系统测试(选择)	软件系统测试是对集成硬件和软件系统进行测试以验证软件是否满足其特定要求的过程。 这种测试可以在开发环境和目标环境中进行。 软件验证与软件系统测试不同，因为软件验证验证软件适用于其预期环境以及预期用户的适用性。 软件系统测试只验证对软件的要求已经成功实施。 对于由软件控制的生产系统，过程验证测试可以涵盖部分或全部这些测试。 对于高质量的系统应用程序，执行软件工作指令所需的所有步骤可以覆盖软件测试要求。
使用测试（选择）	我们的案例测试是指根据用例进行的测试，包括软件使用案例中定义的替代流程和错误条件。
正常情况下的测试（选择）	正常情况下的测试是用普通输入进行测试
强度测试(压力测试)（选择）	意外性测试应该证明，当给出意外的无效输入时，软件产品的行为是正确的。 它的目的是评估一个系统或组件是否超出其特定要求的限制。 识别足够的这种测试用例的方法包括等价类划分，边界值分析和特殊情况识别（错误猜测）
强制输出测试(选择)	输出强制测试意味着选择测试输入以确保所选（或全部）输出由系统正确生成。 输出强制涉及制作一组测试用例，用于从系统产生特定的输出。 重点在于创建所需的输出，而不是启动系统响应的输入。
组合输入测试(选择)	组合输入测试是一种测试技术，通过这种测试技术，软件单元或系统在操作过程中可能遇到的输入组合被执行。
Beta 测试（选择）	Beta 测试正由供应商针对一小部分客户在现场环境中进行测试。
性能测试（选择）	性能测试可以测量软件系统按照其要求的响应时间，中央处理单元（CPU）使用情况以及运行中的其他量化功能执行的情况。

表 A.4 - 开发阶段：部署

活动	定义
用户程序审查（选择）	用户程序审查是对与使用该软件相关的用户程序和说明进行审查。 这样的审查可以确保正确定义软件的使用。
内部应用程序培训(选择)	内部培训是指针对软件特定的文档化培训活动。
安装认证（选择）	安装认证意味着根据文档安装说明建立软件安装和运行的信心。
操作和性能鉴定(执行过程验证时)（选择）	操作认证确信制造过程和相关系统能够始终在规定的限制和容差范围内运行。

	性能认证确定了过程的有效性和可重复性。
最终验收测试（选择）	最终验收测试是指在最终部署之前应用于系统的测试。 它也被称为上线测试。
操作员认证（选择）	操作员证明是确认受过培训的个人在培训中显示出胜任力的证据。

表 A.5 - 维护阶段

活动	定义
维修计划（裁剪）	与维护计划相关的方法如下。 远期规划。 此方法涵盖了前瞻性计划和对软件更改的预测。 在进入维护阶段之前，可以在软件的初始实施期间使用此方法，但在维护阶段的任何时候都可以使用此方法。 计划未决更改。 此方法涵盖了软件更改未完成时的计划。 计划通常侧重于特定于待处理更改的活动。 这个计划是在软件维护阶段完成的。
已知问题分析（选择）	已知问题分析是评估供应商已知的软件的任何和所有问题的过程，以评估它们对已安装软件的使用或已验证状态的影响。
兼容性测试（选择）	兼容性测试是确定两个或更多软件系统交换信息的能力的过程。
基础架构兼容性分析（选择）	基础架构兼容性分析是确定软件基础架构更改如何影响安装的软件的过程。 这些更改可能包括硬件更改或系统位置更改
系统监控（选择）	系统监控包括用于在软件生命周期的维护阶段评估软件系统的总体健康状况的技术。 系统监控的方法可以包括以下内容： • 定期评估预期用途是否已经改变; • 最终用户的实际使用; • 培训效果评估; • 缺陷分析; • 数据审计。
备份和恢复过程（选择）	备份和恢复过程包括系统备份，备份介质的存储和保留以及从备份介质恢复数据的恢复过程。
操作控制（选择）	除了备份和恢复过程，监控和报告外，还可以使用操作控制来帮助确保软件按预期运行。 常用方法包括以下几点： • 安全; • 访问权限管理; • 数据库管理; • 归档; • 应急计划。
回归分析（选择）	回归分析包括诸如影响分析的可追溯性分析等任务。 它旨在确定维护系统有效状态所需的活动。

附录 B
（资料）
风险管理和基于风险的方法.

B.1 总则

正如本文档的核心部分所述，验证的内容和严格性由与软件相关的风险决定。

为了扩展这个概念，参考 ISO 14971. ISO 14971 描述了一个应用于医疗设备的风险管理流程。 但是，基本原则以及术语可以应用于符合 ISO 14971 的软件。

B.2 术语

下面列出的定义要么取自 ISO 14971，要么取决于 ISO 14971 中的定义。

- 危害：可能的伤害来源;
- 危险情况：人员，财产或环境暴露于一种或多种危害的情况;
- 风险：将伤害发生的可能性与伤害的严重程度相结合;
- 伤害：人身伤害或人身健康损害或财产或环境破坏;
- 严重程度：衡量危害可能造成的后果;
- 风险控制措施：将风险降低到或维持在指定水平的措施。

B.3 基本原理

基本原则是将与软件相关的风险降低到可接受的水平。 为了实现这一点，制造商需要使用软件识别可能的危险情况，估计相关的风险并评估这些风险是否符合接受标准，否则，例如， 按规定，由制造商定义。

特别是由于软件本身无法对其造成损害，因此受软件控制的整个过程都需要进行风险管理。

B.4 识别危险情况和估计风险

按照 ISO 14971 的方法开始使用，应该确定可能的危害和危险情况，并估计相关的风险。但是，考虑到可能造成的危害与 ISO 14971 中正在考虑的有很大不同。

生产和质量系统的故障很少导致对其制造或质量受软件控制的医疗设备的患者或用户直接伤害。 这种情况下的危害几乎总是间接的。 对设备造成的危害最终会成为患者或设备用户的伤害源。 这并不是说间接损害在任何方面都不那么严重。 事实上，在某些方面，生产和质量体系失败的严重程度可能会被认为更严重，因为这些系统中的单一故障可能导致许多设备出现故障，最终在检测到许多患者之前对其造成伤害。 一台设备中的软件故障一次只能损害一名患者。

直接和间接的多重危害都可能来自生产或质量体系的失败。 请注意，下面列表中的危害并不相互排斥。 每个人都有可能对医疗设备的患者或用户造成间接伤害。 示例包括但不限于以下内容：

- 对医疗器械的伤害：
  - 机床不会产生严格的误差;
  - 校准系统错误地校准药物输送装置;
  - 消毒器控制器以未消毒组件生产的方式失败;
  - 最终测试系统的故障不能检测到潜在的设备缺陷;
- 对制造过程的危害：
  - 由于使用了手动变通方法，软件控制的过程失败会降低生产速度;
  - 软件驱动的过程失败会造成高比例的超差部分;
- 对法规遵从的危害：
  - 投诉处理系统错误地报告失败统计数据，从而允许现场报告的缺陷未被发现;
  - 设备服务或维修系统未能突出显示可能指向以前未检测到的缺陷的问题趋势;

- 植入设备的数据库发生完整性损失;
- 发生与制造物品安全检查相关的质量控制记录丢失;
- 合规数据丢失发生;
- 设备验证数据丢失;
- 无法控制和报告制造设备中的软件配置;
- MRP 系统未能提供可追溯性导致未能通知潜在用户设备安全召回;

● 对制造人员或环境的危害：

- 操作员受伤;
- 有毒化学物质被释放。

在分析与依靠软件实现生产和质量体系自动化相关的风险时，应考虑所有类别的危害。
风险估计包括估计可能造成的伤害的严重程度和发生该损害的可能性。
严重程度的评估通常通过分类来完成（例如参见 ISO 14971：2007，附件 D 或 G.4，它们与验收等级相关（见 B.5）.

可能会发现，很难估计损害的可能性，特别是在考虑导致损害的软件故障的可能性时。在这种情况下，应该记住，软件故障只是导致损害的一个因素，可能涉及软件外的其他一些因素（事件序列）。 对于未知的事件可能性假设最坏的情况是有用的，最后是损害可能性的最坏情况。

IEC 62304：2006 / AMDl：2015 采取了类似的方法。 作为决定严格过程控制的基础，它假定软件出现故障的可能性最大，但考虑到与软件以外的一系列事件相关的损害的可能性较低（参见 IEC / TR 80002-1）。

B.5 风险评估

一旦估计出风险，就需要对风险进行评估，以确定风险是否可以接受。 否则，制造商应确定并实施风险控制措施，将风险降至可接受的水平。

也许风险管理中最困难的活动是确定什么是可接受的风险水平。 这种决定高度依赖于潜在危害的严重程度。 每个制造商都需要制定标准来定义和记录风险的可接受性，并确定所有风险的格式，以便评估是否符合这些标准。 一般来说，如果将可接受的风险降低到适合捍卫同事，管理层或审计师的水平，那么风险可能设定在适当的水平。
建议可接受性阈值超出了本文档的范围，但有关建立它们的过程的一些建议是适当的。

- 请明确点。接受标准，如“尽可能低”或“与其他产品一样安全”没有用处。验收标准应该像可测试规范一样阅读，以便客观确定是否符合可接受标准。
- 如果难以估计损害的可能性，则验收标准只能基于严重程度。
- 验收标准可以涉及预定义的软件过程控制选择（即选择表 A.1 至表 A.5 中列出的工具）
- 尽早确定接受标准。一旦发现潜在的伤害风险，立即设定目标或规格。在任何尝试控制风险之前设定可接受性目标非常重要。一旦企图控制风险，对可接受性的看法往往会转移到更高的风险水平。提前记录可接受性标准可以避免迁移过程。
- 记录您确定风险可接受性的基本原理。此类文件对于未来维护过程以及将思考过程传达给监管调查人员很有用。

B.6 风险控制

B.6.1 不可接受的风险

如果风险评估为不可接受，制造商应确定并实施风险控制措施，以将风险降低至可接受的水平。 这些风险控制措施可能会影响软件或过程的其他部分。

B.6.2 风险控制措施不影响软件

不影响软件的风险控制措施包括程序变更，硬件冗余，备份系统，监控系统，输出验证（下游验证）或供应商检查。

通常，嵌入式生产过程软件难以访问，并且制造商可以获得很少的细节。 一个常见的例子是嵌入在用于制造医疗设备的机床中的软件。 当软件在独立基础上进行验证时，验证此类软件的预期用途可能会很困难。

在这些情况下特别有效的风险控制措施是下游验证软件输出或由软件控制的设备输出。换句话说，通过监控软件控制过程的输出，可以直接确定软件是否适合其预期用途，以确定是否存在任何潜在的有害缺陷。 通过应用生命周期控制方法，这种方法可以替代推断软件对其预期用途的适用性。 这种方法仅适用于那些可以在每个零件上进行检查的关键操作数量相当少的流程，或者对统计上确定的零件进行抽样的流程。 验证工程师应详细说明替代下游验证的理由以及用于证明选择抽样验证而不是连续验证的任何假设，然后应对这些假设进行测试。

正如其他风险控制措施将被记录一样，下游验证应该被记录下来。尤其重要的是要证明验证过程是一项风险控制措施，以便在以后节省成本的措施中不会被淘汰。此外，下游验证结果应该形成文件，因为验证的定义要求提供验证的“客观证据”，而验证步骤取代验证的很大一部分。随着产品的发展，软件控制过程的预期用途也会发生变化。作为例子，考虑最初对医疗设备的组件执行一个关键操作的机床。后来，医疗设备设计稍作修改，使软件驱动机床需要两个关键操作。机床的预期用途发生了变化（两个安全关键操作与一个安全关键操作），因此下游验证应该改变以验证两个操作。

下游验证可以通过手动操作或其他人工操作完成。 示例可能包括视觉检查磨边或机械对准，以及手动测量机械公差或电气连续性。 无论测试的性质如何，如果它是软件控制过程的下游验证，并且如果它被用作该过程的风险控制措施，那么应该记录验证测试。 测试人员的测试程序应详细说明，并对测试的每个参数进行明确的通过和失败结果范围。 测试人员还应提供书面证据证明他们已经执行了测试过程输出的程序。

B.6.3 影响软件的风险控制措施

影响软件的风险控制措施，要么

- 设计更改或
- 过程控制。

在本文中，过程控制的选择也被称为验证的严格性，并且意味着选择表 A.1 至表 A.5 中定义的工具。

优选地，在仅依赖于过程控制之前，应当实施在软件外部充分理解的风险控制措施，例如“下游验证”以及软件设计改变。 然而，应该采用一套最低限度的过程控制措施，特别是为了提供对软件设计变更作为风险控制措施的适当实施的信心。

B.6.4 验证风险控制措施和评估剩余风险

应该验证风险控制措施的实施。 应该验证超出过程控制的风险控制措施的有效性。 在这种情况下，应该评估剩余风险的可接受性。

附录 C
（资料）
例子

本文件适用于用于自动化部分质量体系和制造过程的软件，包括用于监管提交，质量体系，生产和数据处理的数据的生成，测量，评估或管理。 其他预期用途可能包括直接或间接从仪器捕获数据，操作和控制设备以及处理，报告和存储数据。 对于这些不同的活动，软件可以从包含在可编程逻辑控制器（PLC）或个人计算机（PC）中的软件到包含在具有多种功能的实验室信息管理系统（LIMS）中的软件。 以下是预期用途的一些示例：

- 对产品做出合格/不合格决定的软件;
- 在质量体系内用于定制记录的软件;
- 数据处理和分析软件用于产品提交;
- 数据处理和分析软件，用于向监管机构报告;
- 任何用于受控过程软件的软件开发工具或编译器;
- 任何负责验证和验证生命关键软件的软件工具或从属软件工具;
- 用于质量体系内部件，产品或患者可追溯性的任何软件;
- 用于上述目的的任何“未知来源的软件”（即不知道软件的质量和稳定性是否可用）。

本附件中提供的示例代表本文档作者尝试提供医疗产品制造商可能遇到的软件的实际实际示例。 体验批判性思维方法并了解软件类型，软件风险和预期用途的可变性的最佳方式是提供这些示例。

请注意以下限定符。

这里使用的例子包括本文作者所执行的批判性思维的结果，并表示验证工作的可接受水平和严谨性，这将增加价值并提供软件将按预期运行的信心。 强烈建议读者从工程角度考虑哪些活动和工作量有意义，并根据医疗设备质量管理系统过程所用软件的关键因素确定所需的严格程度。

总是有不止一种方式来确认验证工作的适宜性。 本文档中提供的示例提供了一种基于方法的方法，该方法基于当前思考和本文档作者的经验。

强烈建议读者将的该标准参考不视为权威性或规范性。 所引用的例子在格式上仅用于数据表示，并且包括用于展示批判性思维使用的关键思维过程。 此布局不适用作验证模板，也不包含实际验证文档所需的所有深度和细节。

所用的例子假定第 6 章中确定的先决条件过程存在并且处于良好的工作状态。 虽然这些示例并未广泛提及前提条件流程，但应确保这些流程能够确保软件和所有相关方面（如文档和其他基础架构）都受控于更改。

每个例子都从明确定义要控制的过程开始。因此，已经确定该过程和因此软件在范围内。然后确定和总结批判性思维活动。

这里使用的示例旨在提供关于批判性思维过程中所用决策的决策和驱动因素的信息，并不一定代表所讨论软件的全面验证。

示例中使用的任何公司名称，团队或个人都是纯虚拟的，仅供参考以便于讨论。

这里使用的例子通常集中在使特定系统进入验证状态。尽管为系统建立验证状态非常重要，但在系统维护阶段维护验证状态对于确保软件和周边流程的正确运行也至关重要。 维护活动需要与初始验证活动所要求的相同的控制和批判性思维。

示例 1：用于制造设备的可编程逻辑控制器（PLC）

背景

Tubing 供应公司已经签约为其主要的医疗设备制造商提供静脉注射（IV）系统的导管。该公司已经收到了管材的规格，包括将管材制成专有形状的要求。 这种特殊的成型要求将在 Tubing 供应公司进行，作为其管段制造过程的一部分。

这种油管的形成过程对供应商特别重要，因为油管的形成是供应商目前没有机器执行的独特过程。 决定开发一种带有可编程逻辑控制器的定制设备来执行这项任务。 根据医疗设备公司的政策要求，该设备和其中包含的 PLC 应针对其预期用途进行验证。

定义过程

管道供应公司和设备制造商建立了一个团队来定义管道形成的过程。 会议中定义的过程使用温度和压力在塑料管中形成一个形状。 这些步骤包括以下内容：

a）获得材料;

b）插入机器;

c）通过压力和热量使管变形到适当的直径;

d）允许冷却管;

e）从机器上取下管子;

f）测量管的适当直径。

分析流程风险

医疗设备制造商已经向 Tubing 供应公司通报了风险分析过程中出现的以下问题和相关危害。

- 缺少与流体袋的良好连接会导致泄漏，这种泄漏不是危险的，但可能存在护理人员滑动的风险。 泄漏也可能延误治疗。
- 化妆品问题可能会影响客户的接受程度并导致治疗延误。
- 操作员在成型过程中被烧伤的可能性。

在缓解之前，由于护理人员滑倒，治疗延误和操作人员灼伤，导致产品失败的风险水平中等。目前有以下流程风险控制措施：

- 上游操作（如进货检验和生产线清理），以确保管材可以使用;
- 下游验证检查，包括泄漏测试，过程中检查和测试配件，以减少设备错误;
- 一个防护罩，一个独立的温度传感器和一个冷却液喷射器，以防止操作人员受伤。

利用这些信息，供应商与医疗设备制造商一起工作，得出结论，由于管子成型过程导致管子失效的风险很低。

定义软件的目的和意图

Tubing 供应公司知道，为了验证软件的预期用途，首先应该定义预期用途。 为了就设备的意图达成共识，团队成员问自己一系列问题，旨在确定系统目的和意图的简洁但可用的定义。他们产生以下声明。

该软件控制的设备旨在使所定义的过程的步骤 2 至 6 自动化。 该系统旨在用于设施 B，生产线 3，用于创建 PN 001.该系统将自动插入，成型，移除和测量 IV 用于输送一般非危险解决方案的管道

验证计划

计划验证的第一步涉及确定可交付成果的严格性和审查。 由于残余过程风险被确定为低，采取了以下方法。

- 文档严谨性：
  - 这个项目中的文档将具有中等严格性，这意味着将有可交付成果合并的实例，并且在实施之前设计不会被转换为详细的设计规范。
- 详细程度：

- 审查和批准将由负责该流程开发和实施的人员（Tubing Supply Company 代表）和独立质量人员（医疗设备公司代表）负责交付。
- PLC 代码和所有规格/设计将被置于正式配置管理之下，例如在文档控制系统或配置控制系统中。

- 定义系统：
- 将创建工艺要求，并将包括详细描述设备功能的系统要求规范，包括设备的预期输入和输出（例如整个功能设备的设计控制元素）。
- 从运营商的角度来看，该团队将创建使用该系统的操作员手册。 此外，软件需求将被创建，并将包括逻辑功能流程，这也足以涵盖软件的设计。

建立对软件的信心和控制

Tubing 供应公司和医疗设备制造商都没有使用过此 PLC 编程软件包。 Tubing 供应公司没有可用的历史记录来帮助建立对软件按需要工作能力的信心。 但是，通过审查需求，配置控制和通过测试协议测试系统的功能，可以控制 PLC 的编程。

定义与其他系统的软件边界

PLC 包含该设备中唯一的软件。 该软件没有链接到任何其他系统。

软件风险分析

该软件可能会失败，因为生产线上形状不正确的导管会导致渗漏，并可能导致护理人员滑落。 软件也可能发生故障，导致过热，从而导致操作员灼伤。 软件本身不会给产品带来任何尚未在流程风险分析中捕获的新风险。 因此，该组织确定当前的下游流程应该保持并且足以减轻与软件故障相关的风险。

完成验证计划

现在团队成员对软件及其使用有了更多的了解，他们应该按照以下方式完成验证计划。

- 实施工具：
- 设备中的一系列可编程参数包括时间，温度和压力。 设备中这些参数的所需设置和范围均在软件要求中捕获。 因此，软件需求规格对于设计而言是足够的，无需额外的设计活动或文档。
- 该团队将建立软件需求与其相关测试之间的可追溯性矩阵，并进行可追溯性分析以确保可追溯性完整。

- 测试工具：
- 软件系统测试将基于操作员手册中的软件要求和程序。
- 如果需要，将执行回归测试。

- 部署工具：
- 系统操作员和工程师将为了清晰和可用性而检查工作说明。
- 设备的使用将需要操作员证明。
- 在完成验证计划并执行其活动之后，团队感到满意的是该系统将始终如一地提供期望的和确定的输出。

维护考虑

如果对此过程的任何部分进行更改，或者如果软件的预期用途发生变化，则应执行分析以确定任何当前的缓解措施将受到影响，或者是否有任何新风险与此更改相关联。 该分析包括审查与成型设备相关的软件风险。

工具箱的使用

工具箱中使用了以下工具。

- 开发定义阶段：
- 过程需求定义;

- 过程故障风险分析;
- 有可能的使用;
- 验证计划;
- 软件需求定义;
- 在制造过程中识别风险控制措施。

- 开发 - 实施阶段：
  - 分析软件故障;
  - 可追溯性分析。
- 开发测试阶段：
  - 软件系统测试;
  - 回归测试。
- 开发 - 部署阶段：
  - 用户程序审查;
  - 运营商认证。

示例 2：自动焊接系统

戴夫是验证新生产线上所有系统的团队的一员。 他的工作是验证案例封面焊工。 对于这个项目的努力，他是项目经理。

过程描述

戴夫的团队花了很多时间讨论谁在开发和验证新生产线的哪些部分。当 Dave 拿到零件时，它们已经被标记，所有材料都经过检查和认证。 部件在上游经过验证的系统上进行测试。

要建立焊工，需要四个步骤：

- 打开机器;
- 确认要运行的零件中是否存在条形码;
- 从制造执行系统中拉取零件的程序;
- 根据设备主记录确认正确的程序版本。

箱盖焊接过程本身有 10 个步骤：

a）打开门;
b）加载零件;
c）关上门;
d）启动程序;
e）将视觉系统索引放置在起点;
f）打开激光器;
g）确保运动控制移动零件焊缝;
h）关闭激光器;
i）打开门;
j）移除零件。

这个过程完成后，这些零件将转移到不是戴夫责任的系统。 他知道下游活动包括焊接渗透的破坏性测试，罐体尺寸的高度检查以及气密密封的泄漏检查。

定义预期用途

为了定义软件的预期用途，Dave 收集信息。 他知道视觉，运动，功率和速度的准确性对于保护操作人员的安全和实现一致的焊接穿透过程都是非常重要的。

戴夫首先通过如下说明软件的目的和意图来首先定义他的预期用途。

该软件旨在焊接箱盖，保护机器操作员免受直接访问操作激光器的影响。 这包括上述步骤 e）至 h）中的步骤。

风险分析

戴夫想从流程中删除人为错误。他知道控制激光，伺服机构和视觉是这个过程的关键组成部分。软件首先检查门是否关闭。出于安全原因，如果软件无法感测到门已关闭，则该软件不会启动该过程。软件结束时确认激光关闭，然后打开门。急停或意外打开门会切断激光器的电源。 戴夫使用过程中的信息和设计风险管理活动，这些活动是焊接过程设计的一部分。他提到 FMEA，重点关注三个领域：关键部件参数，密封和用户界面。戴夫发现了与此过程有关的多种危害。首先，如果暴露在激光下，操作人员可能会被烧伤。与产品相关的过程可能会焊接不当，从而导致可能泄漏并伤害最终用户的不良产品。戴夫确定这个过程的风险很高。

验证计划

对于这个项目，Dave 查看工具箱中的定义工具，并确定他需要创建一个软件需求定义和维护文档。 他的软件包括工具和激光时间和功率调整的配置参数。 他还需要将软件定义为硬件接口。 具体而言，Dave 包括视觉系统的准确度要求，激光时间和功率范围，运动控

制精度要求和门传感器安全措施，包括激光激活时硬件门锁的接口。

戴夫还确定他需要进行正式的软件需求审查，其中包括自动化工程师，制造工程师和质量工程师。

这个系统的软件将是一个购买的软件包，但戴夫知道他的公司需要进行自定义修改。他需要为工厂制造执行系统（MES）添加一个接口。

风险控制措施

戴夫接下来关注风险。 他认为焊缝深度和其他关键参数的严重程度很低，因为他确信下游检漏和周期性破坏性检测是足够的。 同样，泄漏检查将确认密封是可以接受的。 这在用户界面方面留下了风险，特别是在门打开时软件可能启动激光器的风险。 Dave 知道有门封的软件检查，但是如果软件无法按预期运行，风险的严重程度会很高，他会添加一个冗余的硬件互锁，以防止门打开时的激光激活。

验证任务

接下来 Dave 转向验证任务。他选择的工具供应商提供了广泛的编程工具。因此，先前创建的软件需求规范和评论对于设计而言是足够的，而无需使用工具箱中的其他设计，开发和配置工具。

Dave 从工具箱的测试部分选择的另一项任务是测试计划。测试计划将包括软件环境的细节和预期的测试结果。测试计划需要由自动化工程师，制造工程师和质量工程师以及 Dave 进行审查和批准。测试报告将包括实际测试结果并将其与预期结果进行比较，提供合格/不合格指示，包括测试识别，并提供问题解决文档和任何故障的回归测试。对于本报告，Dave 希望获得自动化工程师，制造工程师，质量工程师和项目发起人的额外批准。

部署

为了部署焊工，Dave 检查工具箱中的部署工具，并决定需要制造操作员程序，并且需要由自动化工程师，制造工程师和质量工程师进行审查。 为了确保操作员理解如何操作焊工，戴夫创建了包括测试的操作员培训和认证程序。 他知道 MES 不允许操作员在没有认证的情况下将焊接程序从系统中拉出来，因此他对操作员受伤的风险已被成功减轻感到满意。

保养

戴夫知道他的公司有一个配置检查工具。 因此，在此验证过程中没有执行特定的维护计划。

示例 3：自动焊接过程控制系统

本例中的表 C.1 至表 C.14 说明了图 2 中所示的过程步骤。

表 C.1 - 实施例 3 - 工艺要求

<table>
<tr><td rowspan="2">开发</td><td rowspan="2">确定</td><td>处理</td><td>更换风险分析</td><td>验证计划和报告</td><td>软件系统</td></tr>
<tr><td colspan="4">工艺要求见 5.3.2.2
设备公司是 C 类（参见 GHTF / SG1 / N77：2012）医疗设备制造商。 设备公司选择实施自动焊接过程控制系统。为确保设备外壳得到适当的焊接，设备公司将采用使用参数发布决策流程隔离产品的方法。设备公司也选择使用此过程中的信息来支持其设备历史记录。
设备公司已经指派了一名新的项目经理来验证自动焊接过程控制系统。项目经理认识到该系统需要符合 ISO 13485 的软件验证要求。因此，项目经理认识到所提出的焊接过程控制系统需要验证
为了更好地理解验证焊接系统所涉及的要求和风险，项目经理将过程定义如下。
a）操作员将批号输入批次的第一部分的系统中。
b）操作员将子部件插入机器夹具中
c）操作员按下循环开始按钮。夹具通过液压机械地移动到配合位置。
d）焊接循环与固定子组件的固定速度旋转一起开始。
e）红外温度计监测焊接过程中的材料温度温度记录在一个文件中，以及焊接每个零件的批号和零件序号。
f）机器在循环结束时打开夹具
g）操作员根据序列号移除焊接零件并将零件放置在批次托盘中的相应位置。
h）操作员重复步骤 b）到 g），直到批盘充满。
i）操作员点击结束按钮。
j）机床操作员界面显示焊接温度超出过程极限的零件序号。
k）操作员从批盘中丢弃相应的零件号码。
i）操作员打印被拒收的零件清单，并将批次托盘和报告发送到下一个工作站。
m）操作员通过重复步骤 a）开始新批次。
项目经理还意识到，关键的自动化功能如下：
• 存储批号;
• 存储每个顺序零件号的焊接温度;
• 显示焊接过程中超过过程温度极限的部件序号;
• 打印拒绝报批次告。</td></tr>
</table>

表 C.2 - 实例 3 - 过程失效风险分析

<table>
<tr><td rowspan="2">开发</td><td rowspan="2">确定</td><td>处理</td><td>更换风险分析</td><td>验证计划和报告</td><td>软件系统</td></tr>
<tr><td colspan="4">过程失败风险分析（见 5.3.2.3）
然后项目经理会考虑当前流程中会出现什么问题。 项目经理意识到，如果过程出现故障，释放不正确的焊接部件会使患者暴露于非无菌设备。 由于焊接过程控制系统错误或操作员错误，可能会发生不良产品的意外泄漏。
项目经理然后考虑采取什么风险控制措施来降低风险。 项目经理得知过程组有一个过程来验证焊接操作员在下一个工艺步骤中正确地拒绝了零件。 此外，项目经理得知焊接系统是一个商业 OTS 系统</td></tr>
</table>

表 C.3 - 例 3 - 软件的目的和意图

开发	确定	处理	更换风险分析	验证计划和报告	软件系统
		软件目的和意图（见 5.3.2.5.2） 通过对流程的基本了解，项目经理随时准备为焊接过程控制系统编写目的和意图。 - 焊接过程控制应用程序对焊接外壳的通过或失败状态做出闭环质量决定。根据这些决定，焊接操作员手动拒绝不合格产品。 项目经理回顾了在过程中恰当地捕捉软件边界的目的和意图，并决定如下修改声明。 - 焊接过程控制应用程序对焊接外壳的合格或不合格状态做出闭环质量保证决定。在这些决定的基础上，焊接操作员然后手动拒绝参数不合格的情况。焊接站是整个设备过程中确保设备密封完整性的唯一控制点。 项目经理然后考虑哪些其他系统（如果有的话）需要与焊接系统连接。他确定软件是连接到红外温度设备，操作员界面，打印机和机器 PLC 输入/输出的 PC 上运行的单个应用程序。焊接系统没有连接到网络		

表 C.4 - 例 3 - 验证计划

开发	确定	处理	更换风险分析	验证计划和报告	软件系统
		验证计划（见 5.3.2.4） 既然项目经理了解过程并确定了新系统的预期用途，那么项目经理已准备好在高层次上制定验证计划。 早些时候，项目经理确定焊接过程中存在很高的剩余风险，因为它将作为不可验证的过程来实施。因此，项目经理确定需要对验证工作进行广泛的审查。项目经理决定，关键审批角色应由过程工程和质量工程部门以及运营过程培训师完成。 此外，最终产品验收经理应批准这些要求。 项目经理决定开始编写验证计划，因为质量体系要求验证计划在批准任何其他验证可交付成果或项目可交付成果之前已被批准用于高风险系统			

表 C.5 - 示例 3 - 软件使用要求和软件要求

开发	确定	处理	更换风险分析	验证计划和报告	软件系统
		软件使用要求和软件要求（见 5.3.2.5） 项目经理认为，有必要在验证工作中提供高级别的细节或方面，并确定详细的流程和软件需求。 项目经理现在编写软件需求。 项目经理决定在软件应包括温度验证和排出决策过程中的冗余。 项目经理也要求系统能够在线路清理活动发生之前的任何时候重新打印拒绝报告。 如果系统支持参数值，项目经理还会包含安全权限，同时列出哪些数据值可以通过系统访问级别进行更改。			

表 C.6 - 实例 3 -软件故障风险分析

开发	实施测试和部署	处理	更换风险分析	验证计划和报告	软件系统
		软件故障风险分析（见 5.3.3.2） 项目经理现在需要决定应该采用什么方法来建立对焊接系统的充分信心。 项目经理指出，焊工设计要求工业中常用的商用现货（COTS）系统。 项目经理发现，制造商已经快速识别并公布了此产品过去的问题或问题。 虽然项目经理已经确定焊接工艺具有高风险，但该项目 经理仍然想要正式分析软件失败的风险。 为了证实这个直觉，该项目 经理回顾公司风险模型的问题。			

		a）如果软件发生故障，产品安全是否存在潜在风险？ 是 1）为何？ 系统根据默认温度限制接受不良部分。 限制重置为默认值电源故障后设置 2）应该做些什么来控制这种风险？ 要求操作员在每次批次运行的开始和结束时验证限制。 b）如果用户犯了错误，产品质量是否存在潜在风险（安全风险除外）？ 是 1）如何？ 在手动模式下，如果两个部件传感器均触发 3 秒，则焊接激光器会发射。 2)应该做些什么来控制这种风险？ 将默认配置更改为仅在自动模式下触发。

表 C.7 - 例 3 - 验证计划

开发	实施测试和部署	处理	更换风险分析	验证计划和报告	软件系统
		验证计划（见 5.3.3.2） 通过了解软件需求，项目经理有足够的信息来完成验证。 项目经理已经决定实施方法并分析了软件风险。 在这一点上，项目经理回过头来问这个问题，根据所有关于这个系统的知识：“什么样的验证活动能真正让我相信焊接系统适合它的预期用途。” 项目经理考虑系统是如何由第三方开发的，并且担心开发人员会正确地翻译报告定制的要求。 由于系统将取决于各种数据字段，因此项目经理会在代码审查中添加验证步骤活动，以确认开发人员工作的正确性。		

表 C.8 - 实例 3 - 软件实施

开发	实施测试和部署	处理	更换风险分析	验证计划和报告	软件系统
		软件实施（设计，开发，构建和测试）（见 5.3.3.4） 购买宁可在内部开发软件的决定是基于商用现成（COTS）系统的可用性而制定的。 但是，项目经理仍需要证明 设备公司的质量部门认为焊接控制软件是在有效的软件开发生命周期（SDLC）下开发的，因为预期的使用风险被分类为高 在与 COTS 供应商讨论此问题后，项目经理获悉供应商的 SDLC 流程最近由独立审计公司审计。 然后项目经理联系独立审计公司并购买 COTS 供应商 SDLC 审计报告的副本。最终结果是质量部门相信 COTS 供应商在有效的生命周期模型下开发软件。。		

表 C.9 - 例 3 - 验证报告

开发	实施测试和部署	处理	更换风险分析	验证计划和报告	软件系统
		验证报告（见 5.3.3.5） 项目经理完成并获得验证报告的批准。			

表 C.10 - 示例 3 - 软件版本

开发	实施测试和部署	处理	更换风险分析	验证计划和报告	软件系统
		软件版本（见 5.3.3.6） 项目经理验证放置在正式配置管理系统下的软件与验证报告中引用的软件相匹配。			

表 C.11 - 实例 3 – 变更分析

保持	处理	更换风险分析	验证计划和报告	软件系统
	变更分析（见 5.4） 项目经理验证，在验证计划下，公司有一个正式的变更控制流程，管理焊接系统的任何验证后变更。			

表 C.12 - 例 3 - 维护验证计划.

保持	处理	更换风险分析	验证计划和报告	软件系统
	维护验证计划（见 5.4.2） 项目经理会提前考虑哪些活动适合确保系统继续实现其预期用途的信心。 考虑到系统的高风险，项目经理决定应该进行季度校准和认证，以确保批量报告中印刷的实际温度测量值与温度值的准确性和精确性。 项目经理在验证计划中包含一个部分，以记录该结论，并提出开发和实施校准和认证程序的请求，以确保系统投入生产后进行季度评估。			

表 C.13 - 例 3 - 软件维护

保持	处理	更换风险分析	验证计划和报告	软件系统
	软件维护（见 5.4.6） 项目经理验证，在验证计划下，公司定期审查过程，确保焊接系统和工艺不会因其预期用途而变化。			

表 C.14 - 示例 3 - 软件退役

退役	处理	更换风险分析	验证计划和报告	软件系统
	软件退役（见 5.5） 项目经理证实，在验证计划下，公司有正式的软件退役流程，管理焊接系统的退役			

工具箱选择

设计，开发和配置工具

- 流程需求定义
- 正式的软件需求审查
- 在制造和业务流程中识别风险控制措施
- 流程开发审查
- 可追溯性矩阵（需求规格中固有的）

- 测试工具
  - 测试计划
  - 软件系统测试
  - 软件配置控制
- 部署工具
  - 用户程序审查
  - 对应用程序进行内部培训
  - 安装资格
  - 流程验证

例 4：C / C ++语言编译器

背景

C 类医疗设备公司需要验证其嵌入式系统的现成软件 C / C ++语言编译器。 已经确定编译器是受监管的，因为它生成放置在医疗设备设计记录中的医疗设备产品软件（软件源代码和可执行软件）。

质量体系流程描述

这个案例研究涉及两个质量体系流程。 首先是实施 C 类医疗设备软件的整体质量体系流程（见图 C，1）。 第二个是开发实施软件设计并满足所有软件需求的可执行软件单元的过程。 这些软件单元包括 OTSS C / C ++语言编译器（参见图 C.1 中的“软件实现”）。

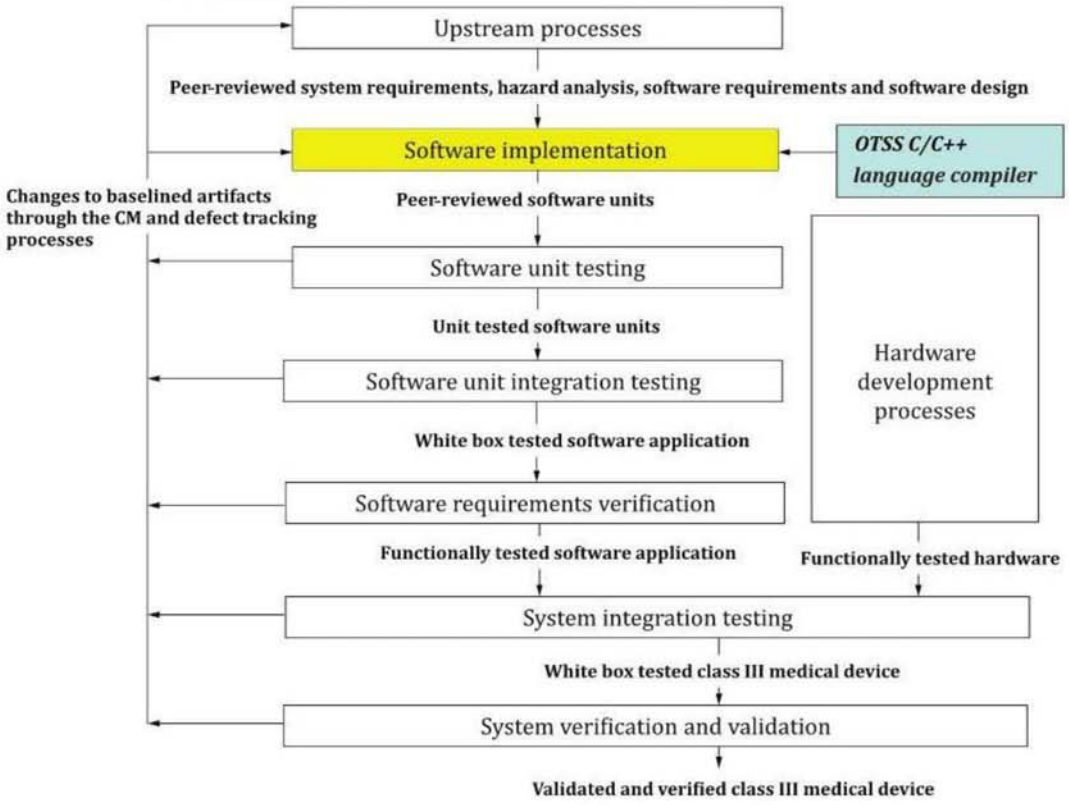


图 C.1 - 实施 C 类医疗设备软件

上游流程

实施软件过程的上游是开发系统级文档（例如需求，设计，危害分析）的过程，这些文档是要开发的医疗设备的特征。 然后在软件中执行的系统部分通过流程进行表征，以开发软件需求，软件设计和其他软件文档或计划。 在软件开发的同时，执行额外的流程来开发医疗设备硬件。

软件实施过程

使用的正式软件语言是 C / C ++软件语言。 OTSS C / C ++语言编译器 1 用于将高级软件语句编译为可执行的机器代码。 软件实施过程的输出是基线软件单元，由其他技术成员进行同行评审以获得完整性和正确性。 对于软件单元同行评审，软件单元应在最高编译器级别进行无差错编译，并且应在同级评审中解释任何编译器警告。

降低测试工作量

软件单元在以下几个测试过程中进行测试或验证。

- 软件单元测试。测试各个软件单元的逻辑正确性和每个单元的边界条件。此测试可能发生在开发系统或目标系统（医疗设备硬件）上。当确定代码对等评审足以检测单元逻辑

错误时，简单的软件单元可以放弃此测试。

- 软件单元集成测试。软件单元经过集成和测试，以确保软件设计得到正确实施，并且测试了与设计相关的边界条件。该测试发生在目标系统上。
- 软件需求验证。完整的软件应用程序是根据完整的软件需求进行验证的。此验证在目标系统上执行。
- 系统集成测试。对医疗设备中的软件和硬件进行测试，以确保正确实施系统设计，并测试与系统设计有关的边界条件。
- 系统验证和验证。医疗设备在系统要求级别进行验证，另外还对其预期用途进行了验证。

分析过程失败风险

该项目遵循公司的流程风险评估程序。 实施 C 类医疗设备软件（包括图 C1 中描述的所有过程）的整个质量体系过程本质上是高风险的，因为它会生成在 C 类医疗设备内运行的软件。

作为软件实施过程的一部分，OTSS C / C ++语言编译器在两个因素的基础上被评估为低风险：

- 编译器不会直接对患者，操作员或旁观者造成严重伤害或死亡;
- 对工具的输出（软件源代码和可执行软件）（例如软件单元测试，软件单元集成测试，软件需求验证，系统集成测试，系统验证和验证）执行下游验证。

预期用途定义

OTSS C / C ++语言编译器在上述软件实现过程中的目的和意图是编写嵌入式系统源代码并执行编译过程以生成 C 类医疗设备的可执行软件。

软件使用要求

a）该工具应该交叉编译 C 和 C ++代码以在精简指令集计算机上工作
（RISC）处理器使用选定的供应商操作系统。

b）编译器应该有一个源代码调试器。

c）编译器应该是美国国家标准学会（ANSI）C 和 C ++标准。

d）编译器应该与各种已批准的行业标准集成开发环境集成。

e）供应商应该发布可搜索的已知错误列表。 该清单应作为参考，以便根据需要进行咨询。

f）供应商需要在受监管行业内拥有庞大的用户群。

分析软件故障风险

OTSS C / C ++语言编译器的风险分析表明，如果发生错误，则可能发生以下事件。

- 风险 1.供应商未能提供适当的业务流程，开发方法和支持能力。
  - 缓解措施 1.请参阅下面的“供应商选择流程”部分。
- 风险 2.编译器生成不正确的可执行语句。
  - 缓解措施 2.请参阅“验证计划”部分。
- 风险 3.未进行最严格级别错误检查的用户错误地使用编译器。
  - 缓解 3.改善培训，程序和工作指示。

供应商选择过程

该项目遵循公司选择和批准供应商的质量体系程序，并将该信息记录在项目的设计记录中。该程序包括现场评估供应商的 SDLC 策略，程序，任务和活动。 验证供应商提供的 OTSS C / C ++语言编译器的功能，以满足上面定义的软件使用要求。

验证计划

OTSS C / C ++语言编译器选择了下游验证方法。 供应商选择流程确定供应商符合所有记录的软件使用要求。 编译器在供应商处有很长的运行时间，并且在项目的调试和测试过程中会有很长的运行时间。 编译器的输出受下游进程中的以下动态测试的影响：

- 软件单元测试;
- 软件单元集成测试;
- 软件需求验证测试;
- 系统集成测试;
- 系统验证和验证。

验证报告

验证报告的内容如下：

- OTSS 说明
- 软件使用要求
  - 硬件要求
  - 软件需求
  - 补丁
- 风险评估和危害分析
- 供应商选择
- 安装活动
- 验证
  - 软件使用要求测试用例和结果
- 已知的错误列表
- 配置控制
  - 训练
  - 安装位置
  - 保养
  - 退休流程

工具箱选择

- 定义阶段：
  - 有可能的使用;
  - 验证计划;
  - 风险管理计划（风险评估）。
- 实施阶段：
  - 风险控制措施;
  - 供应商审核。
- 部署阶段：
  - 安装资格;
  - 应用程序的内部培训;
  - 最终验收测试。
- 保持阶段：
  - 维修计划;
  - 已知问题分析。

例 5：自动软件测试系统

背景

在这个例子中，制造商是 C 级医疗设备制造商。该制造商生产的医疗设备由软件控制。该软件在体系结构上分为两个主要组件：操作员控制台和实时嵌入式控制软件。 操作员控制台是系统的主要人机界面。 实时嵌入式控制软件是执行机电控制，数据采集，定时等的软件。 操作员控制台软件（驻留在运行行业标准操作系统和数据库的 PC 中）和实时嵌入式软件（驻留在板载嵌入式 CPU 卡中）通过标准传输控制协议/ Internet 协议（TCP / IP）硬件和协议接口。.

该项目的软件管理者已经决定，通过引入软件的自动化测试来改进软件开发和测试过程将是有价值的。 软件管理员决定首先实施只有操作员控制台软件的自动化软件测试。 自动化软件测试将在集成测试点和软件系统测试点进行。

确定软件是受管制的

由于自动化测试软件将用于执行制造商软件开发程序所要求的测试，并且由于它将在集成和系统测试点提供所需的回归测试证据，所以自动化测试软件被确定为自动执行部分测试开发过程，因此被确定为符合 ISO 13485 的验证要求。

定义过程

为了更好地理解引入操作员控制台的自动化软件测试所涉及的要求和风险，软件管理员在软件开发过程中定义了如下对自动化测试软件的使用。

在设备软件的开发过程中，各种模块计划在不同时间集成到系统软件中。另外，由于缺陷修正和需求修改，已经集成到系统中的模块将会发生变化。计划自动化测试系统用于集成系统软件的回归测试以及系统中特定模块的最终测试。软件项目计划要求模块的整合或更新每周发生两到三次。自动化测试将在每个集成点上运行，以确保新功能正常工作，并且以前的工作功能没有受到已添加的新代码或特定构建中代码更改的不利影响。自动化测试将在软件系统测试级别上运行，以供最终发布验证并最终发布给客户。如果在开发的最后阶段发现缺陷，需要对其进行修正以提供补充计划手动测试的回归测试级别，那么也将使用自动化测试。

分析风险

如果使用自动化测试软件的方法不正确，软件管理员现在将通过分析过程来确定任何潜在的影响。

软件管理人员需要评估的第一件事是，自动化测试过程的失败，自动化测试软件的失败或任何使用自动化测试软件的人犯的错误最终是否会导致医疗设备中的缺陷 潜在地伤害患者，操作者，旁观者，服务人员或环境。

- 软件管理人员最关心的问题是，自动化软件测试系统会给出一个错误的指示，即当测试中的操作员控制台软件实际上仍然存在缺陷时，它可以正常工作。.
- 如果未检测到的缺陷位于软件的关键区域，则可能会导致医疗设备故障，从而造成危害情况
- 软件管理员意识到，这种风险可能来自错误的管理，使用自动化测试软件或自动化测试软件本身的缺陷。
- 软件管理人员决定在自动化软件测试系统可以使用的时候提供边界条件，以及可以用来确保软件开发和测试团队不过度依赖系统，这一点非常重要。
- 需要参与配置，编程和操作自动测试软件的个人需要接受他们的角色培训。
- 软件经理认为，如果这些因素得到控制，潜在的相关风险将被降低到可接受的水平。

定义软件的预期用途

在分析了自动化测试软件的潜在用途和相关风险后，软件管理人员随时准备为自动化软

件测试系统制定目的和意图声明。 声明如下。

- 自动化测试系统将用于在开发过程中在集成测试点测试软件的构建。
- 自动化测试系统将用于在软件系统测试点测试验证和候选版本构建。
- 自动化测试系统将执行系统的回归测试，以确保工作流程没有受到新引入软件或更改软件的不利影响。
- 自动化测试系统的一般作用是提供补充回归测试以进行手动测试。
- 对于低复杂度，可预测的工作流程，考虑到特定协议已经验证与相应的手动测试一致，自动化测试系统可以用作软件正确性的最终决定因素。
- 自动化测试系统将运行软件，为软件系统或整个医疗设备提供安全保护（风险缓解）。

验证计划

软件管理员现在对自动化过程，自动化测试系统的特定用途以及所涉及的潜在风险有清晰的了解。 软件管理人员已经确定需要对软件的使用进行某些控制，并且自动化软件测试系统（如果按照软件管理员规定的适当控制方式使用）将具有可接受的等级 与其使用相关的风险。

- 在这种情况下，软件管理人员已经确定，当自动化软件测试系统得到适当使用时，很少或根本没有风险存在，它会导致医疗设备缺陷。 软件管理者已经适当地定义为软件开发和测试团队不会过度依赖使用系统来确定软件的正确性。 考虑到低风险的决心，软件管理人员已经确定系统的验证要求将在软件测试系统测试的努力和严谨性的低端。

验证文档：验证报告方法

软件管理员选择的方法是为自动化软件测试系统开发软件验证报告，其中包括所有与获得系统信心水平有关的活动的总结。

批判性思维

现在，软件管理员决定如何最好地达到必要的信心水平，该系统将被恰当地使用，并且不会对医疗设备中的严重缺陷造成影响。

他认为，在达到系统信心水平的最重要因素中有以下几点。

- 严格遵守适当的预期用途
  - 确保参与软件开发和测试的所有人员清楚地了解系统的边界条件和适当的预期用途。
  - 文档：在验证报告中包含一个部分，描述具体的预期用途以及通过项目软件开发计划传达信息的方式。
- 尽职调查
  - 从信誉良好的供应商处购买行业标准的自动化软件测试系统，其测试系统用于相同级别的关键性或更关键的应用。
  - 与供应商一起审查系统的预期用途，以确定预期用途是否合适。
  - 获取有关供应商在向商业市场发布之前如何验证软件的信息。从供应商的质量部门获得一份声明，确认商业软件已经过供应商验证。该声明将使人相信自动化软件测试系统已经得到了供应商的充分测试，并为软件管理人员和软件开发和测试团队将执行的其他活动奠定了初步基础。
  - 建立与供应商的关系，以确保软件经理和软件开发和测试团队了解他们将使用的测试软件版本的已知问题和缺陷。
  - 了解供应商未来的软件更新计划，以确保可以预期到新软件和重新验证活动的迁移计划。
  - 文档：在描述供应商尽职调查结果的验证报告中包含一个部分，包括供应商验证自动化软件测试系统的信息，访问供应商缺陷（缺陷）列表的方法以及预期迁移计划到新版本的软件。

- 安装测试
  - 确认软件所在的计算环境是否符合供应商的规格。
  - 建立初始的高级测试协议，以确保软件安装正确。
  - 文档：在描述安装确认活动结果的验证报告中包含一个部分。
- 风险管理
  - 确保只有软件经理在软件用途和意图中定义的系统才能使用。
  - 在使用自动化测试系统的项目的软件开发计划中包含特定的允许边界条件。
  - 进行分析以确定系统测试的确切覆盖范围，以确保手动测试解决自动化软件测试系统未涵盖的领域。
  - 文件：在验证报告中包含一个描述初始风险分析中识别的风险的部分，并指出如何减轻每个风险
- 软件使用要求
  - 开发他们打算使用的自动化测试系统功能的列表。该软件管理器和软件开发和测试团队开发的这个列表被称为“软件使用要求”，代表了将要使用的功能。
  - 文档：在验证报告的一部分中包含“软件使用要求”列表，并描述每个软件使用要求。
- 自动化测试系统的验证
  - 使用“软件使用要求”列表来确定必要的置信度。通过采用三种初始自动化测试脚本或协议并针对手动运行的相同协议运行并行测试，可以确定置信度。这三个最初的测试脚本或协议将执行团队将使用的所有功能。
  - 文件：在验证报告中加入一个部分，概述侧面结果并排测试，并包括测试证据，以表明结果是相同的。
- 训练
  - 为所有系统用户制定培训计划，确保他们完全了解如何使用该系统并且有资格使用该系统。软件经理认为，培训是确保自动化软件测试系统安全有效使用所需的最重要因素之一。
  - 文档：在验证报告中加入描述系统用户所需的必要培训的章节。
- 个别自动化测试协议的验证
  - 如果自动化测试系统将用于测试旨在减轻系统，硬件或软件风险和危害的软件，请确保每个协议均已通过使用自动化测试和手动测试的并行测试进行验证。
  - 如果自动化测试系统将用于低复杂度，可预测的工作流程的最终测试，请确保每个协议均已通过自动化测试和手动测试的并行测试进行验证。
  - 文档：确保医疗设备的软件验证记录包括符合该类别的测试脚本或协议的并行测试证据。
- 配置管理
  - 确保只安装并使用自动化测试软件的适当，经过验证的版本。
  - 随着自动化测试软件的新版本可以从供应商处获得，控制这些新版本或更改的实施，以确保在适当的时候引入版本或更改。
  - 确保在每个更新点都考虑自动化测试系统的重新验证，并且系统的每个重新验证都要进行并形成文档。
  - 文档：在描述系统配置管理计划的验证报告中包含一个部分。

验证报告

作为建立信任活动的结果，软件管理者提交验证报告以供最终审查和批准。 该报告传达了确定要进行的增值活动的思维过程，以便软件管理人员可以得出结论，即使用自动化软件测试系统会导致相关医疗设备在开发过程中无意中出现瑕疵。 报告还包含证据表明所有

确定为重要的活动都按计划进行。

以下是验证报告的内容：

- 过程定义;
- 风险分析;
- 风险管理;
- 有可能的使用;
- 供应商尽职调查;
- 训练;
- 安装测试;
- 自动化测试系统的预期使用验证;
- 维护，重新验证和配置管理。

验证报告审查和批准

软件经理将验证报告发送给项目经理，项目软件质量保证经理和软件测试经理进行审核和批准。

所有评论者都认为，软件经理已经清楚地了解了系统的预期用途，并了解了系统使用中涉及的所有相关风险。 评审人员认为，为了达到系统使用所需的系统信心水平，必须进行所有必要的活动。 审查人员批准该计划。 该系统被视为经过验证并投入使用。

例 6：一个简单的电子表格

背景

ZYX 公司的实验室分析师已经厌倦了从文档控制系统中为他们分析的每一种产品提取不同的规格表，然后手动计算他们需要与规范进行比较的角度数。实验室中的仪器用于接收检查。该仪器测量三个坐标位置，分析人员使用该位置计算与规范相比较的角度。该实验室最近遇到了三个分析师错误地计算角度的例子（因为分析师说“胖手指”），分析师想要防止这个错误再次发生。他们决定创建一个电子表格来执行角度计算，并将他们分析的所有 50 种产品的规格结合到该电子表格中。他们将输入他们仪器测量的三个坐标对，从下拉菜单中选择产品名称并获得通过/失败结果。分析人员还认为该仪器的界面直接将坐标传递给电子表格，但由于界面的成本，这种增强功能将推迟到明年。

过程的定义

当前进程包含以下步骤。

a）让仪器测量零件。

b）记下三个坐标对。

c）计算角度。

d）从文件控制系统中拉出部件的规格。

e）将角度值与规格进行比较并确定合格或不合格。

f）在零件上放一张表格或一张失败表格，并将它们发送到产品零件清单中。

新流程将包含以下步骤。

a）从文档控制系统中获取电子表格。

b）让仪器测量零件。

c）在电子表格中输入三个坐标对。

d）目测检查输入的坐标对与仪器值的对应关系。

e）在电子表格中选择部件号。

f）在电子表格中选择“计算结果”。

g）目测检查是否选择了正确的零件号。

h）根据不同的结果，在零件上放一张表格或一张失败表格，并将它们发送到产品零件库存中。

预期用途的定义

分析师定义电子表格的目的和意图如下：电子表格将输入三个输入的坐标对，计算角度，然后将该角度与所选产品的产品规格进行比较，报告合格/不合格结果。

风险分析

分析师集体讨论与电子表格相关的可能危害。 他们认为不正确的结果可能意味着不符合规格的零件可用于生产。 对于这些有缺陷的部件，将其交给医疗设备的最终用户，至少另外两个下游分析师对与电子表格相关的可能危害进行头脑风暴。 他们认为不正确的结果可能意味着不符合规格的零件可用于生产。 对于这样的有缺陷的部件，使其到达医疗设备的最终用户，至少另外两个下游

验证计划

由于生产不合格产品的风险较低，因此验证工作的努力程度很低。 分析师决定将电子表格要求和验证计划合并到同一文件中。分析师还决定将设计文档与高级测试计划相结合。对于这些文件，分析师计划由整个分析师团队（4 人）以及质量保证代表进行评审。 此外，分析师计划咨询技术专家来开发一组具有代表性的测试数据，以建立对计算按预期运行的信心。 技术专家也会批准文件。

风险控制措施

分析师会查看电子表格中的每个项目，这可能会导致错误并导致错误的结果。 对于每个项目，分析师都确定他们将如何减轻风险（见表 C.15）。

表 C.15-示例 6 - 风险和缓解措施

风险	缓解措施
可能输入不正确的值。	通过程序控制确认针对仪器输入的每个值对。 步骤 d）被添加到新过程中来执行此操作。
计算可能不正确。	确认公式是正确的，并提供准确的结果。
可能选择错误的产品。	通过程序控制确认零件编号。 步骤 g）被添加到新过程中来执行此操作。
指示结果的宏可能不正确	确认宏是否正确并且按照预期执行。
电子表格中的规格可能不正确	根据 50 个产品规格表确认电子表格规格。 如果规格更改，则增加规格表更改的过程以要求更新电子表格。 （这从来没有发生过但可能。）
验证后可以更改计算公式或宏。	带有配置控件的验证电子表格将被放入文档控制系统并在每次需要时进行检索。 配置控件将包含所有非数据输入单元的密码保护和锁定单元。

验证任务

使用的公式可以理解，开发人员在电子表格宏观开发方面经验丰富。验证将确认以下项目：

- 计算;
- 宏;
- 锁定功能（锁定的单元格不能更改）;
- 数据输入检查（允许范围内的值，适当的产品选择，信息错误信息）。

由于电子表格一次只能生成一个结果，因此不需要进行压力测试或性能测试。 将为所有测试创建一个测试计划和报告。 该报告还将发布电子表格以供使用，并将确认对公司文档控制系统中此电子表格的控制。

部署

在部署新系统之前，测试已经完成，制造操作员已经通过了新视觉系统的操作认证。

工具箱中的工具

- 需求定义（记录在验证计划中）
- 过程失败和风险分析（记录在验证计划中）
- 预期用途（记录在验证计划中）
- 验证计划
- 测试计划
- 操作员认证
- 维护计划（需要回归分析）

保养

每当产品规格发生变化或添加新产品时，都需要在电子表格中进行维护。 维护测试计划将与完整验证测试用例的代表性子集一起开发，以确保新项目不会中断电子表格。 维护计划将要求进行回归分析，以确定是否需要将其他测试用例添加到特定于所做更改的测试用例子集中。 该计划还将介绍如何更新电子表格（例如解锁单元格，更改，重新锁定）。

例 7：一个（不是很）简单的电子表格

软件说明

一个软件开发团队已经使用 Microsoft Excel 电子表格作为开发辅助工具。电子表格将记录 C 类或 D 类设备中使用的设备消息转换。 该设备的原始版本是用美国英语编写的。 后续版本将支持七种语言。 电子表格由七列组成。 最左边的列是英语设备消息
用于设备中的每条消息。 剩下的每一列表示要支持的国际语言之一，而列中的每一行表示该行最左列中的英语翻译为国际语言的特定英语语言消息。

预期用途

电子表格满足短暂的需求

- 视觉上组织英语消息及其翻译，
- 创建一个可发送给当地代表的电子表格，以便将翻译后的信息直接收集到电子表格中，或以手写的形式收集在电子表格的硬拷贝上，
- 为翻译后的消息提供一个瞬态数据存储工具。

一旦翻译被收集并翻译成设备软件，就不需要保留或维护电子表格。
没有计算单元格或宏是此电子表格的一部分。

确定软件是否在范围内

Excel 仅用于格式化信息以便传播和收集设备消息的外语翻译。 乍一看，电子表格似乎是 Excel 这样一个简单的应用程序，人们试图决定它不需要验证

5.2 中提出了以下问题：a 软件的故障或潜在缺陷是否会对医疗设备的安全性或医疗设备的质量产生不利影响？ “

“如果软件或电子表格以破坏存储在那里的信息翻译的方式失败，失败可能会影响设备的安全性，尽管团队认为失败的可能性 对于这种“简单应用”而言，可能性仍然在 ISO 13485 验证要求的范围内。

风险评估

如果设备消息没有正确翻译，则可能导致用户混淆或错误解释消息。 因此，使用该装置的患者可能存在间接伤害的可能性。 软件故障将被检测到，并且在设备开发和验证过程中有许多机会进行交叉检查，以检测和纠正软件的任何故障。
可能对设备软件造成负面影响的预期故障模式如下所示：

- 通过丢失入口文件，丢失单个消息，通过对消息进行错误排序，从而导致丢失上下文，或者通过随机丢失，替换或转换字符来损坏单个消息来破坏原始英文消息的损坏;
- 从区域办事处编写和收集的个人翻译消息的腐败。 腐败可能是由于入口文件丢失，单个消息丢失，消息乱序造成的，从而导致背景丢失或由于随机丢失，替换或转换字符而损坏单个消息。 如果在 Excel 中没有正确安装字体，则可能会损坏任何需要非英文字体的语言;
- 收集的结果电子表格腐败，显示每个翻译结果的积累。 腐败可能是由于入口文件丢失，单个消息丢失，消息乱序造成的，从而导致背景丢失或由于随机丢失，替换或转换字符而损坏单个消息。 除了在电子表格中对行进行乱序外，还可能会发生列错误排列。 在原生字体和字符集中不显示其翻译消息的列在将消息转换为代码时将被软件工程师误解。

验证计划

如果新设备的消息是错误的，软件开发工程师会认识到患者的潜在风险。软件故障的严重程度可能很高。需要做一些事情来建立信心，即电子表格中组织的信息是正确的翻译。

但是，Excel 仅用于组织信息。 Excel 的任何测试数量似乎都不会发现任何会破坏消息的缺陷。当工程师考虑这个问题时，他们抱怨说人为错误比简单的 Excel 应用程序更可能导

致错误。

在思考人为错误时，工程师意识到没有明确的过程来收集翻译或验证没有人为错误进入过程。

工程师创建一个用于收集和验证翻译消息的书面程序。然后他们考虑他们的流程可能存在哪些风险，如何软件（即 Excel 电子表格）失败可能导致这种崩溃，以及最后可以采取什么措施来验证流程，包括电子表格。

风险控制措施

在更好地定义翻译收集过程之后，工程师们确定风险控制措施，以防止过程将错误嵌入到信息翻译中。

保护翻译收集过程的风险控制措施也可以保护软件无法达到其预期用途。

- 当从区域办事处采购时，翻译应以纸质（硬拷贝）格式或电子格式提供，并附带相应的印刷本。如果地区办事处提供电子版本，则该电子表格中的数据将在传输到主翻译电子表格时与硬拷贝相核对（并形成文件）。此验证将防止在传输过程中由于电子表格的损坏导致的结果误解，或者在采用翻译和接收翻译的计算机之间的字体功能差异。
- 一旦收集了所有翻译并放入主电子表格中，就应该将硬拷贝电子表格发送到每个区域办事处进行审核和批准。此区域批准将防止在传输过程中由于电子表格腐败导致的结果误解，或者在采购翻译的计算机与接收翻译的计算机之间的字体功能差异方面出现误解。
- 一旦所有区域办事处，开发批准人和质量保证批准人都接受主电子表格，主电子表格的硬拷贝应作为设备软件开发过程的输入。此外，主电子表格的硬拷贝应该是设备软件翻译验证测试的预期结果的来源。

验证任务

除了这些风险控制措施外，还应完成其他验证和验证任务，以确保软件充分满足其短暂的预期用途。 这些任务如下。

- 对于从地区办事处收集的每个翻译，应更新主电子表格的硬拷贝，并逐行对照个别翻译电子表格硬拷贝的硬拷贝进行验证。 必须核对硬拷贝的硬拷贝，以排除计算机平台或打印机之间字体差异导致的任何错译。
- 版本控制过程应详细记录。 该过程应特别考虑以下内容：
  - 随着设备功能在开发过程中的演变，消息需求（即英语）的变化;
  - 由于提供了翻译，主文件发生变化，并且由区域办事处审查和修改更新后的主电子表格。
- 尽管电子表格非常简单，但一些非常实际的版本控制风险与其使用有关。
- 电子表格的配置应包括电子表格本身的版本号，使用的 Excel 版本，计算机平台配置以及用于创建电子表格硬拷贝的打印机配置。完整的配置非常重要，因为在不同的 Windows 或 Office 安装中以及不同版本的打印机固件中可能存在字体差异。确保翻译不会无意中改变的唯一方法是在使用电子表格时使用相同的配置。
- 电子表格的配置（即操作环境和版本控制）需要进行控制，以防止混乱，不协调的变化。指定一个人负责决定何时更改配置以及何时记录更改历史记录。
- 每个电子表格的版本应该在其硬拷贝版本中可见。
- 设备软件中的转换表应指出哪个版本的硬拷贝主电子表格被用作翻译消息软件的输入。
- 个人翻译验证任务应包括以下内容。
  - 通过比较电子表格的主版本和翻译版本，应该逐行验证英文信息。 这种比较可以防止文件传输到区域办事处时以及区域办事处返回文件时可能发生的电子表格文件的任何损坏（例如损坏或丢失的消息）。

- 将翻译插入主电子表格后（手动或通过使用 Excel 的剪切/粘贴功能），应修改主电子表格的硬拷贝输出的逐行比较， 翻译电子表格。
- 当测试设备软件的消息实现时，测试程序应使用主电子表格最新版本的硬拷贝（并应引用版本号），以便将实现的消息与预期的消息进行比较。

所有这些验证任务都应作为验证过程和 Excel 电子表格的客观证据进行记录和收集。

这种验证方法可以 100％验证输出与软件的输入。 计划不再对电子表格进行测试。

尽管缺乏传统的测试，工程师们对他们的过程充满信心，并相信他们的验证原理是一项宝贵的工作。 工程师认为软件的任何故障都将被检测到，并且他们通过在过程中的适当位置收集并记录的硬拷贝具有恢复路径。硬拷贝和记录的逐行验证提供了活动的书面证据。

保养

电子表格旨在满足短暂的需求。 一旦翻译的信息被嵌入到代码中，它就会被退役。 没有维护计划创建。

讨论

电子表格的预期用途和初始风险分析对确定电子表格需要进一步确认注意力至关重要。在其他预期的使用情况下，完全相同的电子表格可能会导致电子表格风险低且肯定复杂度低的结论。 如果预期的用途仅仅是为了追踪翻译收集的进度（即，电子表格上的翻译将不会用于设计中的投入活动），那么可能确定的是实际上没有风险存在 设备的完整性，事实上，电子表格是一种商业管理工具，甚至不属于该法规的范围。

该软件“自动化”的“过程”是数据收集，设备格式化和信息翻译存储过程的一部分。从几个角度来看，这个例子很有趣。

- 验证需要很少的软件测试来验证软件的使用。请注意，软件（Excel）和电子表格已针对此特定用途进行了验证，但未进行任何用途的一般验证。该团队认为测试不太可能发现软件中的任何缺陷，但是如果软件确实以某种不可预知的方式出现故障，那么设备存在漏洞。
- 验证包括 100％验证电子表格的输出。硬拷贝版本依赖于 ugold 标准。“一旦硬拷贝被批准并用于设计历史文件，任何随后的软件故障都是无关紧要的。软件在批准之前的任何失败都会被审查所捕获和审批流程。
- 修改了“过程”，使其免于电子表格软件的任何故障。
- 工程师们认为，人为失败的可能性远高于此应用软件失败的可能性。用户可能会发生印刷错误，可能会使用电子表格的错误版本或可能发生类似的错误。在这种情况下，“软件验证”也使该过程更容易受到人为错误的影响。
- 这个例子强调了配置管理的重要性，即使是常规办公生产力工具也是如此。

注：这个例子是基于一个并不那么干净的真实案例。 在实际情况中，电子表格的版本控制会导致人为错误。 意外的是，与在不同 PC 上不同安装的 Excel 链接的字体版本相关的问题给出了不同的硬拷贝结果。（打印机字体在不同的打印机上也出现了问题。）看似简单的电子表格，几乎被认为不需要验证的电子表格实际上在其消息翻译的损坏方面变得有问题。

实施例 8：参数灭菌器

玛丽一直负责领导一个新的自动灭菌器系统的验证工作，该系统将由她的公司 Always Medical Device Company 定制开发。

定义过程

玛丽首先定义并记录了她所了解的关于正在引入她的工厂的 100％环氧乙烷（ETO）灭菌过程。

- 手动将医疗器械放入消毒器中。 该过程包括灭菌周期参数评估以支持参数化释放。
- 自动灭菌器系统软件控制灭菌循环活动。
- 循环完成后手动移除医疗设备并将其转移至脱气室。

过程风险分析

玛丽非常关心这个过程带来的风险。 这一过程失败可能会造成严重后果，其中包括：

a）医疗器械消毒不当。 这种故障可能导致严重伤害或死亡，这是由于使用非无菌产品导致的感染;

b）设备历史信息丢失和产品可追溯性;

c）向制造设施或环境释放有毒化学品。 这种故障可能导致当地社区的灭菌器操作人员或个人严重受伤或死亡。

因此，玛丽考虑应采取哪些风险控制措施并进行验证以减轻这些风险。 玛丽认为，通过使用参数灭菌技术可以控制风险，以确保在正确的温度和正确的相对湿度下，适当的时间使用适量的气体。 此外，手动检查消毒器的数据以获得适当的参数值将独立确认消毒是否足够。 最后，她认为需要故障安全关闭和密封结构来控制化学品泄漏进入设施。

有了这些风险控制措施，多个同时发生的系统故障将不得不导致无菌设备。 但是，由于这种故障发生的影响，Mary 确定剩余过程风险很高。 因此，严格的验证是适当的。

定义软件的目的和意图

玛丽希望详细了解如何使用该系统中的软件。首先，她考虑软件应该做什么。在这种情况下，该软件控制使用 100％ETO 消毒容器对医疗器械进行灭菌的过程，包括记录包含在器械历史记录中的信息以及分析灭菌值以支持参数释放。购买新的灭菌器是因为它可以容纳比当前系统更大的批次;需要更大批量的产品来满足当前的产品需求。灭菌操作员将使用该系统和质量保证团队来确定释放医疗器械的可接受性。玛丽明白，这项工作将通过在灭菌周期中对灭菌容器进行实时控制和监测并将信息存储在数据库中来进行。玛丽很高兴知道该系统将实际位于现场灭菌设施中，此外，该系统通常会每周关闭一天，以便进行任何必要的维护。

玛丽确定软件将自动化灭菌周期的各个方面，从手动将设备放入容器到手动将设备从容器中取出。

玛丽记录软件的目的和意图如下。

- 灭菌软件将控制和监测灭菌过程，并将评估参数释放的灭菌周期参数。

验证计划

现在玛丽明白软件的目的是做什么，她已准备好在高层次上制定验证计划。她知道她以后需要添加更多细节，但现在要开始验证计划，以便她能够以知情的方式识别软件故障风险，并使用已识别的风险来完成她的计划。

由于她早先发现的高剩余流程风险，玛丽认为她需要在验证工作中提供细节和形式。她期望在文档中使用高级别的严谨和细节，并将大多数文档作为独立文档而不是合并文档，这通常是为了减少工作量而完成的。由于与系统相关的高风险，她决定以与开发医疗设备软件时使用的严格程度相同的水平对待开发。因此，她决定遵循 EC 62304：2006 / AMDl：2015 作为生命周期控制方法论。有关软件风险管理的指导，她参考了 IEC / TR 80002-1。此外，为了确保所有可能的伤害源都被考虑在内，Mary 决定将软件故障树分析应用于开发工作。她

还决定形式化定义和记录用户业务流程要求和软件要求。任何特别关注的功能都将被明确标识。玛丽还安排正式的软件需求审查。质量保证小组，灭菌工程师和灭菌经理将要求批准。由于该系统的重要性和风险，验证报告的最终批准将包括高级管理人员。

定义软件需求

玛丽现在编写软件需求定义。她决定软件要求应该处理报警，错误处理和信息，确认参数设置，与设备历史记录系统的接口，传感器控制和监测，运动控制和监测。

建立对软件的信心和控制

Mary 使用 Always-Safe 的内部开发控制程序作为驱动程序，在整个开发生命周期中使用内部控制。 因为一切都是在内部完成的，所以不需要进行供应商活动。

定义与其他系统的软件边界

玛丽然后考虑新的灭菌器需要与其他系统连接。 她确定唯一的接口将与 Always-Safe 的现有数据库系统一起使用，该系统将存储灭菌周期中生成的数据。

分析软件故障风险

尽管玛丽已经确定自动化业务流程具有高风险，但她仍然需要分析软件故障的风险。以此文档为参考，Mary 为此活动选择了一个定量风险模型。 她将新系统排列如下。

- “严重性”的风险很高（10），因为系统故障可能导致死亡或严重伤害。
- “可能性”的风险也很高（10），因为软件故障本身可能导致伤害，因为软件正在确定灭菌的可接受性。

她计算出 20 分的风险分数，这转化为高风险分类。 高风险分类意味着应该应用严格的验证方法。 遵循的方法是严格和全面的，好像灭菌器本身就是一种医疗设备。

由于缓解措施，该自动化系统的剩余风险尽可能低。 由于来自系统的伤害的严重性，绝育本质上是一个高风险的过程。还执行了来自 IEC / TR 80002-1 的与风险相关的其他活动。

完成验证计划

因为玛丽现在已经完成了软件需求的定义，决定了实施方法并分析了软件风险，她有足够的信息来完成详细的验证计划。

在编写验证计划的初稿时，Mary 已经决定采取严格的风险管理方法。 她已经计划以高度正式的方式对待验证工作。

因此，她描述了她打算如下使用的风险管理工具（在 IEC / TR 80002-1 中标识）。

- 风险管理工具：
  - 软件故障树分析;
  - 风险管理计划;
  - 在制造或业务流程中识别风险控制措施;
  - 软件故障分析（风险分析）

然后玛丽考虑在软件设计，开发和配置阶段她将如何获得软件的信心。 她已决定遵循 IEC 62304：2006 / AMDl：2015 的生命周期控制。 她现在确定了将用于确保软件在设计，开发和配置阶段正确开发的其他特定工具。

- 设计，开发和配置工具：

  IEC 62304：2006 / AMDl：2015 架构文档和审查;
  - 设计规范;
  - 软件详细设计和审查;
  - 软件编码标准;
  - 可追溯性矩阵;
  - 在软件系统设计中识别风险控制措施;
  - 代码审查和代码验证;

- 开发和设计评审。

玛丽毫不怀疑她需要广泛地测试这个新系统。 她首先决定，正常的单元测试，集成测试和界面测试活动将需要正式的测试计划活动。但是，由于该系统将实时发布完成的设备，她决定通过压力测试，性能测试和更广泛的输入测试组合来推动系统的极限，以模拟尽可能多的操作条件。

- 测试工具
  - 测试计划;
  - 单元测试;
  - 集成测试;
  - 接口测试;
  - 回归测试（如有必要）;
  - 软件系统测试;
  - 健壮性（压力）测试;
  - 输入测试的组合;
  - 性能测试。

最后，Mary 知道系统在生产环境中完全实现之前还没有完成，所以她把注意力转向了她希望在部署阶段看到的验证活动。 她希望确保系统有充分的文件记录，并且用户在正确使用方面训练有素。 她还希望确保该系统实际上按预期安装。 所以 Mary 的部署阶段的验证计划现在包含以下项目：

- 部署工具：
  - 使用程序审查;
  - 内部培训;
  - 安装资格;
  - 运营和业绩资格;
  - 运营商认证。

规划维护

由于高度的剩余风险，玛丽担心软件的维护。 她计划进行一些维护活动，以确保系统部署后的软件质量，包括评估用户培训的有效性，系统监控技术，系统输出的正确性检查和缺陷报告。 她还确认除了软件维护活动之外，还发生了校准和其他硬件维护活动。

退休活动

玛丽努力退休之前的系统，因为该系统生成的数据需要归档以便于设备历史记录的目的，旧格式与新格式不兼容。 新系统采用通用数据格式，在退役时可以灵活地将剩余数据迁移到新系统。

示例 9：不合格材料报告系统 - 整个系统升级

先进医疗公司正在升级其不合格物质报告系统（NCMRS）软件 - 一种商业软件包。高级医疗在最后一个重要方面没有选择升级，所以该系统现在正在运行两个主要模块。（Advanced Medical 目前运行的是版本 2，最新版本是版本 4.）维护当前的软件维护

协议，高级医疗需要升级。 NCMRS-Pro 软件的第 4 版与以前的版本相比有了很大的改变。除此之外，该产品已从典型的客户端 - 服务器应用程序重新编译为基于 Web 的应用程序。新软件还包含重要的新功能和新功能。高级医疗公司的业务流程所有者和项目经理 Frank 对他现有的软件和流程没有新的要求，但他希望利用新的软件功能。

弗兰克与监管团队进行磋商，并决定 ERP 系统和 NCMRS 之间的当前界面可以保持不变，而无需修改。然而，弗兰克认识到，新版本能够将数据写回 ERP 系统，并且在验证过程中这个扩展接口应该受到彻底的挑战。弗兰克和他的同事，制造质量工程师和监管团队开始确定验证工作的范围。这个小组在本例的其余部分被称为“小组”。

定义过程

弗兰克从分析他当前的手动流程开始，以确定新软件将自动化哪些工作流程元素。 新软件将改变以下功能：

a）认可潜在的不合格材料或产品（超出范围）；

b）输入与材料有关的信息和围绕其发现的情况（范围内）；

c）路由信息以便对材料进行适当的识别，评估，调查和处置（范围内）；

d）向正确处理财务，采购，计划和时间安排交易（范围）所需的重要利益相关者和其他计算机系统分发信息;

e）物质的物理处置，尽管关于处置的相关数据将被记录在系统中（超出范围）。

过程风险分析

弗兰克知道这个过程和支持软件存在风险。 该过程失败可能会造成严重后果，其中包括：

- 无意中将不合格材料释放到生产车间内;
- 无意中将不合格产品发布为商业分销;
- 增加了因废品，返工等造成的成本或制造成本。

弗兰克和监管团队考虑采取哪些风险控制措施来缓解这些风险，包括以下方面：

- 程序控制以检测，隔离，控制和纠正不合格材料;
- 管理和质量审查统计过程控制数据及其措施，以确定发展趋势，这些趋势可表明过程未得到适当控制;
- 对运营商进行持续培训以确保遵守程序;
- 财务报告，以帮助识别材料使用，这将暗示制造工艺中存在一些非常规问题。

有了这些风险控制措施，就会出现多个同时发生的系统故障，导致无法正确控制不合格材料或产品。 但是，由于这些故障可能对质量，监管和财务造成影响，Frank 认为剩余的过程风险需要进行严格的建立信任活动，以帮助确保软件正常运行并且符合预期用途。

定义软件的目的和意图

弗兰克希望详细了解软件升级将如何影响他的用户和组织。弗兰克得出结论，该软件本质上是一个自动问题跟踪和管理工具。使用标准工具，设备和其他仪器的制造人员负责识别和隔离潜在的不合格材料和产品。一旦问题得到认可，有关情况的详细信息将被输入到软件中。软件然后管理工作流程，分配和通知来解决问题并记录处理材料和产品处置所需的各种活动。软件升级应该简化流程，从而提高流程效率，并且应该为质量保证团队提供更强大的工具来分析和趋势数据，并提高团队在质量问题上的可见度。弗兰克明白，升级所需的流程变更主要针对工作流程和信息分配。软件本身没有做出最终决定，也没有独立确定任何结果，但是软件确实保存并记录人类与系统交互作出的决定。

Frank 认为该软件将自动化不合格处理的工作流程方面。以下关于软件目的和意图的声明由监管团队撰写。

- NCMRS 软件旨在支持处理不合格材料和产品。 该系统用于记录 SOP 定义的过程步骤，并记录所执行的过程步骤，执行时间，执行过程的人员以及每个步骤的结果。 该系统使数据随时可用于质量监控和改进活动。

定义软件与其他系统的界限

NCMRS 软件有两个界面，包括一个与 ERP 系统的主界面和一个与公司人力资源（HR）系统的次界面。 主要接口设计为每日两次的预定批处理过程，以使用成品，在制品，物料清单和工序清单数据更新系统。 该界面还将向 ERP 提供不合格物料报告（NCMR）数据，并提供有关质量保留，物料处置和其他交易信息的数据。 次要接口是 HR 系统的单向接口，旨在更新 NCMR 员工数据以用于计划和分配目的。

初始验证计划

弗兰克获得了更多的信心，即 NCMR 过程和软件得到了充分的理解和适当的记录。 他现在准备开发一个高级验证计划。 其他细节将在规划过程中制定。

监管团队确定将增加最大价值的文件，并充分指定软件预期的功能。 这些文件通常被称为“要求”，但本身并不是一套典型的用户权限。 相反，这些文档将成为软件如何运行的一系列详细描述。 因此，自动和手动测试的分析将在其审查和结果方面更加定性。 它将整体看待输出结果以及系统是否按预期执行，而不是单独测试以确定是否满足特定的用户需求。文件集将包括以下内容。

a） 工作流程和业务规则文档。 软件的这个区域是可配置的，所以团队将准备好所需的一组配置，并将开发详细的过程流程和描述操作的逻辑图。

b）接口文件。 这些文件将描述哪些数据元素从 ERP 和 HR 系统移动到 NCMRS，哪些数据元素从 NCMRS 移动到 ERP 系统，以及这些元素何时移动。

c）数据迁移文件。 这组文档将描述哪些历史数据将被移至升级后的系统。

验证计划将包括审查和批准每个这些文件的结果。 质量保证小组，制造工程部门和信息系统小组将要求批准。 由于系统的重要性和风险，高级管理层的所有成员都应最终批准验证报告。

定义软件使用要求

Frank 和团队成员参考了上述组装文档集的业务，参考了供应商提供的系统文档和现有接口的先前文档。

建立对软件的信心和控制

Frank 对这个软件和供应商有着积极的经验。弗兰克现在确定了团队将用于建立对软件的信心的五项主要努力。

a）供应商根据 Advanced Medical 的内部政策和程序具有与公司的核准地位。先前的审核表明供应商具有适当的质量体系和 SDLC。供应商生产商用软件，这些软件在受管制的行业中有着与 Advanced Medical 预期用途相似的用途。供应商将定期审核以保持批准的状态。

b）Advanced Medical 将使用供应商提供的自动化测试工具来验证软件是否已正确安装，并在测试套件的边界内运行。这个工具可以在几个小时内处理 8000 多笔交易。然而，该工具并不测试该公司计划包含的某些配置选项。

c）该团队将制定附加测试计划，其中包括并行处理实际纸质不合格报告的统计显着抽样。将对产出进行审查，以确保准确性，数据完整性和符合程序。

d）该小组将通过采用确保历史记录保持其完整性的抽样技术来验证数据转换和现有系统记录的迁移。记录计数将用于验证 100％的转换。

e）数据接口将使用采样技术进行验证，以测量数据传输的完整性和准确性。

分析软件故障风险

Frank 将此文档用作参考来确定将要求的验证严格性。软件故障可能导致电子记录丢失，损坏或处理不当。风险的缓解由供应商的内部质量体系，软件的安装认证（自动化测试工具）以及辅助用例测试和验证来控制。由于下游过程控制，该系统的剩余风险被视为尽可能低。

最终验证计划

这一决定意味着将采用相当严格的验证方法。所遵循的方法在合理的程度上确保软件将按照预期执行。团队成员得出结论：他们已经充分定义了系统的要求，他们已经决定实施方法，他们已经分析了软件风险，并且他们已经获得了足够的信息来进行详细的验证计划。

要执行的大部分测试将使用自动化测试套件完成，该测试套件已经被团队审查并确定对此用途有效。 额外的附加用例测试将使用源自制造车间的实际业务案例进行。 这些测试的目的是 a）验证过程是否按预期工作，b）加速用户接受和培训，c）验证配置更改没有对软件产生不利影响。 附加测试的目的不是取代供应商的内部系统测试，该测试以前已通过审核进行验证。 自动化测试的成功完成将确定软件安装正确且功能可接受。

该团队从本文档中选择以下工具来执行剩余的安装，配置，测试，验证和验证工作。

- 设计，开发和配置工具：
  - 架构文档和审查;
  - 在软件系统设计中识别风险控制措施;
  - 配置设计评论;
  - 审查供应商的“已知问题”清单;
  - 审查供应商的基本系统验证文件;
  - 审查“开箱即用”的软件工作流程图;
  - 审查“开箱即用”的标准报告库;
  - 对标准工作流程和业务规则进行配置更改的差距分析。
- 测试工具：
  - 测试计划;
  - 供应商提供的用于安装验证和鉴定的自动化测试工具的描述和结果;
  - 安装和性能测试（自动化测试套件的一部分）;
  - 用例测试覆盖了使用实际的不合格记录作为输入而不是人工构建的测试用例的配置变更;
  - 抽样计划以验证迁移的数据;
  - 系统检查以验证操作界面。
- 部署工具：
  - 使用程序审查;
  - 内部培训;
  - 运营商认证。

规划维护

Frank 计划在系统部署后使用多种维护活动来确保持续的软件质量，包括在内部和向供应商评估用户培训的有效性，系统监控技术，定期审核系统输出和缺陷报告。 Frank 已经与供应商建立了联系，以便在 Advanced Medical 负责维护软件的合适员工注意到错误，维护版本和其他通信的通知。

退休活动

弗兰克计划一旦切换发生，就可以保持当前系统的可用性，以此来比较吞吐量和可以编译性能指标的结果。 新升级成功运行 6 个月后，弗兰克将完全停用先前的系统。

示例 10：用于安排不合格材料报告（NCMR）审核董事会会议的软件

一家拥有 1000 名员工的公司决定尝试一种新的软件解决方案，帮助公司以电子方式安排所需的 NCMR 审查活动的会议。被分配实施自动化的项目组学习了刚刚发布的商业软件程序。供应商声称，该软件可以使用通过其他计算机化系统接口接收的数据安排会议。如果软件能够从公司经过验证的 NCMR 数据库系统收集 NCMR 数据，项目团队决定该软件可能适合安排公司的 NCMR 评审委员会会议。

定义过程

团队聚在一起讨论安排 NCMR 审查委员会会议的过程，并审查公司的 NCMR 处理程序。讨论结果在以下定义的过程中。

a）一旦发现不符合，相关材料将被贴上标签，分离并登录到经过验证的 NCMR 数据库中。

b）每周举行一次会议，审查所有与不合格情况相关的调查结果和建议的处置行为。

c）对于每次会议，确定可供审核的 NCMR 清单，以及需要参加的人员，介绍结果并参与处置行动和批准。

d）在 NCMR 审查委员会会议前一天，向需要参与的人发送会议请求。这项要求包括将要讨论的 NCMR 清单。

流程风险分析

通过头脑风暴活动，团队成员评估此过程中可能出现的失败可能带来的危害：

- 会议请求未发送;
- 会议请求没有在正确的时间发送;
- 请不正确的人参加;
- 确定一份不正确的 NCMR 列表供审查。

团队注意到，公布的 NCMR 处理程序要求将一个人指定为 NCMR 处理经理。 该人员负责确保及时处理所有 NCMR，并从 NCMR 数据库中发现的数据发布 NCMR 处理指标。 在团队发现的所有情况下，会议安排软件造成的危害都会影响 NCMR 审查委员会会议的效率。这种中断给 NCMR 处理经理的时间带来了额外的负担。因此，过程故障风险分析在监管风险，环境风险和对人类危害的风险方面被确定为低。

定义预期用途

该团队定义了软件使用的目的和意图，监管使用和边界如下。

- 软件使用
  - 谁？ 软件将主要由 NCMR 处理管理器使用
  - 什么？ 软件会自动发送电子会议邀请给被确认需要参加本周会议的人员。
  - 什么时候？ 需要安排 NCMR 会议时将使用软件。
  - 哪里？ 因为所有与会者都是本地的，所以软件只需要在局域网（LAN）上使用。
  - 怎么样？ 软件检索一系列 NCMR，这些 NCMR 是开放的，需要由 NCMR 审查委员会审查。 NCMR 处理经理在下次会议中确定要审查的 NCMR。 然后，软件使用由 NCMR 处理管理器设置的表格来识别需要参加给定会议的个人。 会议日期由 NCMR 处理经理识别。 一天前，软件向适当的参与者发出电子会议邀请。
  - 为什么？ 软件将用于改善及时通知适当人员参加每周 NCMR 审查委员会会议。
- 边界
  - 该软件的界限位于 NCMR 数据库和图形用户界面的界面上。
- 监管使用
  - 该软件不存储任何用于证明符合任何监管要求的信息。 所有与 NCMR 有关的设备历史记录信息或 NCMR 的处理都记录在纸上或经过验证的 NCMR 数据库中。

在创建和审查这份目的和意图声明后，团队确定所提议的软件不会自动执行法规要求的

活动，也不会创建法规要求的质量记录。虽然它被用于促进作为受监管活动（NCMR 过程）一部分的会议，但该软件本身不会使受规管活动自动化。结果，团队记录了以前列出的预期用途，并清楚地表明不需要正式验证。然而，团队也认识到，在维护阶段使用量的小改动可能会显着影响团队的原始验证决策。例如，如果软件用于存储会议纪要或用于制作出席会议以供监管调查员审核的个人清单，则原始“超出范围”决定将受到影响。因此，团队会更新其质量体系程序，以便定期评估预期用途或对相关过程进行更改。

工具箱的使用

采用了以下工具。

- 开发定义阶段：
  - 过程需求定义;
  - 过程故障风险分析;
  - 预期用途定义。
- 保持阶段：
  - 计划维护

讨论

由于识别了该软件自动化的具体用途和活动范围，因此团队能够适当地声明该软件不符合医疗器械质量管理体系流程软件的定义，因此该软件不能 需要验证。 在识别此类软件时应格外小心，以确保软件的实际使用完全涵盖在预期的使用定义中。 即使在软件没有改变的情况下，在生命周期的维护阶段也很容易改变预期的使用，这也很重要。 因此，维护计划是确保对公司使用的软件进行适当控制的重要部分。

示例 11：批准的供应商列表系统

Acme 公司是 B 级医疗器械制造商。 该公司一直使用手动程序来维护经批准的供应商清单（AVL）。 Acme 公司希望开发 AVL 系统来自动检查供应商是否被批准提供特定部件。
新的 AVL 系统的 Acme 项目经理 Jack 确定，AVL 过程是与 ISO 13485 中的采购控制相关的医疗设备质量系统过程。
因此，拟议的 AVL 系统符合软件验证的要求。

定义过程

为了更好地理解开发 AVL 系统所涉及的要求和风险，杰克如下定义了相关的业务流程。
a）当工程组要求批准新供应商时，供应商零件的样品将提交给质量组进行资格认证。
b）质量组对供应商的部件进行认证后，质量部门向采购部门发送一封电子邮件，授权输入供应商的名称和批准的部件号和说明到批准的供应商清单（AVL）中。该清单在采购组的纸张上维护。接收检查组可以访问 AVL。
c）采购组执行手动检查以确认供应商的名称已正确添加到 AVL。
d）当采购组订购部件时，它指 AVL 确保供应商被批准并且供应商有权提供所要求的部件。
e）如果供应商被批准，采购组织签署请求表明他们已经检查了 AVL。

分析流程风险

杰克然后考虑当前过程中会出现什么问题。如果进程出现故障，可能会从 n 个未经批准的供应商处获得零件，或者是因为未批准的供应商已添加到 AVL 中，或者是因为在订购零件之前，采购组未能检查 AVL。

然后杰克考虑采取哪些风险控制措施来缓解这些风险。杰克发现采购团队已经制定了一个程序来手动检查供应商名称是否已正确添加到 AVL 中，并且对列表的访问仅限于授权员工。此外，杰克认为目前的采购程序要求采购团队签署该供应商

在发出采购订单之前在 AVL 上。确保订单与经批准的供应商联系的控制措施位于接收检验部门，AVL 在收到部件后再次进行检查。

在这些风险控制措施的基础上，杰克确定剩余过程风险较低。因此，他怀疑新的 AVL 系统可能是一个低风险系统。

定义预期用途

现在，杰克将自己的业务流程自动化，并写下了拟议的新 AVL 系统的目的和意图声明。

- AVL 系统将根据电子 AVL 列表自动检查供应商和零件，以确保仅从授权供应商订购零件。 新系统将采用 AVL 数据库，该数据库将与现有采购订单系统相关联，供应商资格审核流程期间由总部 Acme Corporation 质量部门使用，并在采购订单生成过程中通过采购代理商使用。

Jack 还考虑了 AVL 系统将与之接口的其他系统和过程，并在他的陈述中增加了一些语言，以澄清新系统的界限

- 采购流程将与 AVL 系统自动进行的流程接口。该界面将包含对 AVL 数据库的查询，以了解采购订单上指定的供应商的状态。采购流程不能确认 AVL 中数据的准确性，也不会与供应商评估流程相互配合。

验证计划

既然杰克了解业务流程是自动化的，并且已经确定了新系统的目的和意图，他已经准备好在高层次上制定验证计划。他知道他会在稍后详细讨论这个计划，但他现在要开始验证计划，以便他能够确定所需验证工作的级别。

此前，Jack 确定现有 AVL 流程中存在较低的剩余流程风险。因此，他认为他不需要在验证工作中提供非常详细的信息或形式。他知道为新系统定义用户业务流程要求和软件要求非

常重要。但由于系统风险较低，杰克无需在每份文件上单独签署单独的文件。因此，他决定使用表格格式将用户业务流程需求，软件需求和测试计划组合成单个文档。

此外，由于该系统风险低，杰克认为不需要对验证工作进行广泛的管理评审。他决定供应商开发经理和质量保证代表的批准应该足够了。但杰克也相信，为了确保用户的要求是正确的，他还应该由采购组的代表添加评论。

杰克根据他的决定开始他的验证计划草案。 Acme Corporation 拥有所有验证计划应遵守的标准格式。验证计划的某些部分未定义，但杰克将在初始系统设计获得批准后更新计划。

定义软件需求

杰克现在写软件要求。他决定软件需求应包括“什么”（AVL 流程或系统所需的操作），AVL 系统如何与购买系统连接的接口规范，数据字典以及新系统的有效查询示例应该能够处理。

定义与其他系统的软件边界

Jack 然后考虑新的 AVL 系统将需要与其接口的其他系统。他确定唯一的界面将与 Acme 的现有采购系统一起使用，该系统可以通过简单的结构化查询语言（SQL）查询来查询 AVL 数据库。

建立对软件的信心和控制

杰克现在需要决定他应该使用什么方法和技术来构建新系统。由于业务需求相当简单，交易量会很低。由于 AVL 系统是一个低风险系统，杰克决定使用 Microsoft Access2）开发它，这是一种广泛可用且易于使用的数据库系统。

由于微软是一名外部软件开发人员，杰克需要决定他应该执行什么类型的活动来建立对 Microsoft Access 的信心。杰克指出，微软 Access 是一个广泛使用的工具，并且在过去，这个产品的任何问题或问题已经在互联网留言板上被迅速识别和公布。结合 AVL 系统是一个低风险系统这一事实，Jack 决定他不需要为 Microsoft 作为数据库开发人员进行供应商审计。

由于新系统包含电子记录，Jack 决定围绕 Microsoft Access 实施第三方“包装”软件，以提供必要的控制措施以确保记录的有效性。

分析软件故障风险

虽然杰克已经确定自动化业务流程的风险很低，但他仍然需要分析软件故障的风险。他决定使用此活动的定量风险模型（范围为 1 到 10）。 他将新系统排列如下。

- 杰克将“严重性”定为中等（6），因为软件故障只会间接造成伤害。 他将此排名建立在流程中存在下游控制的基础上。
- 杰克将“可能性”评为低（1），因为数据库设计非常简单，使得在测试过程中不会发现严重错误的可能性较小。
- 排名的组合转化为低风险分类。

因此，杰克将执行适合低风险的验证任务。

完成验证计划

现在 Jack 已经定义了软件需求，决定了实施方法并分析了软件风险，他有足够的信息来完成验证计划。在这一点上，他退后一步，根据他对系统所知道的一切，实施方法和软件风险问问自己：以下问题：“什么验证活动真的会让我相信这个系统适合它有可能的使用？”。

由于该系统是购买的数据库工具，风险相对较低，杰克认为他计划的验证活动已足够，但他需要解决环境要求，以确保操作系统和 Access 版本的更改得到良好控制。他更新验证计划以呼吁进行正式的软件配置控制。

由于该系统正在由第三方开发，因此 Jack 需要确定开发人员能够正确地转换对定制，输入，接口，数据存储和输出的要求。由于该系统将取决于现有系统的输入，Jack 在验证计划中添加了接口测试和集成系统测试作为重要活动，以确认开发人员工作的正确性。

最后，Jack 希望确保开发人员在开发过程中保持适当的版本控制，因此他将软件版本控制作为其验证计划的必需活动添加进来。
因此，杰克的批判性思维引导他为剩余的开发和验证工作包含以下工具。

- 设计，开发和配置工具：
  - 软件架构文档和审查;
  - 可追溯性矩阵（需求规范中固有的）;
  - 风险控制措施（用户规范中记录）。
- 测试工具：
  - 整合测试（记录在要求中
  - 规范
  - 阳离子）;
  - 接口测试（在需求规格中记录）;
  - 软件系统测试（在需求说明书中记录）。
- 部署工具
  - 用户程序审查;
  - 应用程序的内部培训;
  - 安装资格。

规划维护

一旦部署系统，杰克就会考虑采取哪些活动来确保软件质量。 鉴于系统残留风险低，杰克决定应该每季度审查数据库中 AVL 数据的准确性。 杰克在他的验证计划中包含一个部分，用于记录季度评估，并发布一个程序的开发和实施请求，以确保在系统正式运行后进行季度评估。

例 12：校准管理软件

XYZ 医疗公司正在快速增长。 XYZ 已经收购了欧洲和亚洲的公司。公司的发展意味着 XYZ 的校准管理需求也在不断增长。目前，XYZ 校准管理器保存一本包含所有校准信息的书籍，并且每周检查校准的设备库存以确定是否有任何项目需要重新校准。随着公司的发展，其库存变得过大，并且在全球范围内分散管理一个人使用纸张系统。现在是把计算机系统安装到位的时候了。

定义过程

XYZ 医疗有一个标准的操作程序（SOP），要求计算机化的系统自动化部分质量体系的预期用途。 XYZ 首先定义校准管理过程，以确定过程中固有的风险，并确定软件解决方案是否将自动化全部或部分当前过程。 XYZ 经理审查校准管理 SOP，其中包含以下步骤的详细信息：

a）采购新设备;
b）新设备具有唯一的身份证号码;
c）确定校准程序;
d）新设备已校准;
e）校准状态记录在设备上;
f）维护校准记录，包括校准要求，状态和到期日期;
g）搜索校准记录以进行报告和校准管理活动。

流程风险分析

无论是纸质系统还是电子系统，校准管理过程都会带来一些固有风险。与流程相关的风险如下。

- 在校准过期后使用一件设备时，会记录错误的测量结果。这个问题会产生很多后果，具体取决于设备和使用过程中的阶段。
- 放置在一台设备上的标签不正确表示设备在实际校准时未进行校准。这个错误也有一些后果，取决于设备和过程中的阶段。
- 校准记录可能会丢失，并且可能会发生校准失效的设备积压。这个问题可能会延误工作。
- 如果校准状态记录不正确，可以使用过期的设备。
- 如果两件设备收到相同的识别号码，则记录不会是唯一的。

由于不正确校准的潜在结果，该过程被确定为高风险。 在最坏的情况下，校准设备可以用来测量医疗设备的最终验收，并且设备在不应该有的时候可以被赋予可接受的状态。

为了缓解这个问题，XYZ 经理应该更新 SOP 以包含对设备每个用户的说明。 每个使用者在使用前应检查设备上的校准过期标签。 在执行使用校准设备的协议期间，每个用户应记录设备 ID 号和所使用设备的校准截止日期。 用户还应接受培训，了解如何识别需要校准的物品，并教导不要使用任何标签已过期或没有标签的设备。

执行这些措施会导致系统的剩余风险缓和。 XYZ 经理认为用户指导是一种适当的措施，但不足以将剩余风险降低。

定义软件的预期用途

软件系统不会执行校准活动; 它将是一个数据库，其中包含有关设备及其校准历史和状态的校准信息和数据。 软件系统将控制校准过程的步骤 b），f）和 g）。

XYZ 经理同意 XYZ 将为以下目的和意图验证系统。

- 校准管理系统用于为需要校准的设备提供识别号码，为校准设备打印标签，存储校准结果数据并报告设备的校准状态。 校准管理系统自动完成部分 ISO 13485 要求，包括检测，测量和测试设备。

验证计划

为了确定验证活动的阶段，XYZ 经理通过设置对可交付成果内容的期望以及交叉功能组参与过程来启动验证计划。 他们记录下列步骤。.

- 为所选工具定义严格的文档级别。
  - 系统的文档严格性将适中。 因此，主要可交付成果将被单独创建和批准。
- 确定所选工具的审查水平（管理和跨职能参与和审查）。
  - 鉴于该系统将在全球范围内用于校准管理，因此全球信息技术管理和运营管理部门应以系统验证计划和验证报告的批准形式在系统验证过程中具有可见性。 此外，新的场地设备经理将参与所有文件的审查和批准。
- 从工具箱中选择“定义”工具：
  - 用户和业务流程要求;
  - 软件要求;
  - 正式的软件需求审查。

定义软件需求

软件需求将包含以下内容：

- 功能工作流程;
- 电子记录和电子签名要求;
- 数据逻辑要求;
- 报告要求;
- 设备标签印刷专用要求;
- 用户安全和配置文件;
- 性能要求;
- 容量定义。

建立信心并控制软件

XYZ 经理对这类产品的三家供应商进行调查，并确定一家供应商的产品 b est 与 XYZ 的计划用途相符。 该系统的供应商广泛用于医疗设备行业，尽管该版本的产品是全新的。 可以从以前版本的记录中获得一些机会，但是已知的缺陷分析将在当前报告的问题的基础上进行，并且测试开发小组将对新版本的功能进行特别的审查。

定义与其他系统的软件边界

该软件没有与其他软件系统的接口。

软件风险分析

验证团队与全球校准人员坐在一起，并使用表 C.16 中的问卷确定软件风险。 他们首先识别风险，然后确定这些风险的风险控制措施。 最后，他们评估剩余风险的可接受性（见表 C.17）。

表 C.16 - 实例 12 - 风险分析

	风险识别问题	表示“是”或“否”。 如果“是”，则分配风险标识符（风险 1，风险 2，风险 n）
1.1 产品安全（危害）	如果软件出现故障，是否存在产品安全的潜在风险？ 是的，在所有情况下。 校准设备可能会被软件误认为校准设备。 - 患者伤害 - 是的。 如果使用非校准设备进行测量，则可能会在患者	风险 1 - 使用不校准设备。

	身上使用超出规格的产品。 - 操作员伤害 - 是的。 如果测量温度或力量错误，操作人员可能会被夹伤或受伤。 - 旁观者伤害 - 是的。 这种伤害是依赖于设备的。 - 服务人员伤害 - 是的。 如果测量温度或力量错误，服务人员可能会受伤或受伤。 - 环境危害 - 是的。 如果压力测量不正确且容器中含有对环境有害的物质，容器可能会泄漏。	
1.2 产品安全（危害）	如果软件用户犯了错误，是否存在产品安全的潜在风险？ 是的，在所有情况下，如果用户输入了不正确的校准数据（参见 1.1） - 患者伤害 - 是的。 - 操作员伤害 - 是的。 - 旁观者伤害 - 是的。 - 服务人员伤害 - 是的。 - 环境危害 - 是的。	见风险 1。
2.1 产品质量	如果软件发生故障，产品质量是否存在潜在风险（安全风险除外）？ 是。 产品可能会超出规格，因为校准后的设备可能被软件误认为校准设备。 虽然错误识别不是安全问题，但可能会引起客户的不满。	见风险 1
2.2 产品质量	如果用户犯了错误，产品质量是否存在潜在的风险（安全风险除外）？ 是。 如果用户为该设备输入了不正确的校准数据，并且该设备用于测量产品，则该产品可能超出规格。 虽然不正确的规格不是安全问题，但可能会引起客户不满。	
3.1 记录完整性	在记录存储系统中记录完整性是否存在潜在风险？ 记录丢失 - 是的。 校准记录可能会丢失。 记录损坏 - 是的。 校准记录可能损坏。	风险 2 - 校准记录丢失并导致合规性问题 风险 3 - 校准记录损坏并导致合规性问题。
4.1 证明符合 ISO 标准	证明遵守法规的能力是否存在潜在风险？ 记录丢失 - 是的。校准数据可能会丢失。 记录损坏 - 是的。 校准数据可能损坏。	见风险 2 和 3。

表 C.17 - 实例 12 - 风险评估和控制

风险识别	描述	严重度	措施	剩余风险
风险 1	不校准设备用于测量产品或用于测量压力或力量。（由于软件错误标识了设备或用户输入了不正确的设备校准数据，因此发生了风险。）	严重	系统设计用于打印包含设备 ID 号，序列号，校准状态和截止日期的标签。 在程序上，员工接受培训，在使用设备之前验证这些信息。 另一个过程要求输入的数据在被提交到校准记录之前由第二个人验证。	可接受
风险 2	记录丢失，校准管理活动无法得到保护。	一般	所有校准数据都保存在校准室的纸质记录中。	可接受
风险 3	记录被破坏，校准管理活动无法得到保护。	一般	记录被破坏，校准管理活动无法得到保护。	可接受

在完成风险分析后，XYZ 经理们认为缓解后的剩余风险是可以接受的。

完成验证计划

为完成验证计划，管理人员修改计划以包含以下选择的工具。

- 实施工具：
  - 可追溯性矩阵;
  - 审查系统配置。
- 测试工具：
  - 尺寸分析;
  - 测试计划;
  - 供应商提供的测试套件，以及针对计划配置进行的额外测试以及从以前版本的软件获得的新功能。
- 部署工具
  - 内部应用培训;
  - 安装资格（针对服务器和工作站）

规划维护

除了计划验证系统外，XYZ 经理还决定维护系统的计划将会有所帮助，因为在某些时候肯定需要维护。

将采用系统监测技术来审查所有缺陷，使用问题和对预期用途的改变。

将制定计划来对系统变化（即硬件，升级，补丁，安全问题）进行分类，以便更有效地实施变更。

例 13：自动分拣系统

加里公司的工程师非常擅长工作。 他们知道 Gary 的自动化领域生产的产品，金属棒的长度从 1.27 厘米到 3.81 厘米不等，因此他们可以找到两种应用：一种使用 2.54 厘米或更小的棒，另一种使用棒 3.18 厘米（正负 0.64 厘米）。 所有的棒料都是 0.32 厘米宽。 这两个应用程序都是用于医疗设备的，并且要求棒料具有特定的长度。 Gary 是自动化领域的工程师，负责验证对零件进行分类的自动化视觉系统。 该系统正在取代手动测量/分拣过程。 该流程没有其他更改，因此这是验证的全部范围。

过程描述

棒材厚度的规格对于两种应用都是相同的，并且该尺寸在条形切割机器使用的原材料中得到确认。所有验收标准都在上游得到确认，除了由 Gary 的自动视觉系统测量的钢筋长度。

该机器的过程很简单。棒料被装入一个箱子，使用漏斗将棒料放在传送带上，一次一个。每个棒料被传送到一个停止，在那里一个摄像头看着棒料并测量长度。根据不同的结果，棒料会被传送到两个仓中：一个用于 2.54 厘米或更小的，另一个用于更长的棒料。

在下游，没有对棒长度的额外检查。如果使用尺寸不合适的棒，则会增加患者受伤的风险，因为错误尺寸的棒可能导致正在制造的设备发生泄漏。没有设计方法来测试下游风险的增加。如果在指定尺寸范围内条形码的长度正确，则设备中不会出现泄漏。这些设备已经制造了多年，风险已被很好地理解。自动化视觉系统正在取代手动测量过程。

定义预期用途

因为他理解这个过程是自动化的，所以 Gary 从定义目的和意图开始。

- 该软件旨在确认传送带上每个单根金属棒并测量其长度。

风险分析

Gary 使用本地风险分析过程来确定系统故障风险很高，因为除了通过产品故障或破坏性测试外，没有办法检测何时使用错误尺寸的钢筋。失败可能会导致患者受伤。该过程的关键参数是条的精确长度尺寸。自动化既不增加也不减少这种风险。

验证计划

在他的验证计划的第一次迭代中，Gary 计划使用严格的验证过程（他的风险分析的高风险评级的结果）。在审查了潜在验证工具的工具箱之后，他计划了一份正式的需求定义文档并安排了一份软件需求审核。这次审查将包括制造工程师，另一位自动化工程师和质量工程师。

该系统的软件将在内部开发，但在以往系统自动化的基础上，开发将相对简单。

风险控制措施

确定了两个重点风险领域。

- 确认需要一个棒来衡量。 这台机器将这些钢筋传送到狭窄的道路上，宽度为 0.64 厘米，高度为 0.48 厘米。 因此，如果一根杆位于另一根杆上方，则杆只能沿纵向进入，并且不会进入，因为开孔不够大。 但是，传送带中可能有两个部分相邻。
  为了减轻这种风险，软件会在检查长度之前检查每个栏的宽度。 如果一根钢筋宽度大于 0.32 厘米（± 0.08 厘米，根据先前检查的规范），钢筋将因为输送带中有两根钢筋而被拒绝。 为此目的，机器设计中添加了第三个仓（废品仓）。
- 棒料可能太靠近，以告知一个棒料结束，下一个棒料开始。 该软件会将任何无法确认为等于或小于 3,81 厘米的钢筋输送到废钢箱中。

验证任务

接下来，Gary 转向验证任务。他确定了对正式设计文件的需求，并规划了与审查需求的相同团队成员一起对设计的每个部分进行正式检查。另外，一旦生成代码，将根据另一

位在软件开发方面有丰富经验的自动化工程师和制造工程师的设计对其进行审核。由于该软件正在内部开发，因此未选择供应商管理活动。自动化工程师，制造工程师和质量工程师都将被要求检查软件和设计的可追溯性以回到要求。他们将在测试后执行相同的练习，以确保所有要求都经过完全测试。

Gary 在工具箱测试部分的选择包括测试计划，测试计划包括软件环境的细节和预期的测试结果。他计划在开发的各个阶段进行多种类型的测试，包括单元测试，集成测试和系统测试。将使用正常和错误测试案例，以及与传送带速度有关的性能测试。除了 Gary 之外，测试计划还需要由其他自动化工程师，制造工程师和质量工程师审查和批准。测试报告包括实际测试结果与预期结果的比较，合格/不合格指示，测试识别和问题解决记录以及任何故障的回归测试。对于测试报告，加里需要同一组的批准

实施，测试和部署

为了部署自动化视觉系统，Gary 审查了工具箱中的部署工具，并决定需要安装认证。另外，他确定应该创建用户程序，并且该系统的用户需要该操作员认证。

保养

加里的部门共同计划在制造车间的所有系统进行维护。 这方面不需要特别的计划或行动。

示例 14：拾取和放置系统

Hi-Quality Medical Corporation 是一家 B 级医疗设备制造商。 Hi-Quality 希望将从一个工作站部分完成的部件放置到由公司制造的医疗设备的一部分中。

根据 ISO 13485，新的 Pick and Place（P 和 P）系统的项目经理 Jill 确定 P 和 P 过程是医疗设备质量系统过程，因为它是医疗制造的一部分 设备。 因此，建议的 P 和 P 系统将落入软件验证的要求之下。

定义当前进程

为了更好地理解开发 P 和 P 系统所涉及的要求和风险，Jill 定义了如下关联的业务流程。

a）制造过程中来自工位 11 的部件被放入工位 12 的料筒中（每个料筒 20 份）。 目前，该操作由操作员手动执行。

b）操作员然后手动将墨盒放置到工作站 12 的进入轨道上。

c）操作员手动检查墨盒以确认部件的正确放置。 [步骤 b）和 c）每盒完成约 3 分钟。]

d）墨盒继续进行其他组装步骤，其中包括目视检查，确认过程中以前的所有步骤都没有变形。

分析流程风险

吉尔接下来考虑当前过程中会出现什么问题。 她的分析表明可能会发生以下事件。

a）操作员可能会使部分完成的零件变形。 检测站将在下游检测到畸形。

b）操作员可能会错误地将部件放入墨盒或可能错过墨盒中的插槽。 手动检测期间，在工位 12 处当前检测到插槽错误放置或缺失。

鉴于这些风险控制措施，Jill 确定剩余过程风险较低。 因此，她期望新的 P 和 P 系统也将是一个低风险系统。

定义新的过程

在评估过程风险并使用她对 P 和 P 系统的理解之后，Jill 将新过程定义如下。

a）P 和 P 系统将装入墨盒。

b）P 和 P 系统将从工位 11 中选取零件并将它们插入墨盒（每个墨盒 20 个零件）。

c)P 和 P 系统将视觉检查墨盒以确保所有部件均正确放置并且墨盒中的所有插槽均已充满。任何不正确的墨盒将自动被拒绝。

d）P 和 P 系统将可接受的墨盒放置到工位 12 上。[步骤 b）至 d）现在需要 1 分钟]

e）墨盒将继续进行其他组装步骤，其中包括目视检查，以确认过程中所有先前步骤中缺乏变形。

定义软性用途

吉尔现在了解这个过程是自动化的，并准备为拟议的新 P 和 P 系统写下目的和意图声明。

- P 和 P 系统将拾取来自第 11 站的零件，并将它们放入墨盒。 它将确认所有墨盒插槽已正确填充，将拒绝任何不正确的墨盒，然后将墨盒以每分钟一个墨盒的速率移动到工作站 12 的输入线上。

Jill 然后考虑 P 和 P 系统是否将与其他系统连接。 她总结说，没有其他的互动。 她确定有用户界面但没有软件界面。

验证计划

在分析业务流程自动化并确定新系统的目的和意图之后，Jill 已准备好在高层次上制定验证计划。她稍后需要添加更多细节，但现在开始进行验证规划，她将能够确定所需验证工作的级别。

此前，Jill 确定，对于当前的流程，存在较低的剩余流程风险。她因此认为在验证工作中必须有一点细节或形式。 Jill 知道为新系统定义用户业务流程要求和软件要求非常重要。不过，她指出，这是一个低风险的系统，并不认为单独的文件需要在每份文件上单独签字。

因此，Jill 决定将用户业务流程要求，软件要求和测试计划组合成一个文档。

由于新系统的风险很低，因此 Jill 决定对验证工作进行广泛的管理审查是不必要的，制造商和质量保证代表的批准是足够的。但是，为确保用户要求正确，她在此过程中增加了代表操作员的评论。

Jill 使用 Hi-Quality 的标准格式验证计划开始验证计划草案。验证计划的某些部分仍然是空的;在初始系统设计被批准后，Jill 将完成空白部分。

定义系统和软件要求

然后 Jill 转向系统和软件要求。她决定软件需求将包括 P 和 P 过程或系统步骤以及 P 和 P 系统如何与站 11 和 12 连接的接口规范。系统要求包括 P 和 P 的速度和准确度系统运动。为了降低伤害风险，Jill 增加了一项安全要求，以在操作员与 P 和 P 手臂之间提供物理屏障。

建立信心并控制软件

吉尔现在应该决定采用何种方法和技术来购买新系统。鉴于业务需求非常简单，交易量将会很低。由于新系统风险低，Jill 决定追逐第三方 P 和 P 系统。出于价格和质量的原因，她决定从 P 和 P 系统的行业领导者 Controlsys Inc.购买 P 和 P 系统。

Controlsys 是一家外部系统供应商。因此，Jill 现在应该决定她应该执行哪些类型的活动来建立对 Controlsys 的信心。她评估她对 Controlsys 的信息。 Jill 指出，Controlsys 拥有广泛使用的 P 和 P 产品，并拥有强大的记录。过去，该产品的问题和问题已在互联网留言板上迅速确定并公布。对这些信息的审查表明，这只是少数未知的问题，Jill 确认这些问题与她对软件的预期使用无关。此外，Controlsys 还提供自动化安装认证/操作认证/性能认证（IQ / OQ / PQ）测试套件。鉴于公司的历史以及 P 和 P 系统是低风险系统的事实，Jill 决定她不需要为 Controlsys 执行现场供应商审核。她批准了作为供应商的 Controlsys。

分析软件故障风险

Jill 已经确定自动化业务流程的风险很低，但她仍然需要分析软件故障的风险。 她决定使用定量风险模型并按如下方式对新系统进行排名。

- Jill 在 1 至 10 的范围内将“严重性”评为低（3），因为下游活动会检测到软件故障。
- 她将“可能性”评为低（1），因为系统设计非常简单，使得在测试过程中不会发现所有重要错误的可能性较小。
- 她计算的风险评分为 4，这意味着低风险分类。

因此，Jill 决定执行适合低风险的验证任务。

完成验证计划

jill 现在已经定义了软件需求，选择了实施方法并分析了软件风险。 因此，她有足够的信息来完成验证计划。

由于所提议的系统具有较低的剩余风险，因此 Jill 为剩余的开发和验证工作选择以下工具。

- 设计，开发和配置工具：
  - 软件架构文档和审查;
  - 可追溯矩阵（集成到需求规格中）;
  - 风险控制措施将记录在用户规范中。
- 测试工具：
  - 集成测试（在需求说明书中记录）;
  - 接口测试（在需求规格中记录）;
  - 软件系统测试（在需求说明书中记录）。
- 部署工具：
  - 用户程序审查;

- 应用程序的内部培训;
- 供应商提供的测试套件（来自 Controlsys）。

规划维护

Jill 现在考虑一旦系统部署后哪些活动适合确保系统质量。 由于该系统的残留风险很低，因此在将校准运动机制添加到校准计划时，她遵循制造商的建议。 吉尔将该系统置于该公司用于验证审核的最长周期（3 年）。

批判性思维评论

最后，Jill 问她自己是否考虑了确保她对验证方法有正确信心的所有要素。 她得出结论认为，所选择的和已完成的验证活动提供了可以接受的信心水平，即软件将按照预期执行。

参考书目

[1] ISO 9000，质量管理体系 - 基础知识和词汇
[2] ISO 12207，系统和软件工程 - 软件生命周期过程
[3] ISO 13485：2016，医疗设备 - 质量管理体系 - 监管目的的要求
[4] ISO 14971：2007，医疗器械 - 医疗器械风险管理的应用
[5] 1S0 / IEC 指南 51，安全方面 - 将其纳入标准的指南
[6]！EC 62304：2006 / AMDl：2015，医疗设备软件 - 软件生命周期过程 - 修正 1
[7] IEC / TR 80002-1，医疗设备软件 - 第 1 部分：将 ISO 14971 应用于医疗设备软件的指南
[8]国家标准与技术研究院（NIST）特别出版物 500-234，软件验证和验证过程参考信息，Dolores R. Wallace，Laura M. Ippolito，Barbara Cuthill，1996 年 3 月 19 日
[9]软件工程研究所。能力成熟度模型集成（CMMI）
[10] PRESSMAN R.软件工程，从业者的方法。 McGraw-Hill，Inc，第三版，1992
[11]医疗器械分类原则，GHTF / SG1 / N77，2012