第5章 软件质量保证

以扔掉被检验出有缺陷的东西为目的的检验已经太迟了，没有效率并且成本很高。质量不是来自于检验而是来源于过程的改进。

——戴明（W. Edwards Deming）

CMMI在第一级的改进方向中就提出“开展软件质量保证（SQA）活动”，可见SQA工作在软件开发流程中具有非常重要的作用。而所有的SQA活动都离不开组织中的人，SQA组织的好坏在一定程度上也决定了SQA活动被执行的情况。下面是被广泛使用的SQA活动流程。

（1）建立SQA组织。
（2）选择SQA任务。
（3）建立/维护SQA计划。
（4）执行SQA计划。
（5）建立/维护SQA流程。
（6）定义SQA培训。
（7）选择SQA工具。
（8）改进项目的SQA流程。

5.1 软件质量保证体系

软件质量保证和一般的质量保证活动一样，是在软件生存期所有阶段确保软件产品质量的活动。软件质量保证的目的是把对软件开发过程及其相关工作产品的客观洞察结果提供给项目的开发人员和管理人员。SQA是为了确定、达到和维护需要的软件质量而进行的所有有计划、有系统的管理，主要包括以下功能。

- 制订和展开质量方针。
- 制订质量保证方针和质量保证标准。
- 建立和管理质量保证体系。
- 明确各阶段的质量保证任务。
- 坚持各阶段的质量评审、整理面向用户的文档与说明书等。
- 收集、分析和整理质量信息。
- 提出和分析重要的质量问题。
- 总结实现阶段的质量保证活动。

软件质量保证的主要作用是通过开发过程的可见性给管理者提供实现软件过程的保证，因此 SQA 组织要保证如下内容的实现。

- 选定的开发方法被采用。
- 选定的标准和规程得到采用和遵循。
- 进行独立的审查。
- 偏离标准和规程的问题得到及时的反映和处理。
- 项目定义的每个软件任务得到实际的执行。

相应地，软件质量保证的主要任务有 SQA 审计与评审、SQA 报告、处理不符合问题和实施。

1. SQA 审计与评审

SQA 审计包括对软件工作产品、软件工具和设备的审计，评价这几项内容是否符合组织规定的标准。SQA 评审的主要任务是保证软件工程组的活动与预定义的软件过程一致，确保软件过程在软件产品的生产中得到遵循。

2. SQA 报告

SQA 人员应记录工作的结果，并写入报告中，发布给相关人员。SQA 报告的发布应遵循 3 条基本原则：SQA 和高级管理者之间应有直接沟通的渠道；SQA 报告必须发布给相关的软件项目管理人员；在可能的情况下，向关心软件质量的人发布 SQA 报告。

3. 处理不符合问题

这是 SQA 的一个重要的任务，SQA 人员要对工作过程中发现的不符合问题进行处理，及时向有关人员及高级管理者反映。在处理问题的过程中要遵循以下两个原则。

(1) 对符合标准过程的活动，SQA 人员应该积极地报告活动的进展情况以及这些活动在符合标准方面的效果。

(2) 对不符合标准过程的活动，SQA 要报告其不符合性以及它对产品的影响，同时提出改进建议。

4. SQA 任务实施

软件质量保证任务的实现需要考虑以下几方面的问题。

(1) SQA 人员应具有良好的素质、专业技术能力和丰富的经验，保证胜任 SQA 工作以及 SQA 任务的有效执行。

(2) 组织应当建立文档化的开发标准和规程，使 SQA 人员在工作时有一个判断的标准。如果没有这些标准，SQA 人员就无法准确地判断开发活动中的问题，容易引发不必要的争论。

(3) 高级管理者必须重视软件质量保证活动。如果高级管理者不重视，SQA 人员发现的问题不能及时处理，软件质量保证可能会流于形式，很难发挥其应有的作用。

(4) SQA 人员在工作过程中一定要抓住问题的重点与本质，不要陷入对细节的争论之中。SQA 人员应集中审查定义的软件过程是否得到了实现，及时纠正那些疏漏或执行不完全的步骤，以此来保证软件产品的质量。

(5) 做好软件质量保证工作还应有一个计划，用以规定软件质量保证活动的目标，执行审查所参照的标准和处理的方式。对于一般性项目，可采用通用的软件质量保证计划；对于那些有着特殊质量要求的项目，则必须根据项目自身的特点制订专门的计划。

5.2 软件质量保证的组织

在软件质量保证过程中,质量保证组织起着至关重要的作用。因为只有在一个得到良好规划和管理的流程下,才能生产出高质量的产品。

5.2.1 软件质量组织

在SQA组织建立之前,首先要考虑的问题是质量对于企业的重要性,例如:

- 质量的重要性超过了按时发布一个关键的产品?
- 产品中包含多少个Bug就不能发布?1个?10个?100个或者更多?

当意识到软件质量对于企业已经如此重要时,SQA组织的创建也就顺理成章了。在企业中,基本的质量保证组织主要有软件测试部门和SQA小组(部门)。

1. 软件测试部门

软件测试是对开发出的半成品或成品进行测试,找出软件中存在的缺陷。这里所说的软件缺陷并不仅仅是指功能上的缺陷,任何不符合客户需求的地方都可以认为是缺陷。测试小组成员对其本身的技能又有一定的要求,因为软件测试更多的是通过各种测试技术来发现软件中的问题,如白盒测试、压力测试、性能测试等。关于专业软件测试技术在这里不做详细介绍,感兴趣的朋友可以参考《软件测试方法和技术》一书。

随着软件企业的不断发展、成熟,软件测试小组在现代企业中的地位越来越重要,但不同的组织,测试部门的独立性是不一样的,甚至不存在独立的测试部门。至于测试小组的组织形式,则主要有两种:一种是人才库模式,一种是项目模式。

- 人才库形式是指几乎所有的测试人员都处于公司的人才库中,当有项目时,从人才库中选取适合的人员参与该项目。这种形式最大的优点是资源统一分配,不会造成浪费;缺点是对于测试人员的要求高,因为项目千差万别,不同测试人员对项目的熟悉程度也有所不同。
- 项目形式则是将测试人员分配到相应的项目组中,始终从事该项目的测试。这种形式的测试小组对项目非常熟悉,测试效率较高,但因为项目大小、优先级别不同等可能造成人员分配不均,该组织形式较适合项目相对比较稳定的公司。

2. 软件质量保证组织

人们常说"我们都是人,人总是会犯错误的",软件开发人员也不例外,再好的工程师也难保不出错。对开发流程进行监察和控制并保证产品的高质量正是软件质量保证(SQA)组织的重要职能。IBM公司曾说"在超过8年的时间内,SQA组织发挥了至关重要的作用,并使得产品质量得到不断提高。越来越多的项目经理也感觉到由于SQA组织的介入,不管是产品质量还是成本节约都得到较大改善。"那么,应该如何建立SQA组织呢?其实,根据企业的规模和过程能力成熟度的不同可以创建不同形式的SQA组织,但不管采用何种形式,都必须设置独立的SQA工程师,因为只有在职能/行政上独立于受监督人员(项目组),才能保障自身的独立性和评价的客观性。实际上,在许多的大型软件企业中,不仅有独立的SQA工程师,还设有独立的SQA组织,其主要责任就是跟踪和管理软件开发流程的

执行。需要指出的是，很多企业存在一个误区，认为测试就是 SQA，其实测试只是 SQA 中的一个环节，SQA 部门不等于测试部门。SQA 部门需要确保以下工作的正常进行。

(1) 项目按照标准和流程进行。

(2) 创建各种标准文档，以便为后期维护提供帮助。

(3) 文档是在开发过程中被创建的，而不是事后补上的。

(4) 建立变更控制机制，任何更改都需要遵循该机制完成。

(5) 准备好 SQA 计划和软件开发计划。

3. SEPG

软件工程过程组(Software Engineering Process Group，SEPG)通常由软件工程专家组成，在软件开发组织中领导和协调过程改进的小组。他们的主要任务是推动企业所应用的过程的定义、维护和改进。与 SQA 相比，SEPG 类似于一个立法机构，而 SQA 则类似于一个监督机构。

4. 一些虚拟的社区组织

SPIN(Software Process Improvement Network)是软件企业自发组织的地区性机构，为联合各地区对软件过程改进有兴趣的软件专业人员组成的非营利性组织。在国内，仅在少数几个地区有区域性的 SPIN，如北京 SPIN、上海 SPIN、深圳 SPIN、香港 SPIN。

以黑带团队为基础的六西格玛组织(Organization of Six Sigma，OFSS)是领导职能推进六西格玛方法的基础。它的重点在于建立和应用一些展开计划、报告系统和实施过程支持 PFSS(六西格玛过程)和 DFSS(策划)。图 5-1 为六西格玛组织结构图。

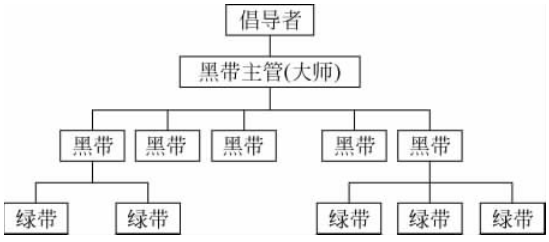

图 5-1　六西格玛组织结构图

5.2.2　软件质量组织结构

SQA 组织模型的选择非常重要，因为组织结构的确定也就决定了人员的分配、角色的职能定义等。在 SQA 发展的初期，通常没有专职的 SQA 人员，几乎所有的 SQA 人员都是由开发人员或其他人员兼任。在这种模式下，SQA 人员的工作职能也非常有限，往往只是从事一些相对比较简单的文档审核等初级的 SQA 任务。随着 SQA 的发展，专职 SQA 人员成为必然的需求。这时，遇到的第一个问题是：如何创建一个有效的 SQA 组织？

常用的 SQA 组织模型主要分为 3 种：独立的 SQA 部门、独立的 SQA 小组和独立的 SQA 工程师。

1. 独立的 SQA 部门

独立的 SQA 部门，顾名思义是在整个企业的组织结构中设立一个独立的职能/行政部门——SQA 部门，该部门和其他职能部门平级，因此这种组织结构模型又称为职能型组织

结构,如图5-2所示。

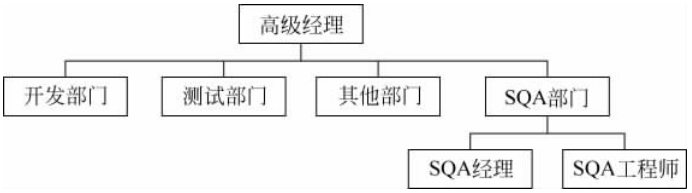

图5-2 SQA组织结构图——职能型结构

在这种结构中,所有的SQA工程师都隶属于SQA部门。在行政上,SQA工程师向SQA经理汇报,在工作上,SQA工程师向项目经理汇报。这种组织结构的优缺点如下。

1)优点

- 保证SQA工程师的独立性和客观性。SQA工程师在行政上隶属于独立的职能部门,因此在流程监控和审查中,更有利于工程师做出独立自主、客观的判断和汇报。
- 有利于资源的共享。由于SQA部门的相对独立,SQA资源被所有项目共享,SQA经理可以根据项目对SQA资源进行统筹分配,这样既避免了资源的相互冲突,又有利于资源的充分应用。

2)缺点

- SQA工程师对流程的跟踪和控制难于深入,往往流于形式,难于发现流程中存在的关键问题。
- 由于和项目组相互独立,SQA工程师发现的问题不能得到及时有效的解决。

2. 独立的SQA工程师

这种组织结构模式又被称为项目型结构。因为在这种模式中,以项目为主体进行运作,在每个项目中都设立有专门的SQA岗位,如图5-3所示。

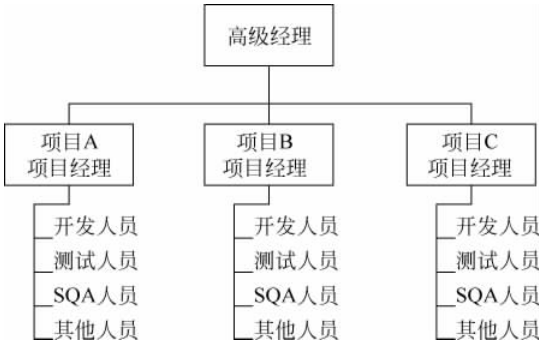

图5-3 SQA组合结构——项目型结构

在这种组织结构中,SQA工程师属于项目成员,向项目经理汇报。该结构的优缺点如下。

1)优点

- SQA工程师能够深入项目,较容易发现实质性问题。
- 对于SQA工程师发现的问题,能够得到较快的解决。

2）缺点

- 项目之间相互独立，SQA 工程师之间的沟通和交流有所缺乏，不利于经验的共享和 SQA 整体的培养和发展。
- 独立性和客观性不足。因为 SQA 工程师隶属于项目组，独立性和客观性有所欠缺。

3. 由独立的 SQA 工程师构成 SQA 小组

该组织结构是上述两种组织结构的综合结果，如图 5-4 所示。

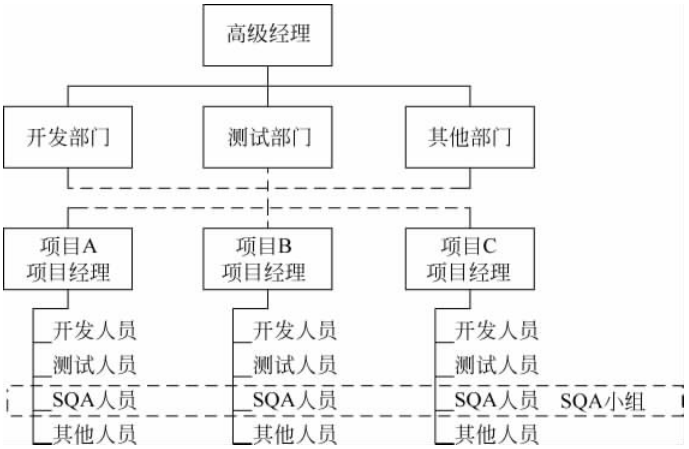

图 5-4　SQA 组织结构图——综合型结构

从职能/行政结构上说，创建了独立的 SQA 小组。SQA 小组虽然不算一个行政部门，但具有相对的独立性。同时，SQA 工程师又隶属于不同的项目组，在工作上向项目经理汇报。该结构综合了上面两种的结构的优点，既便于 SQA 融入项目组，又便于部门之间经验的分享，还利于 SQA 能力的提高。

5.2.3　角色的分类和职能

一旦选定了组织结构，接下来的任务便是确定组织中不同的角色和职能。5.2.2 节中提到的 3 种组织结构都具有一个共同点——专职的 SQA 人员。但需要说明的是，SQA 角色不一定要有专职人员担任，也可以由非全职的项目经理、开发工程师和测试工程师担任，即把专职的 SQA 经理和 SQA 工程师的责任分解给项目经理、开发工程师和测试工程师等角色上。这里侧重讨论专职的 SQA 人员角色与责任。

SQA 是整个企业、整个组织的责任，而不仅仅是某个部门或某几个人的责任。在实际工作中，专职的 SQA 人员承担了大部分的 SQA 任务，对质量保证目标的实现起着非常重要的作用。专职的 SQA 人员主要分为 SQA 经理和 SQA 工程师。

1. SQA 经理

SQA 经理对项目的全部质量保证活动负责，保证 SQA 正常、有序地工作。他们的主要职能如下。

- 制订 SQA 策略和发展计划。
- 管理 SQA 资源。
- 审定项目的 SQA 计划。

- 参加项目的 SQA 工作。
- 评审 SQA 工作状态。
- 提交跨项目的 SQA 计划。

2. SQA 工程师

SQA 工程师的主要职能如下。

- 按 SQA 计划检查指定的产品。
- 执行 SQA 评审/审核。
- 记录各种数据和观察情况。
- 提交不符合报告并处理不符合问题。
- 完成 SQA 计划规定的 SQA 测量和度量。
- 向 SQA 经理报告工作情况。

3. 各角色之间的关系

1) SQA 工程师与项目经理

SQA 工程师与项目经理之间是合作的关系,帮助项目经理了解项目过程的执行情况,如过程的质量、产品的质量、产品的完成情况等。但 SQA 工程师和项目经理的关注点不同:SQA 工程师关注过程和产品的质量;项目经理关注的是项目的目标、任务、进度和风险等。

2) 开发人员与 SQA

开发人员往往对 SQA 人员产生抵触情绪,认为 SQA 工程师本身不写代码,却总是对自己写的东西"指手画脚"。这种抵触情绪不仅会造成 SQA 人员与开发人员之间的对立,而且会影响产品的质量。SQA 工程师和开发人员之间应该是相辅相成的,SQA 工程师虽然不承担具体的开发工作,却要对整个开发过程进行监督和控制保证产品质量。实际上,质量保证并不只是 SQA 工程师的责任,所有的人(包括开发人员)都对产品质量负有责任。SQA 工程师和开发人员的关系在软件开发过程中也非常关键,SQA 工程师和开发人员应该保持良好的沟通和合作,任何对立和挑衅都可能导致质量保证这个大目标失败。

3) SQA 工程师与测试人员

SQA 工程师和测试人员都充当着第三方检查人员的角色。但是 SQA 人员主要对流程进行监督和控制,保证软件开发遵循已定的流程和规范。而测试人员则是针对产品本身进行测试,发现他的缺陷并通知开发人员进行修改。

4) SEPG 与 SQA

SEPG 的职责是制订过程、实施过程以及改进过程,而 SQA 的主要责任则是保证流程被正确地执行。SEPG 人员提供过程上的指导,帮助项目组制订项目过程和进行策划,从而帮助项目组更有效地工作。如果项目和 SQA 人员对过程的理解发生争执,SEPG 人员应当作为最终仲裁者。

总的来说,SQA 人员组织结构中存在不同的角色,不同的角色又有不同的分工,虽然具体工作不同,却有着同一个目标——生产符合条件的、高质量的产品! SQA 工程师需要努力协调各方面的关系,共同完成质量保证的大目标。

5.2.4 SQA 人员的要求和培养

1. SQA 人员的要求

SQA 人员应该具备怎么样的素质？SQA 人员如何规划和发展自己的职业生涯？作为 SQA 组织的重要组成部分，SQA 人员的要求和培养至关重要。下面列出了对一个优秀质量人员的基本要求。

1）扎实的技术基础和背景

质量保证工程师通常要求计算机科学的学术背景，以及扎实的软件开发经验。这些经验将帮助工程师很好地理解 SQA 人员的职责和责任，并在实际工作中起着重要作用。一个优秀的质量工程师至少需要 3～5 年的软件开发经验。

2）良好的沟通能力

这一点对于软件质量工程师也非常重要。有时候，SQA 工程师和开发工程师的观点相对立。开发工程师总是愿意保护他们所开发的东西，而 SQA 人员则需要坚持自己的观点，甚至向高层经理汇报。这时，如果 SQA 工程师具有良好的沟通能力则能够很好地缓和这种对立面，更有利于工作的完成。

3）敏锐性和客观性

SQA 工程师必须保持敏锐的洞察力和客观性。能够准确地发现软件产品和过程的质量问题。

4）积极的工作态度

这一点对于任何工作都是十分重要的。如果一个质量人员出现消极的工作态度，那么就无法在日常工作中认真地对待工作，以及无法对质量问题保持谨慎的态度，从而很可能对产品的质量问题睁一只眼闭一只眼。

5）独立工作的能力

SQA 人员必须具备独立工作的能力。因为当项目需要时，专职的 SQA 人员必须能够独立深入参与项目/产品开发，并保证项目/产品的质量达到预定的标准。

2. SQA 人员的培养

从总体上来说，对 SQA 人员的培养分为两个部分，技术培养和素质培养。所谓技术培养是提高 SQA 工程师的专业技能，而素质培养则是提升工程师的个人素质，以便更出色地完成相应任务。

1）技术培养

技术培养可以分为两个大的范畴：基础软件知识的培养和 SQA 专业知识的培养。基础软件知识的培养包括软件开发工具、开发语言和版本控制等，这些知识通常来源于工程师之前的工作经历。SQA 专业知识培养主要涵盖如下内容。

- 软件工程。
- 软件质量规范和标准。
- 软件质量模型。
- 软件质量控制。
- 软件配置管理和度量。

2) 素质培养

要成为一个优秀的质量保证工程师,不仅要具有良好的技能,还需要具备良好的个人素质,包括良好的沟通技巧、耐心、独立性以及逻辑性等各种能力。这主要依靠工程师在平时的工作和生活中自我提高。

5.2.5 六西格玛的角色和人员培训

1. 六西格玛的角色及其职责

六西格玛是一项以数据为基础,追求近乎完美的质量管理方法。六西格玛中涉及的人员包括绿带、黑带、黑带大师和倡导者等。凡实施六西格玛的企业都必须训练出一支属于自己的黑带、绿带队伍。六西格玛组织架构由组织的倡导者、大黑带、黑带、绿带和项目团队等构成,同时它需要组织的最高管理团队、项目的保证人以及过程的所有人给予充分地支持、沟通与协调。

1) 倡导者

由经过大量六西格玛培训的高级管理人员组成,是推动六西格玛的最高负责人。倡导者为顺利推动六西格玛提供必要的资源和支持,也是项目批准和审核的最终决策者。他们负有的职责如下。

(1) 负责六西格玛管理在组织中的部署。

(2) 构建六西格玛管理基础。例如,部署人员培训,制订六西格玛项目选择标准并批准项目,建立报告系统,提供实施资源管理等。

(3) 负责六西格玛管理实施中的沟通与协调。

2) 黑带大师

通过特别培训的质量技术专家,负责推动质量团队建设和加速过程改进。黑带大师挑选、培训和指导黑带,完善和执行六西格玛实施方案,并确保完成挑选的六西格玛项目。他们负有的职责如下。

(1) 对六西格玛管理概念和技术方法具有较深的了解和体验,并将他们传递到组织中。

(2) 对培训黑带和绿带的六西格玛项目提供指导。

(3) 协调和指导跨职能的六西格玛项目。

(4) 协调倡导者和管理层选择和管理六西格玛项目。

3) 黑带

全职的六西格玛项目领导,需要经过4个月的培训,同时在黑带大师的指导下自主完成两个六西格玛项目,并最终取得认证。他们所负有的职责如下。

(1) 领导六西格玛项目团队实施并完成六西格玛项目。

(2) 向团队成员提供适用的工具和方法的培训。

(3) 识别过程改进机会,并选择最有效的工具和技术实现改进。

(4) 向团队传达六西格玛管理理念,建立对六西格玛管理的共识。

(5) 向倡导者和管理层报告六西格玛项目的进展。

(6) 为绿带提供项目指导。

4) 绿带

半专职的六西格玛项目组成员。是组织中经过六西格玛管理方法与工具培训的,结合

自己的本职工作完成六西格玛项目的人员。一般，他们是黑带领导的项目团队的成员，或结合自己的工作开展涉及范围较小的六西格玛项目。

2. 六西格码培训

近年来，随着越来越多的企业在实施六西格玛，六西格玛的培训也日益成为大家的焦点。2002 年 9 月 16 日，中国质量协会成立了“中国质量协会六西格玛管理推进工作委员会”，宣传六西格玛管理理念，推广六西格玛管理方法和工具，引导我国企业实施六西格玛战略，不断改进企业的经营绩效，提升我国企业的国际竞争力。在企业中，六西格玛人员的培训主要分为下面 3 个部分。

1) 高层管理的培训

该阶段的培训主要是使高级管理层对六西格玛有正确、清晰的认识。因为六西格玛管理是自上而下的管理模式，在整个六西格玛的实施过程中都需要高层管理的大力支持。因此高层管理能否支持六西格玛，对整个六西格玛项目的成功与否非常重要。

2) 黑带/黑带大师和绿带培训

在六西格玛项目中，真正的执行人员是黑带和绿带。因此，黑带和绿带培训需要学员通过培训掌握六西格玛基本概念、基本工具的使用等。这不仅是理论上的学习，黑带和绿带还需要大量的项目实践过程。培训一个黑带，通常要花费 4 个月左右的时间，培训“黑带大师”则需要花费 2 年左右的时间。

3) 全体培训

要使六西格玛项目获得成功仅仅靠几个黑带、绿带是不够的，还需要所有人都能对六西格玛有一个正确的认识，因此还需要在整个企业内部推行六西格玛文化，这是一个循序渐进的过程。

3. 六西格玛中心的黑带培训

在上面提到的 3 个培训中，以黑带培训最为重要。黑带是六西格玛管理中非常重要的角色，是组织十分宝贵的资源，在六西格玛管理中起着承上启下的关键作用。一般说来，黑带是六西格玛项目的领导者，负责带领六西格玛团队通过完整的 DMAIC 或 DFSS 流程，完成六西格玛项目，达到项目目标并为组织获得相应的收益。

通常的黑带培训在 4 个月内完成，整个培训围绕 DMAIC 展开，即 Define 定义、Measure 测量、Analyze 分析、Improve 改进、Control 控制。培训分 4 个阶段，每个阶段的时间为一个月。在这一个月中，一周用于理论学习，其余 3 周均用于项目实践，然后再进行下一阶段的培训环节。

(1) 第一周　学习定义和测量的所有相关知识、方法和工具。

学会识别客户需求(CTQ)，并找出造成不能满足客户需求的关键业务流程。同时需要将现有流程中的度量数据标准化，以便识别业务流程中的缺陷和造成失误的真正原因。

由于企业业务流程包含的许多大小不同的闭环流程，因此，黑带学员有必要回到工作岗位后，根据所学的知识选择需要改善的特定业务流程项目，并收集所有相关数据。

(2) 第二周　学习分析的所有相关知识、方法和工具。

这个阶段的黑带已经选择好了需要改善的流程项目，并在工作实践中带着问题回来。因此，在第二阶段培训师需要和学员共同探讨解决问题的方法。同时，学员们又要重点学习许多分析的工具和方法，如建立当前流程的能力基准线，定义结果的改善目标，判别造成结

果变动的原因等。通过分析导致项目失败的主要原因,为解决真正的问题制订行动计划和时间表。

(3) 第三周　学习改进的所有相关知识、方法和工具。

黑带学员在此阶段将通过诸如实验设计概述(DOE)、单因素实验与分析、全面因素实验与分析等专业的方法过滤少数造成结果变动的最重要的原因,即关键原因,同时发现关键原因与结果的关系,进而建立关键原因的允许变动范围并改善业务流程能力,实施解决方案。

(4) 第四周　学习控制的所有相关知识、方法和工具。

黑带学员们将利用学到的技术控制工具和流程管理方法校准关键原因的测量系统,决定控制关键原因的能力,最终对改进的业务流程实施系统控制。

认证工作:在培训完成之后,为保证黑带学员掌握并且能够应用六西格玛管理方法来实施项目,一般要求黑带学员完成2～3个项目,并且还需要对他们的项目进行评审和认证,以证明他们成为合格的黑带,能够独立地领导企业内部六西格玛流程改进项目的实施。

5.3　SQA组织的目标和责任

根据软件质量保证体系,SQA质量保证过程如图5-5所示。其中,文档化的过程描述/开发标准/规程/模板等属于组织级SQA,通常由质量组织SEPG(软件工程过程组)预先定义好,供各个项目使用、定制,而项目组SQA保证所定义的流程被正确执行。SQA计划、SQA评审和审核、SQA度量、SQA评估和改进、SQA报告等软件质量保证活动跟随软件开发过程例行开展。

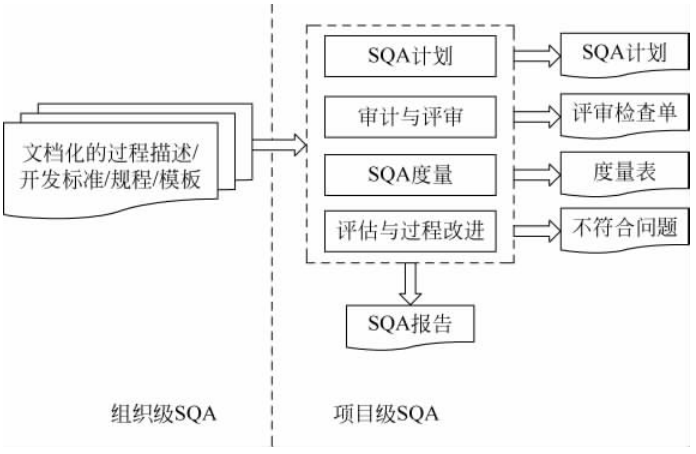

图5-5　SQA质量保证过程

SQA组织并不负责生产高质量的软件产品和制订质量计划,这些是软件开发人员的工作。SQA组织的责任是审计软件经理和软件工程组的质量活动并鉴别活动中出现的偏差。

软件质量保证的目标是以独立审查的方式监控软件生产任务的执行,给开发人员和管

理层提供反映产品质量的信息和数据，辅助软件工程组得到高质量的软件产品。那么，SQA 组织将如何保证软件质量呢？

5.3.1 SQA 计划

SQA 计划实施的步骤如下。

(1) 了解项目的需求，明确项目 SQA 计划的要求和范围。

(2) 选择 SQA 任务。

(3) 估计 SQA 的工作量和资源。

(4) 安排 SQA 任务和日程。

(5) 形成 SQA 计划。

(6) 协商、评审 SQA 计划。

(7) 批准 SQA 计划。

(8) 执行 SQA 计划。

在每个项目开始之前，SQA 人员都需要按照要求完成详细的 SQA 计划。SQA 计划包含如下内容(根据具体情况，有所增减)。

- 目的：SQA 计划的目的和范围。
- 参考文件：SQA 计划参考的文件列表。
- 管理：组织、任务、责任。
- 文档：列出所有相关的文档，如程序员手册、测试计划、配置管理计划等。
- 标准定义：文档标准、逻辑结构标准、代码编写标准、注释标准等。
- 评审/审核。
- 配置管理：配置定义、配置控制、配置评审等。
- 问题报告和处理。
- 工具、技术、方法。
- 代码控制。
- 事故/灾难控制：包括火灾、水灾、紧急情况和病毒等。

下面就 SQA 计划中几项关键内容展开讨论。

1. 管理

在 SQA 计划中，管理主要是指组织结构、任务和责任。

(1) 组织结构是项目的组织结构，包括项目组成员的角色和地位，以及说明项目组中 SQA 人员相对独立的职权。

(2) 任务列出了整个项目需要完成的 SQA 任务，如需求说明审核、设计文档审核等。

(3) 责任指出在软件开发的不同阶段，项目经理、开发小组、测试小组和 SQA 等负有的主要责任。

2. 评审/审核

评审/审核在整个 SQA 工作中，占有十分重要的地位，因此在 SQA 计划中也需要阐述清楚哪些评审需要完成。ANSI(American National Standards Institute)曾建议了如下必不可少的评审内容。

(1) 需求说明评审(requirement specification review)。

(2) 设计文档评审(design document review)。

(3) 测试计划评审(test plan review)。

(4) 功能性审核(functional audit)。

(5) 物理性审核(physical audit)。

(6) 管理评审(management review)。

3. 问题报告和处理

在SQA计划中,需要描述问题报告和处理系统,对该系统的说明确保了所有的软件问题都被记录并解决。所有问题都需要被分析,并归入特定的范畴(如需求、设计、编码等)和响应的级别。

4. 工具、技术、方法

确定需要使用的软件工具、技术和方法,并说明其目的。

5. 代码控制

代码控制包含的内容很多,如防止代码意外丢失,确保代码同步,允许回溯到开发过程中的任一点,允许多个工程师同时修改同一个文件等。

示例:

XYZ项目质量保证计划

质量目标:

XYZ项目需要遵循由QMS提供的并达成协议的质量目标。目标如下:

……

评审:

每周和每月都需要进行项目评审,并按照要求生成相应的状态报告。在项目的实施计划中,需要定义和安排同行评审,同样需要按照相关的模板记录评审情况并生成报告。在软件开发过程中,需要进行的评审包括:

……

测试计划:

该项目的测试计划文档为"XYZ项目测试计划",请参考相应文档(ID 1003)……

5.3.2 评审和审核

自从Michael Fagan的论文《设计与编码的审查过程》发表之后,审查一直被作为一种提高质量和减少花费的重要手段。审查的主要目的就是尽早地发现产品中的问题,减少后期维护成本,因此,评审和审核也成为SQA的主要责任之一。

- 评审(review):在过程进行时,SQA对过程的检查;SQA的角色在于确保执行工程活动时各项计划所规定的过程得到遵循。评审通常通过评审会的方式进行。
- 审核(audit):在软件工作产品生成时,SQA对其进行的检查;SQA的角色在于确保开发工作产品中各项计划所规定的过程得到遵循;审核通常通过对工作产品的审查来执行;

从上面的定义可以看出,评审和审核有不同的侧重点。评审是对工作流程的评审,而审核则主要侧重产品本身。在软件开发过程中,主要的评审或审核如下。

- 软件需求评审(software requirements review)。在软件需求分析阶段结束后必须进行软件需求评审,以确保在软件需求规格说明书中所规定的各项需求的合适性。
- 概要设计评审(preliminary design review)。在软件概要设计结束后必须进行概要设计评审,以评价软件设计说明书中所描述的软件概要设计的总体结构、外部接口、主要部件功能分配、全局数据结构以及各主要部件之间的接口等方面的合适性。
- 详细设计评审(detailed design review)。在软件详细设计阶段结束后必须进行详细设计评审,以确定软件设计说明书中所描述的详细设计在功能、算法和过程描述等方面的合适性。
- 软件验证与确认评审(software verification and validation review)。在制订软件验证与确认计划之后要对它进行评审,以评价软件验证与确认计划中所规定的验证与确认方法的合适性与完整性。
- 功能审核(functional audit)。在软件发布前,要对软件进行功能检查,以确认已经满足在软件需求规格说明书中规定的所有需求。
- 物理审核(physical audit)。在验收软件前,要对软件进行物理检查,以验证程序和文档已经一致并已做好了交付的准备。
- 综合检查(comprehensive audit)。在软件验收时,要允许用户或用户所委托的专家,对所要验收的软件进行设计抽样的综合检查,以验证代码和设计文档的一致性,接口规格说明之间的一致性(硬件和软件),设计实现和功能需求的一致性,功能需求和测试描述的一致性。
- 管理评审(management reviews)。对计划的执行情况定期(或按阶段)进行管理评审,评审必须由独立于被评审单位的机构或授权的第三方主持进行。侧重正确性、可测性等的需求评审、设计评审可归为静态、软件测试。

5.3.3 SQA 报告

SQA 活动的一个重要内容就是报告对软件产品或软件过程评估的结果,并提出改进建议。SQA 人员应记录工作的结果,并写入报告,发布给相关的人员。SQA 报告的发布应遵循以下 3 条基本原则。

(1) SQA 和高级管理者之间应有直接沟通的渠道。

(2) SQA 报告必须发布给软件工程组,但不必发布给项目管理人员。

(3) 在可能的情况下,向关心软件质量的人发布 SQA 报告。

表 5-1 是一个软件评审报告的样例。

表 5-1 SQA 评审报告样例

项目名称			项目标识		
部门/组织名			阶段名称		
主持人			会议地点		
评审类别	定期评审	阶段评审	事件评审	产品评审	
评审性质	管理评审	技术评审	质量保证评审		
评审人					

续表

评审项与结论
项目名称： 评审内容： 评审结果： 存在的问题：

从上述例子可以看出,SQA 报告实际上就是对 SQA 工作的一个总结。作为 SQA 工作的重要输出,需要注意如下几个问题。

1. SQA 报告失去应有的价值

这个问题常常出现在 SQA 体系不太成熟的企业。因为 SQA 流程和技能等的不完善,导致 SQA 工程师不能发挥真正的作用。仅完成缺陷数据的统计,甚至审核一些无关紧要的问题,进行一些无关紧要的争论。在这种情况下,SQA 报告往往会失去其应有的价值。例如,在评审报告中,列出一长串语法、格式错误,却不能发现被评审文档中更深层次的问题。

2. 明确报告原则

在实际工作中,常常会遇到项目经理对 SQA 工程师提出的问题置之不理的情况。因为在项目紧张的情况下,项目经理往往不愿意对文档格式错误、文档全面性不足等问题进行修正。因此 SQA 应该具有基本的报告机制,以便于当问题在项目组内无法解决时,SQA 工程师可以寻找其他的途径。基本的问题报告机制如下。

- 当发现问题时,SQA 首先向项目经理报告。
- 问题无法解决时,SQA 可以向高级经理直接汇报。
- SQA 和公司的高级经理之间应该随时保持联系。
- 在实施过程中,SQA 可以向对质量非常关注的现场负责人或高管人员及时报告。

应该避免的报告机制如下。

- SQA 跨越本地组织进行报告。
- SQA 跨越管理层进行汇报。

5.3.4 SQA 度量

20 多年前,度量还是一个新鲜事物,只有极少数的个人、学校和组织在对此进行研究。而现在,度量已经变成了软件工程的一部分。SQA 度量是记录花费在 SQA 活动上的时间、

人力等数据，并通过大量数据的积累与分析，可以使企业领导对质量管理的重要性有定量的认识，利于质量管理活动的进一步开展。通常，SQA 度量涉及 3 个方面：软件产品评估度量、软件产品质量度量、软件过程评审度量。

1. 软件产品评估度量

该度量需要记录在产品评估阶段所涉及的各项资源，如表 5-2 所示。通过该表格可以看到 SQA 在产品评估阶段分配的资源。

表 5-2　软件产品评估度量样例

软件产品评估	页　　数	评 估 耗 时	报 告 耗 时
软件需求说明	20 页	3 小时	1 小时

2. 软件产品质量度量

该度量需要记录的数据是在软件开发周期中发现的 Bug 的种类和数量。SQA 需要分析这些数据，并保证产品的质量。例如，如果 SQA 发现需求和设计引起的 Bug 占了相当高的比例，则 SQA 需要同开发小组一起对 Bug 的根本原因进行分析，看看为什么这些错误不能在早期被发现？是不是相关审查人员存在问题，或者是由于需求文件太过含糊、不清楚等。

为什么要进行软件产品质量度量呢？因为软件度量是对实际数据进行记录和分析，很大程度上减少了人为判断，从而反映了软件质量的实际情况。而且通过这些分析，可以发现软件开发过程中存在的问题并进行改进，有利于进一步提高软件产品的质量。因此，软件产品质量度量也是 SQA 人员的重要职能之一。

3. 软件过程审核度量

过程审核度量主要是记录 SQA 在过程度量方面消耗的各项资源，如表 5-3 所示。

表 5-3　软件过程审核度量样例

被审核的软件过程	审核准备耗时	评 估 耗 时	报 告 耗 时
错误纠正过程	2 小时	2 小时	1 小时

关于软件产品与过程质量度量的详细内容，可参阅第 8 章“软件质量度量”。

5.3.5　SQA 评估任务

SQA 的评估任务主要是在软件开发前期对项目的软件和硬件资源进行评估，以确保其充分性和适用性。SQA 评估在 SQA 的工作中虽然是必要的，但并不是主要任务。因此，本节中会简要介绍 SQA 评估任务，但不做更详尽的阐述。

1. 软件工具评估

SQA 需要对软件开发和正在使用的，以及计划使用的软件工具进行评估，其目的主要是保证项目组能够采用合适的技术和工具。对于正在使用的工具，从充分性和适用性两个方面对软件工具进行评估。充分性主要是检查该工具是否能提供所需的所有功能；适用性则是指该软件在性能等各方面能否满足软件开发和支持的需求。对于计划使用的工具，则主要是考查其可行性。可行性是评估该软件工具能否在现有的技术和计算机资源上有进一步发展。表 5-4 是工具评估表格的实例。

表 5-4　软件工具评估表格样例

软件工具评估
SQA：_____　　评估时间：_____
被评估的软件工具：
评估使用的方法和标准：
评估结果：
推荐的矫正措施：
实施的矫正措施：

2. 项目设施评估

项目设施评估的内容非常单一,只检查是否为软件开发和支持提供了所需要的设备和空间。通过该评估确保项目组有充足设备和资源进行软件开发工作,也为规划今后软件项目的设备购置、资源扩充、资源共享等提供依据。项目设施评估表如表 5-5 所示。

表 5-5　项目设施评估表样例

项目设施评估
SQA：_____　　评估时间：_____
设施评估(设备,人员等)：
评估使用的方法和标准：
评估结果：
推荐的矫正措施：
实施的矫正措施：

5.4 纠正和预防措施

纠正和预防措施(Corrective And Preventive Action,CAPA)最初由欧盟GMP提出,适用于药业,用于处理药品生产质量保证体系中的不符合问题。由于该CAPA体系的通用性,渐渐被其他行业接受,成为质量管理的一个过程标准。

- 纠正措施指的是为消除已发现的不合格或其他不期望情况的原因所采取的措施。
- 预防措施指的是消除潜在的不合格或其他潜在不期望情况的原因所采取的措施。

5.4.1 纠正性和预防性的过程

纠正和预防措施的目的不是处理或直接修改已经发现的缺陷,而是分析并消除那些缺陷在整个软件部门产生的原因,是一个缺陷预防的过程。

纠正措施是一个常规使用的反馈过程,包括质量不合格信息的收集、非正常原因的识别、改进的习惯行为与流程的建立与吸收、对流程执行的控制与结果的测量等。而预防措施则主要包括潜在质量问题信息的收集、偏离质量标准的识别等。纠正措施是为了防止同样问题的再发生;而预防措施是为了防止潜在的同类或相似问题的发生,如图5-6所示。

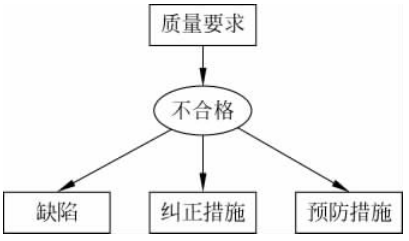

图5-6 纠正和预防措施的概念示例

CAPA的主要信息来自质量记录、报告、内部质量审计、项目风险评审、软件风险管理记录等。CAPA过程重要组成部分如下。

- 收集相关信息;
- 对收集到的信息进行分析;
- 建立解决方案或改进方法;
- 执行新的方案或方法;
- 持续跟踪。

5.4.2 信息收集和分析

为了质量改进性和预防性过程的正常运行,必须建立起与大量产品质量信息相关的文档流,然后进行分析,具体如下。

(1) 筛选信息并找出潜在的改进可能性。评审从各种渠道得来信息或文档,识别出纠正和预防过程的潜在条件,包括与不同单位收到同一类型文档的比较或同一案例不同类型文档的比较。

(2) 对潜在改进进行分析,分析的主要内容如下。

- 由识别出的缺陷产生的损害预期类型和等级定义。
- 缺陷的原因,典型的原因一般是不符合工作条例或规程,技术知识水平不够,对极端时间或预算压力估计不足和对新开发工具缺乏经验。
- 对各种存在于整个组织范围内的潜在缺陷的概率进行估计。

(3) 依照分析结果给出内容上或流程上的反馈。

例如：某软件公司接到市劳动保障局通知，因为国家法规调整，其开发维护的公积金收支软件必须做重大更改，方可继续使用。经公司常务高层会议，准备开发该软件2.0版，增加客户要求新功能的同时，考虑向其他市县销售，必须做到可以定制界面。为了提高该版软件的质量，质量经理和开发部门高层按CAPA过程进行实施。

经过信息收集，市场、开发、质量组织了解到：本市公积金管理较为正规，但是相邻市县差别较大，特别是B市，基本属于个人自己缴纳，只有少数是由单位代缴；相邻市县使用的对手软件功能丰富，但稳定性稍差，基层操作人员抱怨不断；前版软件易用性不佳，表现为有超过3%的操作失误，上岗人员因操作复杂而惧用该软件，需要较复杂的岗前培训。

经过会议讨论和分析，发现易用性不足的主要原因是，开发文档的编写只有质量部门参与，没有市场人员的审阅。会议一致认为，除了增加客户所需功能以外，软件必须在易用性上得到很大改进，其优先级很高。

5.4.3 解决方案及其执行

为了消除同一类型的质量问题一再出现和提高生产效率，需要找到问题的根本原因和解决办法，通常会在以下方向做出努力。

- 更新相关流程，包括开发与维护的规定和其他通用流程。
- 习惯做法的改变，包括相关工作条例的更新。
- 使用新的开发工具，使某些问题不容易发生。
- 培训和再培训或更新人员。
- 更改上报频率或上报任务。

以上方法从一个或几个方面组合，便产生了所需要的方案。对于上一个例子，经过评估，找到了解决方案，包括内部培训计划、市场部招聘具有开发资历的职员计划，制订了部分研发人员与市场部人员互换角色的流程等。为了预防此类问题再次发生，该公司还定期会晤客户，跟踪软件使用的流程，使软件的应用状态得到彻底的改变。

纠正和预防措施解决方案的执行主要在于正确的指导和适度的培训，但更重要的是团队和个人的充分合作，合作是执行CAPA的基础。成功的执行需要选定的人员对提出的解决办法充满信心。

5.4.4 相应措施的跟踪

CAPA过程的跟踪主要有以下几个任务。

- 整理CAPA记录流并跟踪。这些记录流使得反馈能够揭示没有报告以及低质量报告而导致的信息不准确或信息遗漏，而跟踪主要是通过分析长期活动信息而实现。
- 执行的跟踪。主要是CAPA过程执行(指定的措施)的跟踪，如培训活动、新开发工具、新流程改变等，将适当的反馈结果交付给负责CAPA的实体。
- 结果的跟踪，使我们能够准确地评估CAPA措施已经达到预期结果的什么程度。一般会把结果的反馈交付给改进方法的开发人员。

由此可见，正常的跟踪活动有利于及时了解信息和启动合适的反馈流，是纠正性和预防

性活动链中的重要部分。

5.5 支持性质量保证手段

如何把以前的质量控制经验传递到当前项目中？如何把一个开发组织的理解传递到整个组织？除了组织培训之外，还应该加强模板、检查表这类支持质量保证手段的应用。

5.5.1 模板

在软件工程中，模板指的是用于创建和编辑某种特定计划书、设计书、报告或其他形式的格式文档。模板的应用对很多文档是必需的，对有些文档则是选择性的。大多数模板可从 SQA 相关标准中获得或从组织内部取得。

使用模板有很多益处，主要如下。

- 简化文档评审工作，使文档的编制过程更加方便，节省了一些详细构建报告所需的时间和精力。
- 确保开发人员编制的文档更完善。模板一般经过大量的人员评审，所以不太可能出现漏掉主题这样的常见错误，从而减少人为错误。
- 对新组员有利。这是因为模板是根据标准结构编制的，所以新组员比较易于读懂和理解。
- 增加项目的可理解性让分工不同的组员可以理解一致。
- 维护人员在需要时，更容易快速找到所需信息。

SQA 负责编制新模板、应用模板和不定期更新模板。表 5-6 是某产品发布报告的模板内容。

表 5-6 产品发布报告模板

日期： 年 月 日	版本号：XX.XX	发布类型：补丁包
质量第一责任人：	所属部门：	联系方式：
审查组负责人：	签字质量经理：	
背景情况		
相关客户需求	任务编号	界面文档编号
测试环境		
平台：Windows 2K	页面 Java 版本	客户端版本号
数据库包 DB Patch	组件要求	
测试执行范围		
测试阶段	测试目标计划	测试日程
测试序列编号	质量跟踪编号	
没有测试区域	平台与浏览器矩阵表	
Bug 报告集编号		
功能检测		
没有完成的功能：	有 Bug 的功能：	

续表

系统有效性 安装测试完成 安全性测试以及容量测试	升级/恢复测试完成	性能压力测试
本版主要存在问题 严重 Bug 数量 第三方配合问题	一般 Bug 数量	没有解决的问题
全面质量分析 总体质量评估: 能否发布?		

5.5.2 文档建立、应用和更新

一个组织的质量保证体系的文档呈金字塔式的层次结构,如图 5-7 所示。其中,上面 3 层为质量保证活动提供支撑,最下一层是质量保证活动实施的过程和结果记录。

- 质量手册:质量管理体系的纲领性文档。它描述了整个组织的业务流程框架,质量管理的基本要求,为后续质量管理建设提供指导原则。
- 过程流程:组织内的关键业务流程描述文档。它明确了组织内的业务流程运作过程,各个部门之间的分工合作,关键节点上的配合。此部分相对固化,一旦流程出现了大的变化会涉及组织变革。
- 操作指导:具体组织单元如何操作。它指导具体业务工作的执行,提供过程流程的执行细则和配套指导,这些文档包括指导书、标准、规范、模板和检查表等。
- 质量记录:具体的执行过程记录。它是证明工作已按流程执行的证据。

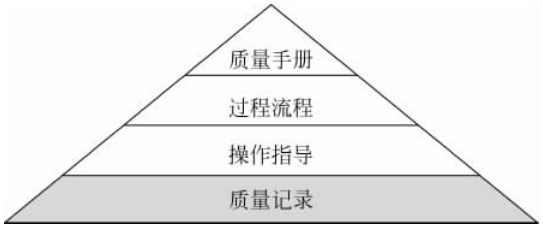

图 5-7 质量保证体系的文档层次结构

SQA 软件质量组织一般负责编制所需的常见类型的检查表和文档模板,编制最多的是第 3 层的操作指导类文档。过程流程一般由 SEPG(软件工程过程组)编制。质量记录一般是以软件项目为单位记录,由项目人员和质量人员共同编制。

1. 建立新的模板或调查表

文档(建立)组一般包括代表各种软件开发组织单元的软件工程师和 SQA 组织成员,其他成员如果愿意,也能加入文档组,这样的加入应该受到鼓励。文档编制的首要任务是找出编写文档所需要的清单,按清单上内容确定优先级。一般来说,最经常使用的功能要点应赋予较高的优先级。

编制新的文档可以从以下信息中获得支持。

- 本组织或机构中已经正式使用的非正式文档。
- 专业出版物或相关书籍中的文档示例。
- 类似组织或机构里使用的文档。

2. 文档模板的应用

一般文档模板的使用很少是强制性的，促进其使用是靠解释宣传和保证其可用性。所有的内部交流渠道都可以用来向组员解释宣传使用文档模板的好处，内部组员是 SQA 编制文档的“消费者”。在进一步应用新模板时会涉及以下问题。

- 应该怎么定位新写的文档模板，用哪些渠道宣传其益处？
- 如何让内部“消费者”在需要的时候能方便地获得这些文档模板？
- 哪些文档模板是强制性使用的？怎么样推进其应用？

3. 文档模板的更新

由于以下一些原因，模板需要更新。

- 用户的建议和意见；
- 技术更新、组织结构或客户关系的变化；
- 设计评审小组在对文档评审时所提出的建议；
- 一些特别的对文档内容有影响的案例；
- 其他组织或机构里的经验。

更新文档的过程与建立新文档模板类似。

5.6 软件质量改进

质量改进过程是质量管理体系的重要组成部分。根据朱兰的质量三部曲，质量管理分为 3 个基本过程：质量策划、质量控制和质量改进。产品在生产过程中由于质量缺陷而不得不返工、造成浪费，而这种经常性的浪费就为质量改进提供了机会。

ISO 9000 将质量改进定义为：致力于增强满足质量要求的能力是质量管理的一部分。

软件质量改进过程是一个持续过程，改进契机往往借助审核发现、审核结论、数据分析、管理评审或其他渠道获得的信息，通过制订改进目标，采用一系列结构化的步骤来达到改进软件过程质量或产品质量的目的。

CMM 体系的最高等级 5 级为优化级，它其实关注的就是软件开发过程的持续改进。通过对软件过程中所得到的经验教训加以改进，提高质量、优化过程，并应用到未来的软件开发过程中，由此促进软件组织不断走向成熟。丰田更是将持续改进(Kaizen)作为其核心理念之一。

5.6.1 软件质量改进模型

不同组织或行业可能有不同的质量改进过程或模型，从这些过程模型中可以看出质量改进的一般过程。泰戈的《质量工具箱》一书中提出了一种通用的质量改进“10 步骤”法。该过程基本上涵盖了改进步骤的所有要点，可以很好地表述质量改进过程。如果将质量改进过程以白话的形式表达出来，它实际上需要解决的是 10 个问题，表 5-7 体现了这 10 个步骤及其对应的质量改进术语。

表 5-7 质量改进的 10 步骤法

步　骤	拟解决问题	质量改进术语
1	我们想取得什么结果?	选择课题
2	谁关注? 他们关注什么?	了解客户需要
3	我们正在做什么? 我们做得怎么样?	调查现状
4	我们在什么方面可以做得更好?	设定改进目标
5	什么事情妨碍我们做得更好?	分析根本原因
6	为了做得更好? 我们需要什么样的改变?	寻求改进方案
7	采取行动!	实施改进方案
8	我们做得怎么样? 要不要再试一次?	效果确认
9	如果有效,如何保证每次都按照这个方法执行?	标准化
10	我们学到了什么?	项目总结

1. 我们想取得什么结果? ——选择课题

课题指的是需要解决的问题及质量改进的方向。课题来源可能是一个产品缺陷、外部客户问题反馈、内部质量记录等。如果能收集到组织内部痛点、TOP 质量问题等信息,将有助于锁定焦点。

选择课题时,特别要分析清楚:为什么选择这样的一个课题,它能带来的价值是什么。当课题还不是十分清晰时,可以采用脑力风暴等发散思维工具集思广益,或采用一些数据统计分析工具帮助发现问题。课题一旦选定,则尽可能地明确课题范围、关键步骤里程碑的项目计划,如什么时候完成现状调查,什么时候完成根因分析等。

2. 谁关注? 他们关注什么? ——了解客户需要

客户是产品或服务的使用者,掌握问题的第一手原始资料。列出该课题的相关客户,可以是内部客户,也可以是外部客户。当客户还不是十分明确时,可以借助 SIPOC 工具(参见 4.4.8 节 SIPOC)识别客户和供应商。

确定客户后,可以列一个信息收集计划确定需要了解的信息,包括:向哪些客户收集、收集信息方式、收集的信息内容等。例如,在访谈前列出访谈提纲,与关键客户进行深度的交流讨论。交流时务必注意倾听客户的声音,这有助于了解他们的真实需要。哪些问题是最困扰他们的,问题解决后给他们带来的直接感受是什么。例如,客户提出要一件丝绸服装,交流后了解到真实需要只是要吸汗,那么提供一件棉质服装也是可以的,还降低成本。收集到客户需求后,需要将客户需求进行分类、排序,可以借助 Kano 模型(东京理工大学教授狩野纪(Noriaki Kano)发明的对用户需求分类和优先排序的工具)分析客户需要与客户满意度的关系。

3. 我们正在做什么? 我们做得怎么样? ——调查现状

确认我们正在做的是否与客户需要一致,当前任务是否有效,可以通过绘制流程图等过程分析工具来帮助梳理任务过程。

要回答做得怎么样,则需要确定度量指标和采集到的数据。制订一个数据度量计划有助于现状的定量分析,包括:测量什么、测量频率、数据如何取得(包括取样方法)、数据如何记录等。需要注意的是,要同时考虑到产品和过程质量的测量。采集到的数据可以通过数据分析工具来进一步确认问题的分布。例如,客户对运维平台满意度不高的问题,主要发生

在运维平台上报给维护人员的告警信息不准确方面。

4. 我们在哪些方面可以做得更好？——设定改进目标

针对获得的信息和当前情况，进行改进可行性分析，确定改进目标。将第 2 步的客户需要与第 3 步的测量结果进行比较，可以看到预期与现状的差异。这个差异可能会很大，不能一下子全部改进。此时，还需要结合现状进行分析，哪些问题当前是可以改进的，哪些问题改进后可能会给客户带来显著效果，哪些问题改进起来相对比较困难。分析完成后设定一个切实可行，又带有挑战性的改进目标。例如，通过整改界面控件使 UI 规范符合度提高 10%。

5. 什么事情妨碍我们做得更好？——根因分析

根因指的是导致问题发生的关键因素，同时这种因素能被识别和纠正，消除了该因素，可防止问题的再次发生。根因是最基本、最深层次的原因。调查现状所得到的信息和数据是问题的现象，只有透过现象看本质，排除根本原因，才能彻底解决问题。在分析根因时，可以借助于鱼骨图、5Why 等根因分析工具，从不同分类层层深入。查找根因过程是复杂的，找到潜在根因后还需要进行确认是否为真正原因，这个过程可能会出现反复。识别出来的根因可能不止一个，存在多个方面不同层次。找到根因后则聚焦根因加以改进。

6. 为了做得更好，我们需要什么样的改变？——寻求改进方案

前面 5 个步骤都属于问题域，到第 6 步则开始进入解决方案域。针对每个根因逐条分析其改进方案。当用户想要一辆更快的马车时，脑子里并没有汽车、火车等形象，寻求解决方案的过程更多是一种创造性过程。可以借助于亲和图等发散思维工具帮助想出多种备选方案。在方案决策时，可以采用一些分析决策工具辅以方案选择。例如，对于界面控件难以对齐时，是人工对齐还是借助于工具对齐，这个时候需要分析人工和使用工具的成本，包括：工具熟悉、培训、引入成本，工具可以解决多大比例的对齐问题等。

方案一旦确定，则需要计划该方案的实施，包括方案实施可能的风险，需要的支持，实施结果如何交付，结果如何衡量等。建议能有一个改进项目文件夹，记录整个改进过程，包括计划、实施过程、输出件、度量数据、会议纪要、邮件讨论等。

7. 采取行动！——实施改进方案

针对改进方案的实施计划一步步执行。为避免方案的失败，可以先小规模实施，确认效果后再大范围执行。实施过程中要考虑验证的问题，以确认实施方案是否符合预期，可能的话，请客户参与实施过程。

8. 我们做得怎么样？要不要再试一次？——效果确认

分析改进方案实施以来所带来的改变。包括：

- 改进前和改进后的对比，现状是否发生了改变？
- 改进结果与预期目标的对比，目标是否已达到？

为了说明实施效果，需要收集一些数据进行量化分析，包括指标分析、趋势分析，与实施前的现状指标数据作对比，有可能的话，还可以做一些质量成本分析。

如果没有达到预期，可能需要返工，返回前面的步骤，找到正确根因并寻求更好的解决方案。

在这个步骤中可以采用一些数据收集和分析工具，如直方图、运行图等。

9. 如果有效，如何保证每次都按照这个方法执行？——标准化

这是一个真正体现质量改进核心价值的地方，它是预防性措施，避免同类问题再次出

现。所谓标准化,是将已取得的成功经验程序化、规则化以供重复使用。检查表、流程图等是常见的固化工具。标准化的过程中需要考虑未来应用中可能会出现的变化,并尽可能在标准程序中加以处理。

标准化的过程是一个经验提取过程,其输出件不可避免地带有一定抽象性。为了让标准化的成果得到更好的推广应用,则需要有配套的培训、宣传及应用案例等,并在应用过程中不断完善标准化成果。

10. 我们学到了什么? ——项目总结

回顾整个改进过程,总结得失成败,以及后续还需改进的地方。可以借助前面的过程记录文件回忆所经历的过程。如果有后续打算继续改进的课题,也可以列出。可以在组织层面分享改进项目经验:该课题是如何改进的,会给其他课题改进带来怎样的启发。

以上是10步骤法的详细的通用改进过程,虽然每个组织可能都有自己的质量改进过程,根据改进目的不同,所包括的过程及步骤描述粒度粗细不尽相同,但可以从通用过程中找出其中的对应关系。六西格玛的DMAIC改进流程与10步骤的对应,如表5-8所示。最终所有的过程都可以统一到PDCA的框架中去。

表5-8 DMAIC质量改进过程统一到PDCA框架

步骤	1	2	3	4	5	6	7	8	9	10
DMAIC	D界定		M测量	A分析		I改进			C控制	
PDCA	P计划						D执行	C检查	A行动	

5.6.2 软件质量改进实践层次

CMM体系的最高等级5(优化级)共有3个关键过程区域:缺陷预防、技术变更和过程变更,分别从缺陷、技术和过程3个方面对组织过程能力的持续改进提出了要求。从改进的层次来看,可以用图5-8描述。其中,最底层是发挥非组织力量,上面3层则属于组织级改进行为。通过例行活动或事件触发等方式,发起质量改进活动,并进而提高质量、增加组织能力。下面对改进的4个层次分别进行说明。

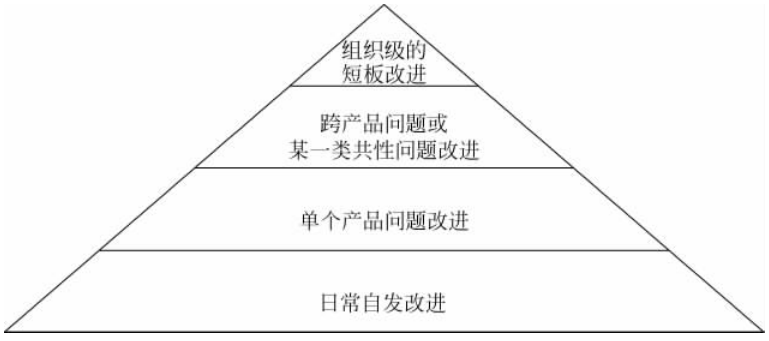

图5-8 软件质量改进实践层次

(1) 日常自发改进：主要来源于基层员工，从日常工作中寻找可改进点，并自主提出或发起改进。主要的改进活动形式有 QCC 及员工改进建议。

(2) 单个产品问题改进：针对单个产品内的问题进行改进，一般在某个项目内组织改进，重要问题则由产品来组织改进。产品问题来源可以是内部测试问题、外部客户外馈的一般线上问题。主要的改进活动形式有漏测问题分析及线上问题分析。

(3) 跨产品问题或某一类共性问题改进：这类问题的改进需要多部门协调进行，一般由主导产品组织联合周边产品、职能部门一起进行分析。问题来源有内部审计报告、外部线上事故等。主要改进活动形式是质量回溯。

(4) 组织级的短板改进：这类问题往往是组织内的共性重大问题，由上层组织发起改进，改进影响范围大，往往会引起组织变更、流程变更、技术变更等。问题来源有外部审计报告、认证报告、客户满意度调查报告等。主要改进活动形式是组织级变革项目。

5.6.3 品管圈

品管圈(Quality Control Circle，QCC)，由石川馨提出。由同一工作场所的员工自发组成一个工作小组，旨在发现并解决日常工作中有关的质量问题。要在一个组织内实施 QCC，需要具备以下基础条件。

- 全员普遍具备质量改进意识。
- 多数员工掌握一定的质量工具和质量管理手法。
- 组织提供支持，如 QCC 辅导员、提供相关培训。
- IT 支持。运用 IT 工具对改进活动过程进行管理。

以上几点都需要组织内管理层重视质量，并有一定的质量管理经验的积累。另外，对于 QCC 的课题选择，改进问题会很多，着重选易于改进的、聚集于技术层面的问题。

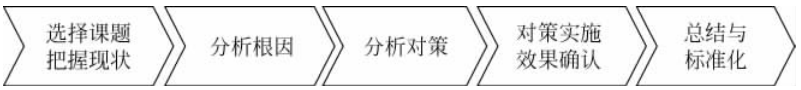

图 5-9 QCC 实施过程

QCC 实施过程如图 5-9 所示，可能存在部分步骤与有些组织内的实施不同，但不管怎么分，实施过程的基本内容与通用质量改进过程是基本一致的，只是名称不同，或将一些更为细致的步骤整合成一个大的过程有利于实施等。

【案例】 QCC——某产品 UI 界面原型与实现差异过大。

1. 选择课题、把握现状。

UI 界面往往是影响用户体验的重要环节，但 UCD 部门辛苦设计出来的高保真界面原型往往与最终实现相差甚远，就连一些基本的界面规范要求也达不到(如控件格式、字体大小统一等)，UI 规范不符合度达 50%，这些易用性问题已严重影响产品质量，过多的问题通过后期测试把关，往往投入成本高。为此，选择该课题有一定价值。表 5-9 为该案例的 QCC 实施应用。

采用 Checklist 界面标准进行符合度对照，并找出现有界面的问题，采用 Pareto 图对问题统计，发现问题主要集中在布局和控件两类，这两类中突出问题是间距、对齐、任务交互流

程不合理。经过讨论,决定将目标设在UI规范符合度达到70%以上。

表5-9 QCC实施应用案例——选择课题

产品版本	问题描述	问题大类	问题小类	检查标准
V1R1C001	按钮长度比文字还要短,显示有问题	布局	间距	按钮的宽度可随其文字长度自适应,考虑左右预留间距
V2R3C002	Tab键顺序设定不正确	控件	交互	Tab键按表单输入焦点、网页元素焦点的顺序切换
…	…	…	…	…

2. 分析根因

针对提取出的间距、对齐以及任务交互流程不合理这3种问题,逐一进行根因分析。图5-10所示为任务交互流程不合理问题的5Why根因分析过程,其他问题还采用了鱼骨图的分析方法,最终锁定了几类根因。

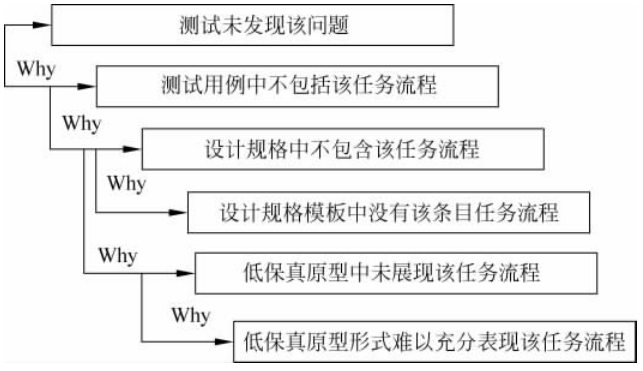

图5-10 QCC实施应用案例——根因分析

3. 分析对策

如表5-10所示,针对根因中的问题逐一进行对策分析。解决方案的好坏与最终的效果有很大关系,在这个阶段可以运用脑力风暴集思广益。另外,需要注意的是,解决方案需要从技术、流程等各方面考虑,真正从根本上解决问题。

表5-10 QCC实施应用案例——分析对策

主要原因	对策	措施	目标	实施地点	完成时间	责任人
间距和对齐主要依靠肉眼判断测试难度高	提供工具和方法,降低测试难度	·提供间距和对齐的测试工具 ·优化现有控件,在控件能力中扩展间隔和对齐的规范要求 ·提高可测试性,控件位置信息格式化输出,方便查看	测试人员不再反馈测试有问题	…	…	…
…	…	…	…	…		

4. 对策实施与效果确认

每条措施分配到责任人，按任务方式实施。实施过程及实施结束后，需要再次进行数据收集，以及与改进前的现状对比。此时，可以采用控制图、运行图等进行数据度量分析，如图 5-11 所示。

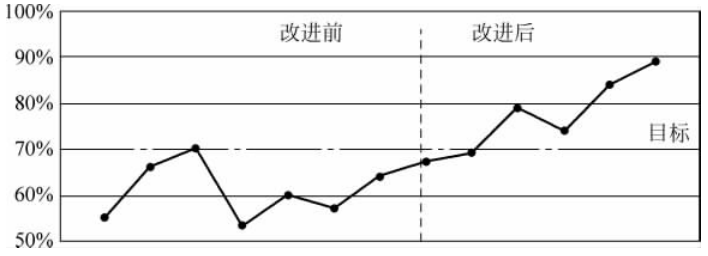

图 5-11　QCC 实施应用案例——效果确认

5. 总结与标准化

对工具、流程、模板、规范、标准更新等一系列进行固化工作，并进行相应的跟踪，如用问卷调查了解推广应用情况等。

5.6.4　漏测问题分析

漏测问题指的是，软件产品缺陷在测试过程中没有尽早被发现，包括后面的测试阶段发现前面测试阶段遗漏的问题，后面的测试或使用过程中发现以前版本的问题。此处的测试阶段可以是：同一个版本中不同的测试阶段，不同阶段中不同测试阶段。漏测问题分析就是针对流出到下一阶段的问题进行漏测分析，并采取相应的改进措施，减少漏测问题的发生。一般来说，漏测问题分析是每轮测试结束后的例行工作，同一个产品的问题在项目组内开展。

图 5-12 为漏测问题分析的过程。是否存在以前版本遗留下来问题，逐个分析漏测问题的根因。找到漏测试的根因后形成相应的改进措施。

图 5-12　漏测问题分析过程

1. 漏测问题识别

如图 5-13 所示，这一步相当于改进过程的现状分析，确认改进范围。判断“是否遗留 Bug”的依据：是否为以前版本遗留下来的 Bug。判断“是否漏测”的依据：测试版本与问题引入版本是否一致，该问题所在功能模块是否在引入版本经过测试，如果经过测试但没有测试出来则认为是漏测问题。例如，某问题是 V1R1C001 版本引入的，且该版本该功能经过测试，但却在后续版本 V1R1C002 中通过测试被发现。

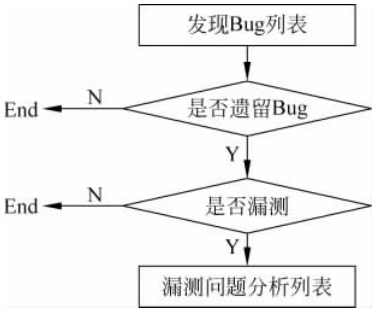

图 5-13　漏测问题识别

2. 分析漏测原因

对测试过程进行回放,结合测试过程及输出件的记录情况,判断是测试流程中哪一步的原因导致的漏测。常见的漏测分析过程如图5-14所示,针对找到的过程节点分析真正的根因。例如,用例执行遗漏还可以进一步深入分析下去,确定是因为用例问题还是执行人的疏忽。

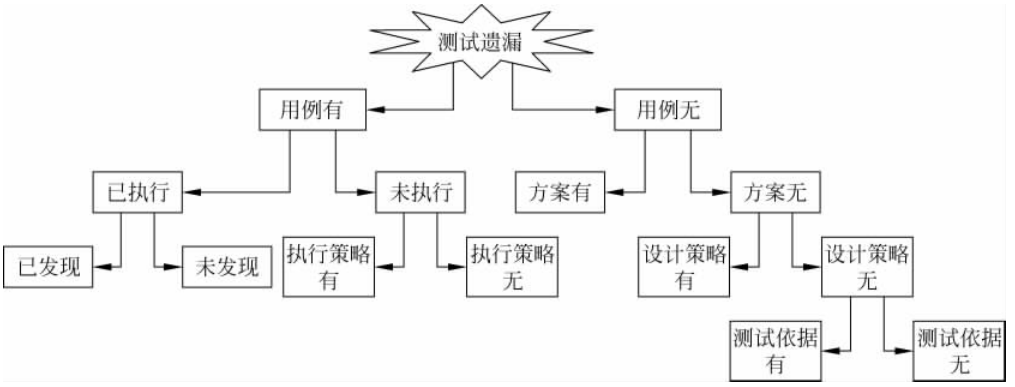

图5-14 分析漏测原因

3. 分析改进措施

对于漏测问题来说,它的改进措施主要是从测试角度来看针对该问题怎么避免漏测,对于问题本身的改进、怎么避免该问题发生则不在此考虑之内。需要注意的是,对于一些工具与测试流程改进等具有共性的、长效改进措施往往是纳入规划项目中统一考虑。常见的漏测改进措施如下。

(1) 更新测试用例。例如,测试用例增加验证点。

(2) 输出测试经验。例如,补充测试案例、添加到测试经验库。

(3) 提高测试技能。例如,进行相关测试技能培训。

(4) 改进测试工具。例如,完善自动化工具,增加对日志的验证功能。

(5) 改进测试流程。例如,修改测试方案模板,测试设计过程中增加观察点分析。

4. 实施与确认

实施相应的改进措施,并做例行的任务跟踪,检查测试用例库、测试案例库等是否增加了相应测试用例、测试案例。一般来说,对于单个问题的改进不需要做特别的数据分析。

5.6.5 质量回溯

回溯的中文意思指向上推导。产生产品质量问题代表没有满足客户需求。此处的产品质量问题指的是,来源于客户反馈的问题,严重一点即是质量事故。质量回溯即指从产品质量问题的现象出发,一步步地向上追踪问题的发生过程,找到导致问题的根本原因,并针对问题根因采取改进措施。此处的问题根因包括技术原因、管理原因和人为原因。改进措施包括纠正措施和预防措施。纠正措施是为消除已出现的该质量问题而提出的改进措施;预防措施是预防将来出现已识别出的质量问题,或类似质量问题而提出的改进措施。质量回溯活动一般在重大质量问题发生后启动,由产品负责人牵头组织,研发当事人、质量小组人员、技术规划组人员等多方代表共同参与。

质量回溯过程框架，如图 5-15 所示。质量回溯起始阶段是指问题原因定位、问题处理过程，即例行的问题解决过程。而后面的过程则是回溯到问题发生的源头、追问为什么问题会发生，消除了这些原因后，可防止问题再次发生。

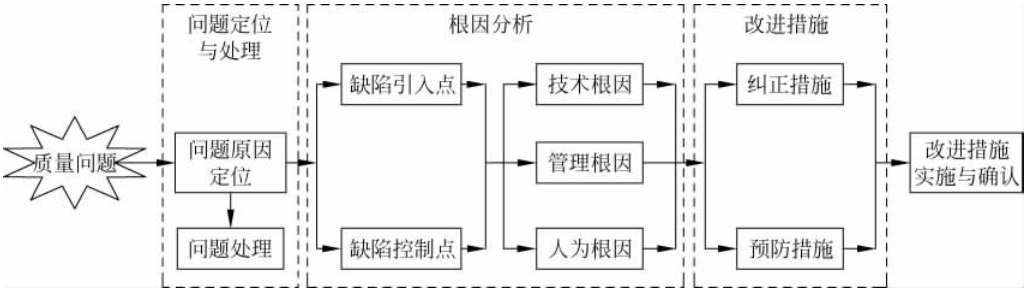

图 5-15　质量回溯过程框架

1. 问题定位与处理

描述问题发生→定位→处理的完整过程，具体内容包括：问题发生现象、问题发生时的场景、问题触发条件、问题发生过程、问题是如何定位的、收集了哪些数据、问题发生的直接原因是什么、做了哪些处理、最终问题是如何解决的、问题现象是否还存在等。

该过程除了一般的缺陷处理过程，还包含了一个应急处理过程，以减少现网事故对客户的影响。例如，某 A 产品的现网业务中断，经定位后发现：因第三方备份平台异常，没有按照约定及时取走数据库归档日志，导致数据库所在主机的磁盘空间满，数据库工作异常，从而业务中断。现场紧急启动数据库放通机制、业务恢复。

2. 根因分析

首先要确认问题的根源对象，明确缺陷引入点和缺陷控制点。缺陷引入点指缺陷在流程的哪一个点上产生、引入的；缺陷控制点指缺陷应在哪一个点被检查出、控制住。对于一个引入的缺陷来说，往往会有多个缺陷控制点，一般取离引入点最近的那个点作为缺陷控制点，最贴近缺陷的地方发现缺陷的效果最好。问题源头一般位于缺陷引入点，只有当缺陷引入点超出范围之外才会将缺陷控制点作为问题源头。例如，第三方供应商质量问题，只有通过缺陷控制点加以把关。软件开发过程中的缺陷引入点和缺陷控制点，如图 5-16 所示。例如，对某质量问题回溯发现缺陷引入点是系统需求，缺陷控制点则应在系统需求的评审阶段。在确认问题根源对象时需要借助研发过程记录，还原产品的研发过程。

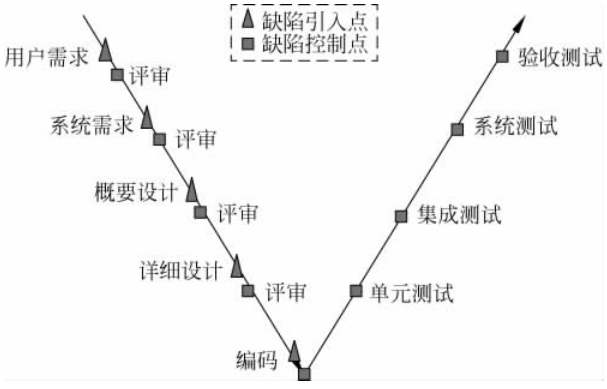

图 5-16　缺陷引入点和缺陷控制点

识别出缺陷引入点和控制点后,分别进行根因分析,从技术、人为和管理3方面展开分析,列出可能的原因以及各种原因之间的可能关系。根因分析工具因果图、5Why等可以在此应用。

技术根因指的是产品不符合某项需求规格(功能、性能等)要求。图5-17是技术根因的分析思路。针对不同的软件质量属性分别从规范、工具、方法、模板等来看,有没有相应的支撑手段。例如,某质量问题的缺陷引入点是可靠性系统需求遗漏,则分析可靠性规范中有无这样的条目,是不是缺少规范应用指导,是否缺少可靠性需求分析模板等。

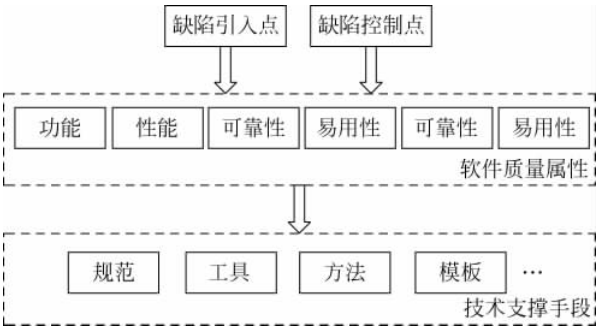

图5-17 技术根因分析

人为根因指的是造成问题产生的人为因素。人因分析从缺陷引入点和缺陷控制点来看,分析人在相应活动中的思考过程,并了解到底是如何产生这个问题的。质量大师克劳士比在《质量免费》一书中将人的错误分为知识技能和缺少关注两个方面,也即一个有意识,一个无意识。英国心理学家Jam Reason在*Human Error*这本书中将人的SRK认知模型分为3种:S-Skill技能;R-Rule规则;K-Knowledge知识。将人因基本错误归为3类:疏忽、遗忘和错误。疏忽指行动计划正确,但执行了错误的动作;遗忘指行动计划正确,但没有执行任何动作;错误指行动计划是错误的,执行的动作也是错误的。还有一种违规,属于故意性犯错,其原因难以控制,在人因分析中往往不予以考虑。错误可以进一步分为知识型错误和规则型错误。规则型指的是情景是熟悉的,但任务执行过程中选择规则时出现了错误。知识型指的是情景是不熟悉的,依赖自身知识经验加以诊断与决策,任务执行时发生了错误。

无意识犯错往往是任务大量执行后自动发生;而有意识犯错往往是对信息了解不足或思维过程出了问题。通常所说的技能不足往往指的是有意识犯错,这个问题可以通过技能培训加以改进,而无意识犯错更多则需要通过工具改进、思维训练等方式加以改进。图5-18对人为根因分析进行了概括。

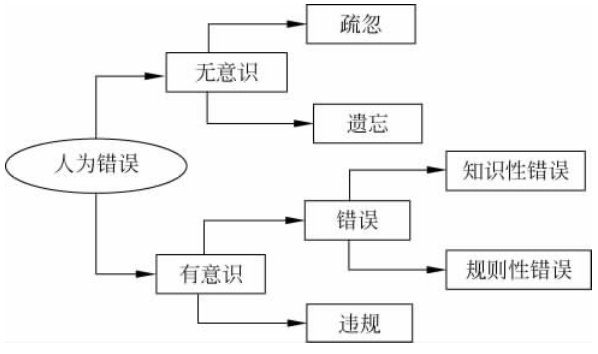

图5-18 人为根因分析

【案例】 某产品上网后出现与第三方设备对接失败。

根因分析结果为测试人员执行不正确，初步改进措施为提高人员技能。

进一步交流发现该产品对接接口中有 N 个相似含义字段且取值各不同，例如，A 接口 0 代表 OK，而 B 接口则 1 代表 OK，测试人员执行过程中面对大量的重复对比搞混了，还以为自己是对的。

识别出真正的人因后，更新改进措施为改进工具并增加功能进行接口字段自动比对。这种机械重复性工作改进工具比提高人员技能更为有效。

管理根因指从组织结构、资源管理和流程制度等各个方面寻找造成问题的原因，可以结合质量管理体系(Quality Management System，QMS)进行分析。在寻找管理根因时，围绕 QMS 的各个方面进行提问，如图 5-19 所示。常见的管理根因有：培训不充分、计划不完善、资源分配不适当、监控不充分、流程不完善、管理制度不完善等。例如，客户反馈用户手册质量差，在管理根因方面分析后，缺少专门的资料测试人员。

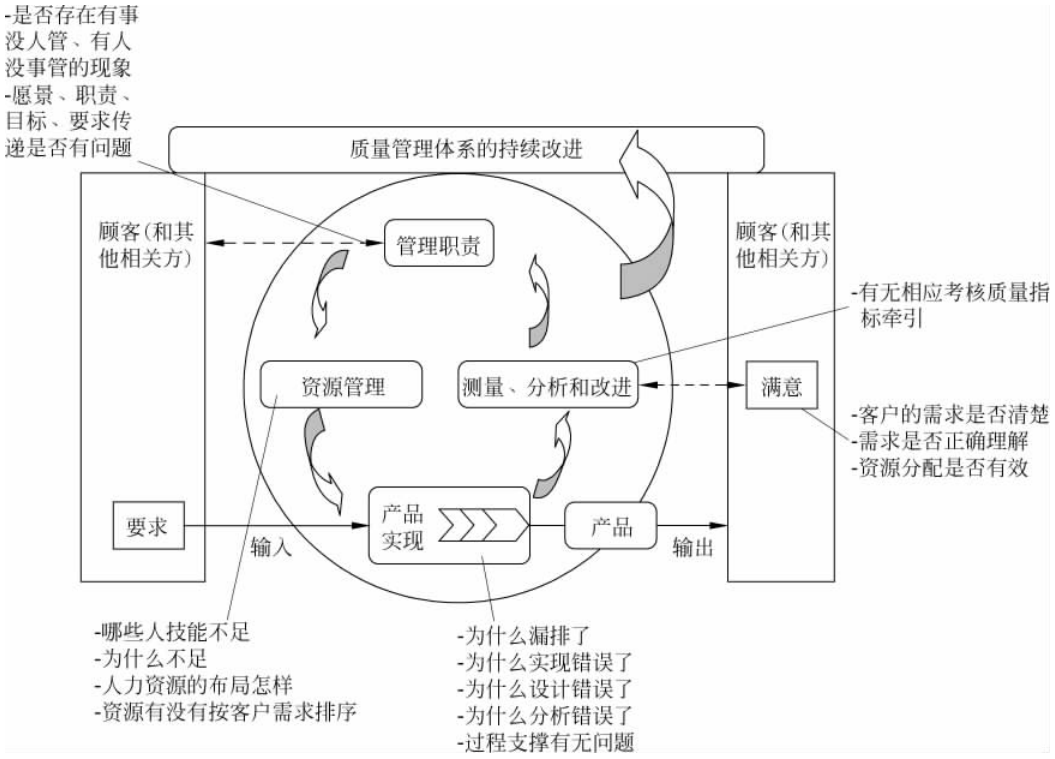

图 5-19 管理根因分析

3. 改进措施

首先确定哪些根本原因值得改进，并确定改进优先级。有些原因理论上可以改进，但不是很迫切或改进起来成本过高，超出组织承受范围的可以暂不考虑。分析改进措施时，要举一反三，与该问题类似的还有哪些情况需要改进。例如，与第三方平台的某接口 A 对接不通，则与此类似的接口 B 是否也需要改进。

对于改进措施的提出，除了纠正措施外，更多的是考虑预防措施，避免问题的再次发生。因此，在质量回溯里很多改进措施属于长期性的预防措施，实施的是固化、标准化的动作。

如规范建设、流程优化、组织机构调整等。

确立改进措施后,然后再制订明确的改进计划来实施改进措施,并纳入例行的计划实施管理中。在改进措施实施与确认过程中,改进措施不可能一下子就能达到一个很好的应用状态,需要在应用过程中不断完善。例如,流程先固化,再学习,再优化,往往存在实施与确认的多次反复,这是一个持续质量改进的循环过程。

综上所述,质量回溯过程中最重要的一环就是确定根因。表5-11是一个根因判别的检查表,供根因分析时对照参考。

表5-11 根因判别检查表

序号	检 查 项	检 查 要 求
1	该原因是否为问题原因链的源头或关键因素	·在逻辑上是问题原因链的源头(缺陷引入点) ·如果多个原因在逻辑层次上相同,则取关键的原因 ·如果缺陷引入点在能力范围之外,找缺陷控制点
2	该原因是否能够被识别	·根本原因应该是客观的、具体的、可度量的原因
3	该原因是否能够被纠正	·在目前的组织能力下,该原因可以被纠正 ·对原因的解决不能超出组织可承受的成本
4	如果消除了该原因,当前问题是否得到解决	·该原因被消除后,当前出现的问题即得到解决
5	如果消除了该原因,是否能避免此类问题再度发生	·当该原因被消除后,以后此类问题将不再发生,得到彻底解决 ·如果有些问题不能马上得到解决,但问题发生频率呈下降趋势

【案例】 某组织的质量回溯报告模板如表5-12所示,包含了问题定位与处理、根因分析、改进措施3个环节。

表5-12 质量回溯报告模板示例

问题描述	
详细描述问题的发生情况。例如,时间、地点、故障现象等	
问题处理	
简要描述对问题的应急处理情况。例如,修改参数、升级等,及时降低问题的影响度	
问题定位	
分析并找出问题的缺陷,缺陷要细化到具体需要修改的地方,缺陷修改后当前产品问题即刻得到解决。例如,某条语句赋值错误、变量类型错误、参数错误;软件函数的算法错误等	
根本原因分析(在缺陷引入点或缺陷控制点中找出导致缺陷的原因)	
1	缺陷引入点分析(缺陷是从哪个环节或活动中引入的): 一般指问题发生的根源对象。可对缺陷引入过程进行详细分析(可按时间或逻辑顺序展开)。例如,软件算法的问题可能是概要设计阶段造成;软件代码变量类型错误,一般是编码阶段造成
2	缺陷控制点分析(可选,缺陷在哪个环节或活动能够被阻止): 如果"缺陷引入点"质量无法控制或在能力范围之外,则需要分析缺陷控制点。一般情况下,需要分析离缺陷引入点最近的控制环节或活动。结合图5-15进行位置分析。例如,如果缺陷引入点是编码,则缺陷控制点是编码评审

续表

3	根本原因(在缺陷引入点、控制点中找出具体的、可识别、可改进的原因；这个原因导致了缺陷产生，可以从技术、管理、人为3个方面分析)： 一般情况，问题根本原因在缺陷的引入点中，但如果缺陷的引入点在公司或组织的能力范围之外，只能通过缺陷的控制点进行控制(例如，入口检查)。根本原因必须SMART化(具体的、可识别、可改进的原因)。 例如，代码某参数设置不正确，造成产品性能下降。问题的根因有可能为“对用户的需求不明确/规格对参数要求不明确”等，需要根据实际情况而定

改进措施：制订对当前根本原因的纠正和预防措施

	根本原因	纠正措施	预防措施	责任人	完成日期
1	根本原因必须SMART化	针对当前根因如何进行解决，纠正措施要SMART化	哪些措施能纳入到当前的质量管理体系中，防止问题重犯？例如：制订或更新流程、管理制度、技术规范、Checklist等；预防措施要SMART化		

5.6.6 持续改善

持续改善(Kaizen)是一个日语词汇，指小的、连续的、渐进的改进。它是由日本持续改进之父今井正明在《改善：日本企业成功的奥秘》一书中正式提出，在此书中认为丰田成功的关键在于贯彻了Kaizen的经营思想。Kaizen被看作是日本独有的全面质量管理(Total Quality Management，TQM)的核心理念。Kaizen活动包括以下6个步骤。

(1) 背景梳理：明确选择改善任务的理由。

(2) 现状分析：弄清当前情况的本质，并予以分析。

(3) 原因分析：弄清事情的真实背景及原因。

(4) 对策制订：在分析的基础上，研究并制订对策。

(5) 效果检验：观察并记录采用对策后的效果。

(6) 引入管理：引入标准和规范(包括修改和创建)，并对相关的标准和规范进行规范化，防止同类问题的再度发生，并建立相关问题的跟踪和解决机制。

可以看出，Kaizen活动过程是以PDCA为基础，与QCC步骤也极为相似，只是在不同步骤拆分存在细微的不同。

Kaizen的实施手法，主要是标准化、5S和消除浪费3个主要方面。

(1) 标准化。把改善成果固化下来，制订成可以重复使用的规则。标准化的目的在于预防，避免类似问题再次发生。

(2) 5S。来自以S开头的日语词汇，聚集于现场的可视化管理，对于软件研发，体现为软件工厂的管理，包括软件开发现场、软件实施现场、软件实验室等。

- Seiri(整理)：区分必要与不必要的，将不必要的丢弃。
- Seition(整顿)：摆放整理有序。

- Seico(清扫):把现场变得整洁。
- Seiketsu(检查):让干净整洁成为习惯。
- Shitsuke(素养):自觉遵守相关规程。

(3) 减少浪费。减少浪费正是精益思想的本质。精益思想源于20世纪80年代日本丰田的质量管理思想,“精”体现在质量上精益求精,“益”体现在成本上利益最大化。按照精益思想,任何不能为客户增加价值的行为即是浪费,而Kaizen就体现为减少浪费创造更多价值而进行的持续改善活动。将这个思想迁移到软件即是精益软件开发,Jack Mulinsky在agilesoftwaredevelopment.com上总结了软件开发过程中存在的7大浪费,如表5-13所示。

表5-13 软件开发过程中的浪费

序 号	浪费类别	浪费说明与举例
1	部分完成的工作	部分完成但没有最终落地的工作 • 没有转化成代码的设计文档 • 未及时合入的代码 • 没有相关说明文档的代码 • 未测试的代码
2	额外功能	研究表明,软件中有多达三分之二的功能几乎或从未被使用过 • 开发完成但没有被客户应用的特性 • 更多的代码 • 更高的复杂性带来额外维护工作量
3	再学习	由于经验不能传承带来的反复、额外的重复学习 • 分布式的开发团队,太多的代码迁移 • 人员频繁流动导致经验不能积累,反复重新学习 • 拥有某领域的专家,但在开发过程中需要此领域经验时,专家却没参与,由团队重新摸索
4	移交	知识信息的传递总是伴随信息丢失,隐性知识尤其困难 • 分工过细往往导致过多不必要的移交(如详细设计和实现分离)
5	任务切换	切换任务时,时间会浪费在切换背景和重置环境上,研究表明多任务工作会导致效率下降20%~40% • 一个人同时指派到多个项目中 • 一个人杂事繁多
6	等待	等待任务条件满足,因任务或资源相互依赖而导致工作停滞 • 集成时被关键模块阻塞 • 等待测试环境就绪 • 任务流程设置不合理,导致过多同步等待
7	缺陷	缺陷引起的返工、不可预期的损失 • 解决缺陷活动本身 • 缺陷越往后遗留,浪费越大

本章小结

本章主要论述了软件质量保证体系及软件质量改进相关内容。软件质量保证体系的内容包括：组织、结构、角色、职责、主要过程及活动、质量保证手段等。软件质量改进包括：软件质量改进模型、实践层次，并重点介绍了常见的一些软件质量改进活动。通过本章的学习，读者可以加深理解软件质量保证体系及活动，掌握软件质量改进的方法，增强质量预防意识，并能发现质量改进契机，以促进软件产品质量不断提高。

思考题

1. 简述 SQA 的体系、功能以及所进行的活动。
2. 常见的软件质量保证组织有哪些？
3. SQA 组织有几种组织结构模型？分别是什么？
4. 软件质量保证涉及多少种角色？他们的职能分别是什么？
5. CAPA 由哪几部分组成？各部分包括哪些内容？
6. 针对软件开发过程的某个具体活动编制一个模板。
7. 简述文档体系结构层次。
8. 简述软件质量改进模型的 10 步骤法，并说明每个步骤包含的主要内容。
9. 软件质量改进实践层次分为哪几层？每一层的主要活动形式有哪些？
10. 软件开发过程中常见的缺陷引入点和缺陷控制点有哪些？
11. 如何判别某问题原因是否为问题根因？
12. 软件开发过程中存在哪些浪费？结合实际开发过程，对每个浪费类别至少列举一个实例。
13. Kaizen 有什么含义？
14. 简述 QCC 的实施过程。
15. 如何判断某问题是否为漏测问题？
16. 简述软件质量回溯过程框架。

【大作业】

结合 QCC 的质量改进过程，选取一个质量改进主题，并完成相应的改进措施制订。要求如下。

- 现状分析：需要有相应数据支撑。
- 根因分析：需借助根因分析工具寻找到问题根因。
- 对策制订：需要形成具体的可执行的措施。

具体可参考附录 G：软件质量改进方案模板。