

CMM2 级的关键过程区域

[CMM-2 KPA]

Edit by SCMChina ++C

2002.10.01 v1.0



8848SOFTWARE-SCMChina

基本的项目管理过程是为跟踪成本，进度和功能而建立的。设立必要的过程规则可实现同以前项目的成功经验相似的运用。

CMM2 级的关键过程区域：

第一节 需求管理

第二节 软件项目计划

第三节 软件项目跟踪和监督

第四节 软件子合同管理

第五节 软件质量保证

第六节 软件配置管理

第一节 需求管理

目标是在客户和根据客户要求要在软件项目中定义的内容之间建立一种良好的理解。需求管理包括就软件项目与客户之间达成和维持共识。这种共识被称作"软件的系统需求"。"客户"可被理解为系统工程部，市场部，另一个内部机构或外部客户。该共识同时涵盖技术和非技术（如提交日期）的需求。这种共识形成了评估、计划、实行和跟踪软件项目活动的基础，贯穿整个软件生命周期。

系统需求在软件，硬件和其他系统元素（如人力）上的分配可能是由软件工程部之外的部门执行（如系统工程部），软件工程部可不直接控制这种分配。在项目约束下，软件工程部采取适当的步骤确保控制和存档软件方面的系统需求。

为达到这一控制，软件工程部门在最初及修改过的软件方面的系统需求被纳入软件项目之前对其进行审核以解决问题。无论软件方面的系统需求何时变化，都要调整受影响的软件计划、工作产品和活动以保持与更新后的需求一致。

目标

目标 1 控制软件方面的系统需求，建立一个基线用于软件工程和管理。

目标 2 保持软件计划、产品和活动与软件方面的系统需求一致。

行为的责任

责任 1 项目依照一个书面的组织性的原则，以管理软件方面的系统需求。

在这些实践中软件方面的系统需求被称为"分配需求"（allocated requirements）。

分配需求是系统需求的子集，在系统的软件部分中执行。它是软件开发计划中的主要部分。软件需求分析详细阐述和提炼了分配需求，结果体现为文档化的软件需求。

这一原则代表性地说明了：

1. 分配需求是文档化的；

2. 分配需求由：

软件经理及

其他受影响部门

进行审核。

受影响部门的例子包括：

系统测试

软件工程（包括所有的子部门，例如软件设计）

系统工程

软件质量保证

软件配置管理

文档支持

3. 软件计划、工作产品和活动应随分配需求的变化而修改，以与其保持一致。

行为的能力

能力 1 对于每个项目而言，建立分析系统需求和将其划分到硬件、软件和其他系统元素上的责任。

系统需求的分析和划分不是软件工程部的责任，但是是其工作的前提。

责任包括：

1. 在整个项目生命周期中管理和文档化系统需求及其分配。
2. 执行对系统需求及其分配的修改。

能力 2 分配需求是文档化的。

分配需求包括：

- 影响和决定软件项目的活动的非技术需求（如协议、环境、和/或合同条款）

协议，环境和合同条款的例子包括：

- 要提交的产品
- 提交日期
- 里程碑

2. 软件的技术需求：

技术需求例子包括：

- 最终用户，操作者，支持，或整体功能
- 性能需求
- 设计约束
- 程序语言
- 界面需求

3. 用于验证软件产品是否符合分配需求的接受标准。

能力 3 提供充足的资源和资金以管理分配需求。

1. 指定在应用领域和软件工程方面有经验和专门技术的人来管理分配需求。
2. 开发支持管理需求活动的工具

支持工具的例子包括：

- 电子数据表程序
- 用于配置管理的工具
- 用于追溯的工具
- 用于文档管理的工具

能力 4

培训软件工程部门和其他软件相关部门的成员执行其需求管理活动。

培训的例子包括：

- 方法，标准，项目所用的程序
- 应用领域

进行的活动

活动 1 软件工程部门在分配需求纳入软件项目之前审核之。

1. 鉴定未完成和遗漏的分配需求。
2. 审核分配需求，判断其是否：

- 具灵活性，适于在软件中执行。
 - 被清晰，恰当地陈述
 - 彼此一致
 - 可测试
3. 协同负责分析和分配系统需求的部门，审核任何被鉴别出有潜在问题的分配需求，并进行必要的修改。
4. 就由分配需求产生的责任与受影响的部门进行磋商。

受影响的部门的例子包括：

- 软件工程（包括所有子部门，如软件设计）
- 软件评估
- 系统工程
- 系统测试
- 软件质量保证
- 软件配置管理
- 合同管理
- 文档支持

参阅 KPA（软件项目计划）的活动 6 中之涉及责任磋商的实践活动。

活动 2 软件工程部门以分配需求作为软件计划、工作产品和活动的基础。

分配需求：

1. 是受管理和控制的。

"受管理和控制"意指了解某一时期（过去或现在）的工作产品的版本（既版本控制），改动是以受控的方式并入。（既改动控制）

若要求"管理和控制"所指的更深度的形式,工作产品可被放在配置管理的全部规则之下，即如在 KPA（软件配置管理）中所描述的。

2. 是软件开发计划的基础。

3. 是开发软件需求的基础。

活动 3 审核对于分配需求的改动并纳入软件项目中。

1. 评定对现存责任的影响，讨论改动使之适当。

1 协同高级经理审核对于个人和机构的外部组织的责任的改动。

参阅 KPA（软件项目计划）的活动 4 及 KPA（软件项目跟踪和监督）的活动 3 中涉及机构外部责任的实践活动。

1 就对于机构内部的责任的改动与受影响的部门进行磋商。

参阅 KPA（软件项目跟踪和监督）的活动 5，6，7 和 8 中涉及责任改动讨论的实践活动。

2. 对分配需求的改动带来的需要对于软件计划、工作产品和活动进行的改动是：

- 可鉴别的
- 经评估的
- 经风险评估的
- 文档化的
- 经计划的
- 与受影响部门和个人交流过的

□ 跟踪执行的

度量和分析

度量 1 进行度量，用来确定管理分配需求的活动的状况。

度量的例子包括：

- 每个分配需求的状况；
- 对于分配需求的修改活动；
- 对于分配需求改动的累积数量，包括经提议的，未决定的，获批准的和已纳入系统基线的改动的总数。

验证实施

验证 1 定期协同高级经理审核管理分配需求的活动。

高级经理定期审核的主要目的是在适当的抽象层次上及时获知和洞察软件过程活动。审核的间隔应满足机构的需要，可是长期的，只要具备足够的异议报告机制。

参阅 KPA（软件项目跟踪和监督）的验证 1 中涉及高级经理监督审核的典型内容的实践。

验证 2 定期或事件驱动下协同项目经理审核管理分配需求的活动。

参阅 KPA（软件项目跟踪和监督）的验证 2 中涉及项目经理监督审核的典型内容的实践。

验证 3 软件质量保证部门审核管理分配需求的活动和工作产品并报告结果。

参阅 KPA（软件质量保证）

作为最低限度，这些审核要验证：

1. 分配需求经过了审核，在软件工程部门提交之前问题得到了解决。
2. 当分配需求改动后，软件计划，工作产品和活动被相应修改。
3. 就由分配需求变化而引起的责任改动与受影响的部门磋商。

第二节 软件项目计划

软件项目计划目的是为遵循软件工程概念并管理软件开发项目而建立合理的计划。

软件项目计划内容包括：

- 估测软件开发各阶段工作产品的大小，以及所需要的资源。
- 制订时间表，评估相关风险，并协商各方面的责任。
- 按照客户的最终需求制订软件项目计划。

软件项目计划目标：

- 为便于计划和跟踪完成情况将有关软件各方面的估算写入文档。
- 计划完成软件项目的各种活动和相关责任，并将它写入文档。
- 有关的工作组和相关人员需同意承担他们的责任。

承诺：

任命一项目软件监督员（project software manager）负责协调各方面的责任并制订开发计划。

C1. 为软件项目制订计划需要遵循一个标准的组织方针。该方针规定：

1. 以软件的最终需求为基础。
2. 各项责任需要在项目负责人（project manager）、项目软件监督员、和其他软件负责人（software manager）间协调。
3. 需要同其他工作组参与时，要同他们协商并将过程写入文档。
4. 相关组对软件项目提出意见。
5. 当项目相关责任涉及到机构外人员或小组时需要更高层领导的审核。
6. 项目软件开发计划需要管理并控制（managed and controlled）。

前提条件：

A1. 有关项目软件订立一个文档化并得到一致认可的工作陈述（statement of work）。

1. 说明要包括从责任到目标到资源乃至时间表等所有相关内容。
2. 该说明要经由项目负责人、项目软件监督员、其他软件负责人、以及相关组一起审核。
3. 该工作陈述需要管理并控制。

A2. 制订软件开发计划的各项责任要落实到个人。

1. 项目软件监督员需亲自或指定相关人员协调计划的制订工作。
2. 软件工作产品及各项活动的职责需按可跟踪和可记录的方式划分并分配给各软件负责人。

A3. 为订立计划提供足够的资源和资金。

1. 各领域专家应尽可能参与。
 2. 要有订立计划活动的支持工具。
- A4. 相关人员（包括软件负责人和软件工程师）需要做软件评估和计划方面的培训。

执行动作：

AC1. 软件过程组要参与项目提案小组。

1. 涉及的内容包括提案的准备和提交、各项说明的讨论和提交、以及项目相关职责发生变动时的协商。

2. 过程组审核项目提案的各项承诺。

AC2. 软件项目计划需在整个项目计划的早期阶段订立并同时进行。

AC3. 软件工程组同其他相关组一起贯穿项目始终参与计划的制订（和修改），并负责审核项目级的计划。

AC4. 对机构外个人或团队所做的承诺需按照标准化的过程同高层管理者一起审核。

AC5. 软件生命周期中可管理的预定义阶段需标识并确定。

AC6. 项目软件开发计划需要按照标准化过程制订。

1. 软件开发计划需基于：客户标准、项目标准、产品说明、以及客户需求。
2. 其他工程组和软件相关组参与软件工程组活动的计划需相互协商、有关支出需预算、达成一致时需文档化。
3. 软件工程组参与其他工程组和软件相关组活动的计划需相互协商、有关支出需预算、达成一致时需文档化。
4. 制订的软件开发计划需要由项目负责人、项目软件监督员、各软件负责人及相关组审核。
5. 软件开发计划需要管理和控制。

AC7. 软件项目计划要文档化。计划内容包括：

1. 项目的目的、范围、目标以及成果。
2. 所遵循的软件生命周期模型。
3. 为开发和管理软件所选择的规程、方法和标准的标识。
4. 各软件工作产品的标识。
5. 各软件工作产品的大小以及变动情况。
6. 项目各项支出和成本的估算。
7. 关键计算机资源使用的估算。
8. 软件项目的时间表，包括重要阶段的识别和检查。
9. 各种项目软件风险的识别和评估。
10. 有关软件工程各种设施和支持工具的计划。

AC8. 建立和维护对软件项目的控制所需要的软件工作产品需要标识。

AC9. 按照文档化过程推导出对软件工作产品大小（或变动）的估算。

1. 所有主要的软件工作产品和活动的大小要估算。
2. 为达到估算的目标需要将工作产品分解到合适的粒度。
3. 尽量使用历史上已有的数据。
4. 有关大小估算的假设要文档化。
5. 大小估算要文档化、得到审核、并取得一致。

AC10. 按标准化过程导出对项目支出和成本的估算。

1. 应基于工作产品大小估算（及变动大小）作支出和成本的估算。
2. 应尽量使用目前或历史的生产率数据用于估算，相应的数据源及原由要文档化。数据可以来自机构内的其他项目，并且要考虑到生产工作产品的关键支出和成本。
3. 对成本、人员、和支出的估算应基于历史数据。如使用来自相同项目的数据，并确定时间段和预算所估算值在生命周期各阶段间的分布。
4. 估算值及所依据的假设要文档化、得到审核、并取得一致。

AC11. 按照标准化过程导出对关键计算机资源使用的估算。

1. 识别所需的关键资源。
2. 所作估算要相应于工作产品的大小、处理负载、以及通信量。
3. 所作估算要文档化、得到审核、并取得一致。

AC12. 按照标准化过程确定项目软件的时间表。

1. 时间表应基于工作产品的大小（变动大小）、以及软件支出和成本。
2. 时间表应基于已有项目的经验。
3. 时间表应包括关键阶段（milestone）的日期、关键依赖的日期以及其他有关约束。
4. 时间表中的活动应有适当的时间段、关键阶段要有适当间隔以确保进度测定的准确性。
5. 有关时间表的假设要文档化。
6. 时间表要文档化、得到审核、并取得一致。

AC13. 有关成本、资源、时间表、以及技术应用的软件风险要标识、评估、并文档化。

1. 按照潜在影响分析风险并确定优先级。
2. 识别风险的偶然性。

AC14. 制订有关工程设施和支持工具的计划。

1. 功能需求的估算应基于工作产品的估算及其他相关特性。
2. 为获得或开发这些设施和工具有关的职责要分配，相关承诺要协商。
3. 该计划要经由相关组审核。

AC15. 软件计划有关的数据要记录在案。

1. 记录所有的估算值，以及为重建这些估算值和评测其合理性所需的所有相关信息。
2. 计划数据要管理并控制。

度量分析：

M1. 测量主要被用于确定软件计划活动所处的状态。

验证：

V1. 同高层管理者一起周期性检查软件项目计划活动。

1. 技术、成本、人员和时间表的效能要审核。
 2. 底层不能解决的问题和矛盾要处理。
 3. 软件项目风险要解决。
 4. 各项任务需分配、审核并跟踪直至任务完成。
 5. 每次会议的总结性报告要准备好并分发给相关组和个人。
- V2. 项目负责人周期性或当重要事件发生时审核项目软件的计划活动。

1. 有关组要派代表。
2. 基于工作陈述和需求审核计划活动的状态和当前结果。
3. 各组间的依赖关系需要处理。
4. 底层不能解决的问题和矛盾需要处理。
5. 风险要审核。
6. 任务项要分配、审核并跟踪。
7. 会议的总结性报告要分发。

V3. 软件质量保证组要审核并审计计划活动和工作产品，同时汇报结果。审核内容包括：

1. 软件估算和计划活动。
2. 项目承诺的制定和审核活动。
3. 软件开发计划的准备活动。
4. 软件开发计划所采用的标准。
5. 软件开发计划的内容。

第三节 软件项目跟踪和监督

软件项目跟踪和监督目的是提供实际进程的充分的可视性，以便当软件项目的执行严重偏离软件计划时管理者能够采取有效的措施。

软件项目跟踪和监督包括根据文档化的评估、责任和计划跟踪和审核软件的完成和结果并在实际的完成和结果的基础上调整计划

文档化的软件项目计划（即在 KPA-软件项目计划中描述的软件开发计划）用于跟踪软件活动、交流状况和修改计划的基础。软件活动由管理者进行监控。当所选择的软件工作产品完成并处于所选择的里程碑时，首要通过实际软件规模、努力、成本和进度表与计划的对比来确定进程。

如果确认不符合项目计划，采取矫正措施。该措施可包括更改软件开发计划以反映实际完成情况、重新计划剩余的工作或采取行动改善操作。

目标

目标 1 根据软件计划跟踪实际结果和完成情况。

目标 2 如果实际结果和完成情况严重偏离软件计划，采取和管理矫正措施直至结束。

目标 3 软件责任的改变要经受影响部门和个人的同意。

行为的责任

责任 1 指定一个项目软件经理负责项目软件的活动和结果。

责任 2 项目依照书面的管理软件项目的组织性原则进行。

这一原则代表性地说明了：

1. 一个文档化的软件开发计划作为跟踪软件项目的基础运用和维护。
2. 项目经理始终了解软件项目的状况和问题。
3. 软件计划未获完成时采取措施，调整操作或调整计划。
4. 经受影响部门的参与和认可修改软件责任。

受影响部门包括：

- 软件工程（包括所有的子部门，如软件设计）
- 软件评估
- 系统工程
- 系统测试
- 软件质量保证
- 软件配置管理
- 合同管理
- 文档支持

5. 高级经理审核所有责任的变动和新的针对个人和机构外部部门的软件项目责任。

行为的能力

能力 1 文档化和批准软件项目的软件开发计划。

参阅 KPA-软件项目计划的活动 6&7 中涉及软件开发计划的实践。

能力 2 项目软件经理明确地分配软件工作产品和活动的责任。

分配的责任包括：

1. 要开发的软件工作产品或要提供的服务。
2. 软件活动的努力和成本。
3. 软件活动的进度表。

4. 软件活动的预算。

能力 3 提供充足的资源和资金用于跟踪软件项目。

1. 分配特定的责任给软件经理和软件任务负责人，以跟踪软件项目。
2. 开发支持软件跟踪的工具。

支持工具的例子包括：

- 电子数据表程序
- 项目计划/进度程序

能力 4 培训软件经理进行软件项目的技术和人事方面的管理。

培训的例子包括：

- 管理技术项目
- 跟踪和勘查软件规模、努力和成本和进度
- 管理人员

能力 5 一级软件经理接受软件项目技术方面的指导。

指导的例子包括：

- 项目软件工程标准和程序
- 项目的应用领域

进行的活动

活动 1 将文档化的软件开发计划用于跟踪软件活动和交流状况。

参阅 KPA-软件项目计划的活动 7 中涉及软件开发计划内容的实践。

软件开发计划是：

1. 随工作进程更新以反映完成情况，尤其当里程碑完成时。
2. 易于提供给：
 - 软件工程部门（包括所有子部门，如软件设计）
 - 软件经理
 - 项目经理
 - 高级经理
 - 其它受影响部门

活动 2 根据文档化的程序修订项目的软件开发计划。

参阅 KPA-软件项目计划的活动 6 中涉及制定软件开发计划的实践。

这一程序代表性地说明了：

1. 适当地修改软件开发计划，并入计划精炼度和计划的改动，尤其当计划有重大改动时。

分配给软件、设计限制、资源、成本和进度的系统需求间的相互依赖性要反映在所有计划的修改中。

2. 更新软件计划，并入所有新的软件项目责任和责任的改动。
3. 审核每一个软件开发计划的版本。
4. 软件开发计划是受管理和控制的。

"受管理和控制"意指了解某一时期（过去或现在）的工作产品的版本（既版本控制），改动是以受控的方式并入。（既改动控制）

若要求"管理和控制"所指的更深度的控制,工作产品可被放在配置管理的全部规则之下，即如在KPA（软件配置管理）中所描述的。

活动 3 根据文档化的程序协同高级经理审核软件项目为个人和机构外部门规定的责任以及责任的变动。

活动 4 对于影响软件项目的获批准的责任的变动要经与软件工程部门的成员和其他软件相关部门的交流。

活动 5 跟踪软件工作产品规模（或软件工作产品变动的规模），如有必要，采取改正措施。

参阅 KPA-软件项目计划的活动 9 中涉及规模评估来源的实践。

1. 跟踪所有主要软件工作产品的规模（或变动的规模）。
2. 比较实际代码的规模（生产的，完全测试的，提交的）与软件开发计划中存档的评估。
3. 比较实际提交文档的单元和软件开发计划中存档的评估。
4. 精炼，监控和定期调整软件工作产品的全部计划规模（评估的结合实际的）。
5. 和受影响的部门磋商对于软件工作产品规模评估的改动并将之文档化。

活动 6 跟踪项目软件的努力和成本，如有必要，采取改正措施。

参阅 KPA-软件项目计划的活动 10 中涉及成本评估来源的实践。

1. 比较随时间和完成的工作而花费的实际努力和成本与软件开发计划中存档的评估，以鉴定潜在的限度超出或不足。
2. 跟踪比较软件成本和软件开发计划中存档的评估。
3. 比较努力和员工与软件开发计划中存档的评估。
4. 与受影响的部门磋商影响软件责任的员工和其他软件成本的变动并文档化。

活动 7 跟踪项目的关键计算机资源，如有必要，采取改正措施。

参阅 KPA-软件项目计划的活动 11 中涉及计算机资源评估来源的实践。

1. 跟踪实际的项目所用的关键计算机资源，将其与软件开发计划中存档的每一个主要的软件成分的评估进行比较。
2. 与受影响的部门磋商影响软件责任的对于关键计算机资源评估的变动并存档。

活动 8 跟踪项目软件进度，如果必要，采取改正措施。

参阅 KPA-软件项目计划中活动 12 涉及的进度来源实践。

- 1 比较实际完成的软件活动、里程碑和其他责任与软件开发计划。
- 2 对于软件活动、里程碑和其他责任完成延迟或提前对以后的活动和里程碑将造成的影响进行评估。
- 3 与受影响的部门磋商影响软件责任的软件进度修改。

活动 9 跟踪软件工程技术活动，如有必要，采取改正措施。

1. 软件工程部门成员定期向他们的一级经理报告其技术状况。
2. 比较 Software release contents for successive builds 和软件开发计划中存档的计划。
3. 报告和存档任何软件工作产品中发现的问题。
4. 跟踪问题报告直至结束。

活动 10

跟踪与项目的成本、资源、进度和技术方面相关的软件风险。

参阅 KPA-软件项目计划的活动 13 中涉及风险鉴定的实践。

1. 获得新增信息后对首要风险和偶然风险进行调整。
2. 定期协同项目经理审核高风险区域。

活动 11 记录软件项目的实际度量数据和再计划数据。

参阅 KPA-软件项目计划的活动 15 中涉及记录项目数据的实践。

1. 记录信息包括评估以及重建评估所需的相关信息并验证其合理性。
2. 软件再计划数据是受管理和控制的。
3. 将软件计划数据、再计划数据和实际度量数据存档以便为正在进行的和以后的项目所用。

活动 12 软件工程部门定期进行内部审核以依照软件开发计划跟踪技术进程、计划、成果和问题。

审核是在下列部门间进行：

1. 一级软件经理和他们的软件任务负责人。
2. 项目软件经理、一级软件经理和其他软件经理。

活动 13 根据文档化的程序在选择的项目里程碑处进行确定软件项目完成情况和结果的正式审核。

这些审核：

1. 计划于软件项目进度表中有意义的节点上进行。
2. 与客户、最终用户和机构内部受影响部门共同进行。

这些实践中提到的最终用户是指客户指定的最终用户或最终用户的代表。

3. 采用经负责的软件经理审核和批准的材料。
4. 确定责任、计划和软件活动的状况。
5. 产生鉴定结果和有关重大问题、活动项目和决议的文件。
6. 确定软件项目风险。
7. 必要的话，精炼软件开发计划。

度量和分析

度量 1 进行度量，用于确定软件跟踪和监督活动的状况。

度量的例子包括：

- 用于进行跟踪和监督活动的努力和其他资源；

- 软件开发计划的修改活动，包括对于软件工作产品的规模评估、软件成本评估、关键计算机资源评估和进度的修改。

验证实施

验证 1 定期协同高级经理审核软件项目跟踪和监督的活动。

高级主观定期审核的主要目的是在适当的抽象层次上及时获知和洞察软件过程活动。审核的间隔应满足机构的需要，可是长期的，只要具备足够的异议报告机制。

1. 审核技术、成本、员工和进度表的进行情况。
2. 确定在较低层次上未获解决的矛盾和问题。
3. 定义软件项目风险。
4. 分配、审核和跟踪活动项目直至完成。
5. 完成并分发每个会议的汇总报告给受影响的部门和个人。

验证 2 定期或事件驱动下协同项目经理审核软件项目跟踪和监督的活动。

1. 体现出受影响的部门。
2. 根据软件开发计划审核技术、成本、员工和进度情况。
3. 审核关键计算机资源的利用；根据原始评估报告这些关键计算机资源的当前评估和实际利用。
4. 定义部门间依赖关系。
5. 确定在较低层次上未获解决的矛盾和问题。
6. 审核软件项目风险。
7. 分配、审核和跟踪活动项目直至完成。
8. 完成并分发每个会议的汇总报告给受影响的部门和个人。

验证 3 软件质量保证部门审核软件项目跟踪和监督的活动和工作产品并报告结果。

参阅 KPA (软件质量保证)

作为最低限度，这些审核要验证：

1. 审核和修改责任的活动。
2. 修改软件开发计划的活动。
3. 修改后的软件开发计划的内容。
4. 跟踪软件项目的成本、进度、风险、技术、设计限制、功能性和执行情况的活动。
5. 进行有计划的技术和管理审核的活动

第四节 软件子合同管理

软件子合同管理目的是选择具有资格的软件分包人并有效地管理他们。

软件子合同管理包括选择软件分包人，与其确立责任，跟踪和审核其表现和结果。这些实践包括对软件分包人的管理以及对软件分包人的软件元素（包括软件、硬件和可能的其他系统元素）的管理。

对分包人的选择是基于其完成工作的能力。许多因素决定了要将一部分主承包人的工作转包出去。对分包人的选择可能处于战略经营同盟或技术的考虑。

转包当中，包含技术和非技术（如提交时间）要求的文档化的协议建立起来并作为管理这一转包的基础。分包人应做的工作和这一工作的计划文档化。分包人遵从的标准和主承包人的一致。

由分包人进行分包工作的软件计划、跟踪和监督活动。主承包人确保这些活动恰当地进行且分包人提交的软件产品符合其接受标准。主承包人协同分包人管理其产品和过程的界面。

目标

目标 1 主承包人选择具有资格的软件分包人。

目标 2 主承包人和软件分包人就其责任达成协定。

目标 3 主承包人和软件分包人维持现有的交流。

目标 4 主承包人根据软件分包人的责任跟踪其实际结果和完成情况。

行为的责任

责任 1 项目依照一个书面的组织性的原则，以对管理软件分包进行管理。

这一原则代表性地说明了：

1. 文档化的标准和程序应用于对分包人的选择和对分包的管理。
2. 合同式协议构成管理分包的基础。
3. 转包的变更经主承包人和分包人的参与和同意。

责任 2 指派转包经理负责建立和管理软件的转包。

1. 转包经理在软件工程上有丰富的知识和经验或有可指派的具有上述经验的个人。

2. 转包经理负责与相关各方协调转包工作的技术范围以及转包的条款和条件。

项目的系统工程部门和软件工程部门定义转包工作的技术范围。

适当的经营功能部门，如采购、财务和法律顾问，规定并监控转包的条款及条件。

3. 转包经理负责：

- 选择软件分包人，
- 管理软件的转包，
- 安排被转包产品的转包后的支持。

行为的能力

能力 1 为软件分包人和管理软件的转包提供充足的资源和资金。

1. 指派软件经理和其他个人专门负责管理转包。

2. 开发支持管理转包的工具。

支持工具包括：

- 评估模型
- 电子数据表程序

□ 项目管理和进度程序

能力 2 培训参与建立和管理软件转包的软件经理和其他个人执行其活动。

培训的例子包括：

- 准备并计划软件的转包
- 评估转包投标人的软件过程能力
- 评价转包投标人的软件评估和计划
- 选择软件分包人
- 管理转包

能力 3 参与建立和管理软件转包的软件经理和其他个人接受转包的技术方面的指导。

指导的例子包括：

- 应用领域
- 运用的软件技术
- 所用的软件工具
- 所用的方法
- 所用的标准
- 所用的程序

进行的活动

活动 1 根据文档化的程序定义和计划要转包的工作。

这个程序代表性指明：

1. 要转包的软件产品和活动的选择是基于对项目的技术和非技术特性的平衡估价。
 - 选择适合潜在分包人的技术和能力的功能或子系统转包。
 - 基于对系统和软件需求的系统分析和适当的分割来确定转包的软件产品和活动的规格。
2. 要转包的工作和遵从的标准和程序来源于项目的：
 - 工作陈述
 - 分配给软件方面的系统需求
 - 软件需求
 - 软件开发计划
 - 软件标准和程序
3. 工作的转包陈述是：
 - 预备的
 - 经审核的
 - 经同意的

审核和同意上述工作陈述的个人的例子包括：

- 项目经理
- 项目软件经理
- 负有责任的软件经理
- 软件配置管理经理
- 软件质量保证经理
- 分包人经理

1 必要时修改

1 受管理和控制

"受管理和控制"意指了解某一时期（过去或现在）的工作产品的版本（既版本控制），改动是以受控的方式并入。（既改动控制）

若要求"管理和控制"所指的更深度的控制,工作产品可被放在配置管理的全部规则之下,即如在KPA(软件配置管理)中所描述的。

参阅 KPA-软件项目计划中活动 1 涉及的工作陈述的典型内容的实践。

4. 选择分包人的计划的准备和转包工作陈述同时进行并经审核。

活动 2 依据文档化的程序,基于对转包投标人完成工作的能力的评估,选择软件分包人。这个程序代表性地指明:

1. 提交有计划的转包的提议。
2. 如有的话,之前相似工作的完成记录。
3. 与主承包人相关的转包投标人的机构的地理位置。

一些转包的有效的管理可能要求频繁的面对面的互动。

4. 软件工程和软件管理能力。

一个评估分包人的能力的方法是 SEI"软件能力评估"法。

5. 可进行该项工作的人员。
6. 之前的类似应用的经验,包括转包人的软件管理队伍中的软件专门技术。
7. 可利用的资源:

资源的例子包括:

- 设备
- 硬件
- 软件
- 培训

活动 3 主承包人和分包人之间的合同式协议构成管理分包的基础。

合同式协议文件:

1. 条款和条件。
2. 工作陈述。

参阅 KPA-软件项目计划中活动 1 涉及工作陈述的典型内容的实践。

3. 对于要开发的产品的需求。
4. 主承包人和分包人之间依赖关系清单。
5. 要提交给主承包人的被转包的产品。

产品的例子包括:

- 源代码
- 软件开发计划
- 模拟环境
- 设计文档
- 接受测试计划

6. 提交修改产品的条件。
7. 在主承包人接受被转包的产品之前,用以评估该产品的接受程序和接受标准。

8. 主承包人用以监督和评估分包人的执行情况的程序和评估标准。

活动 4 主承包人审核和批准分包人的文档化的软件开发计划。

1. 该计划涵盖（直接或参照性的）主承包人的软件开发计划中的适当项目。

有些情况下，主承包人的软件开发计划可能包含了分包人的软件开发计划而不再要求单独的分包人的软件开发计划。

参阅 KPA-软件项目计划中活动 7 涉及软件开发计划的典型内容的实践。

活动 5 一个文档化的经批准的分包人的软件开发计划用于跟踪软件活动和状况交流。

活动 6 对于分包人的工作陈述、分包项目和条件以及其他责任的改变要依照文档化的程序解决。

1. 该程序代表性地指明所有主承包人和分包人中受影响的部门都要参与。

活动 7 主承包人管理指导对于软件分包人的管理状况的定期审核。

1. 分包人被给予对于产品客户和最终用户的需求的可视性。
 2. 根据分包人的软件开发计划审核分包人的技术、成本、人员和进度执行情况。
 3. 审核指定为项目关键的计算机资源，跟踪分包人对当前状况的评估并与分包人软件开发计划中对于每一软件元素的评估进行比较。
 4. 确定分包人的软件工程部门和其他分包人的部门间的关键依赖性和责任
 5. 确定主承包人和分包人之间的关键依赖性和责任。
- 1 分包人对于主承包人的责任和主承包人对于分包人的责任都要进行审核。
6. 核查与转包要求不一致的地方。
 7. 确定与转包人工作有关的项目风险。
 8. 确定转包人内部未解决的矛盾和问题。
 9. 分配、审核行动项目并跟踪直至结束。

活动 8 与软件分包人进行定期技术审核和交流。

这一审核：

1. 提供分包人对于产品客户和最终用户的需求的可视性。
2. 监督分包人的技术活动。
3. 验证分包人对于技术要求的阐释和实施符合主承包人的要求。
4. 验证责任与实际相符。
5. 验证技术问题以及及时的态度得到解决。

活动 9 根据文档化的程序，在所选择的里程碑处，进行确定分包人完成软件工程情况的正式审核。

这个程序代表性地指明：

1. 该审核在工作陈述中预先计划并章程化。
2. 该审核确定分包人的软件活动责任、计划和状况。
3. 鉴定和存档重大问题、行动项目和决策。
4. 确定软件风险。
5. 提炼转包人的软件开发计划。

活动 10 主承包人的软件质量保证部门根据文档化的程序监督分包人的软件质量保证活动。

这个程序代表性地指明：

1. 定期审核分包人的软件质量保证计划、资源、程序和标准以确保其足以监督分包人的工作执行情况。

2. 定期进行对分包人的审核以确保经批准的程序和标准得以遵循。
 - 主承包人的软件质量保证部门现场检查分包人的软件工程活动和产品。
 - 主承包人的软件质量保证部门审核分包人的软件质量保证记录。
3. 定期审核分包人的软件质量保证活动的记录，评定其遵循软件质量保证计划、标准和程序的情况。

活动 11 主承包人的软件配置管理部门根据文档化的程序监督分包人的软件配置管理活动。这个程序代表性地指明：

1. 审核分包人的软件配置管理计划、资源、程序和标准以确保其适当。
2. 主承包人和分包人就与软件配置管理有关部门的事宜协调其活动，以确保分包人的产品易于整合或并入主承包人的项目环境。
3. 定期审核分包人的软件基线库，评定其遵循软件配置管理标准和程序的情况以及它们如何有效管理软件基线。

活动 12 主承包人根据文档化的程序进行接受测试，作为分包人软件产品提交的一个部分。这个程序代表性地指明：

1. 测试之前，双方共同定义、审阅和批准每一产品的接受程序和接受标准。
2. 接受测试的结果文档化。
3. 为任何未通过接受测试的软件产品建立行动计划。

活动 13 定期对软件分包人的执行表现进行评价，该评价经分包人审阅。

评价软件分包人的执行表现为分包人提供了获得其是否符合其客户（即主承包人）要求的反馈的机会。这种反馈表现为一种机制，如表现奖励费用审核机制，与贯穿项目的定期协调和技术审核相对立。这些评价的文档亦作为以后选择分包人的依据。

度量和分析

度量 1 进行度量，用于确定管理软件转包活动的状况。

度量的例子包括：

- 管理转包活动的成本与计划的对比。
- 实际转包产品的提交日期与计划的对比。
- 主承包人递交转包人的实际日期与计划的对比

验证实施

验证 1 定期协同高级经理审核管理软件转包的活动。

高级经理定期审核的主要目的是在适当的抽象层次上及时获知和洞察软件过程活动。审核的间隔应满足机构的需要，可是长期的，只要具备足够的异议报告机制。

参阅 KPA（软件项目跟踪和监督）的验证 1 中涉及高级经理监督审核的典型内容的实践活动。

验证 2 定期或事件驱动下协同项目经理审核管理软件转包的活动。

参阅 KPA（软件项目跟踪和监督）的验证 2 中涉及项目经理监督审核的典型内容的实践活动。

验证 3 软件质量保证部门审核管理软件转包的活动和工作产品并报告结果。

参阅 KPA (软件质量保证)

作为最低限度, 这些审核要验证:

1. 选择转包人的活动。
2. 管理软件转包的活动。
3. 对于主承包人和分包人的配置管理活动的协调活动。
4. 对分包人审核计划的执行。
5. 完成转包人关键项目里程碑或阶段的审核的执行。
6. 转包人软件产品的接受过程。

第五节 软件质量保证

软件质量保证目标是提供在软件过程中对于软件项目和软件产品的恰当监督和管理。

软件质量保证包括审核软件产品和活动，验证其是否符合适用程序和标准并为软件项目和其他相关经理提供审核结果。

软件质量保证部门在软件项目初期协同制定计划、标准和程序。通过这种协同工作，软件质量保证部门确保其符合项目要求并可用于贯穿软件生命周期的审核的进行。软件质量保证部门在整个生命周期审核项目活动和软件工作产品并为调查软件项目是否符合其计划、标准和程序提供可视性管理。

一致性问题首先在软件项目中发现并解决。对于在软件项目内无法解决的问题，软件质量保证部门将其提交适当的管理层解决。

这一关键过程区域包含了该部门执行软件质量保证功能的实践。鉴定该部门要审核的特定活动和工作产品的实践普遍包含在其他 KPA 的验证实施这一公共特性当中。

目标

目标 1 软件质量保证活动是有计划的。

目标 2 客观地验证软件产品和活动与适用标准、程序和需求的一致性。

目标 3 将软件质量保证活动和结果通知受影响的部门和个人。

目标 4 在软件项目内无法解决的不符点问题由高级经理审定。

行为的责任

责任 1 项目遵循一个书面的组织性原则开展 SQA（软件质量保证）。

这一原则代表性地指明：

1. SQA 功能应用于所有软件项目。
2. SQA 部门拥有一个向高级经理报告的渠道，其独立于：
 - 项目经理
 - 项目软件工程部门
 - 其他软件相关部门

其他软件相关部门的例子包括：

- 软件配置经理
- 文档支持

应战略性经营目标和经营环境之需，机构必须确立支持需要独立性的活动（如 SQA）的组织结构。

该独立性应：

- 提供承担 SQA 任务的个人以作为高级经理的软件项目的“耳目”的机构内自由。
- 保护其不受被审核的软件项目管理层的行为评价。
- 提供高级经理对于软件项目的过程和产品的反对信息得到汇报的信心。

3. 高级经理定期审查 SQA 的活动和结果。

行为的能力

能力 1 建立负责协调和执行项目的 SQA 的小组。（即 SQA 部门/小组）

小组是指负责一系列任务和活动的部门，经理和个人的组合。一个小组可能是一个指定的兼职个人，或从不同部门指派的几个兼职个人，或几个专职个人。成立一个小组要考虑被分配的任务或活动、项目的规模、组织结构和机构文化。一些小组，如 SQA 小组，重点是项目活动，而其他一些，如软件工程过程小组，重点则是机构范围内的活动。

能力 2 为进行 SQA 活动提供充足的资源和资金。

1. 指派一个经理专门负责项目的 SQA 活动。
2. 指派一个具有 SQA 知识并有权采取适当监督措施的高级经理接受软件不符点并采取行动。

所有向高级经理进行 SQA 报告的经理都了解 SQA 的任务、责任和权利。

3. 开发支持 SQA 活动的工具。

支持工具的例子包括：

- 工作站
- 数据库程序
- 电子数据表程序
- 审核工具

能力 3 SQA 小组的成员受到培训以进行 SQA 活动。

培训的例子包括：

- 软件工程技术和实践
- 软件工程部门和其他软件相关部门的任务和责任。
- 软件项目的标准、程序和方法
- 软件项目的应用领域
- SQA 的目标，程序和方法
- SQA 小组在软件活动中的参与
- SQA 方法和工具的有效利用
- 人与人之间的交流

能力 4 软件项目的成员接受关于 SQA 小组的任务、责任、权利和价值的培训。

进行的活动

活动 1 根据文档化的程序制定软件项目的 SQA 计划。

这一程序代表性地指明：

1. SQA 计划是在全部项目计划的初期进行并与之同步发展。
2. 由受影响的部门和个人审核 SQA 计划。

由受影响的部门 and 个人的例子包括：

- 项目软件经理
- 其他软件经理
- 项目经理
- 客户 SQA 代表
- SQA 小组应向之报告不符点问题的高级经理
- 软件工程部门（包括所有子部门，如软件设计和软件任务负责人）

3. SQA 计划是受管理和控制的。

"受管理和控制"意指了解某一时期（过去或现在）的工作产品的版本（既版本控制），改动是以受控的方式并入。（既改动控制）

若要求"管理和控制"所指的更深度的控制,工作产品可被放在配置管理的全部规则之下，即如在KPA（软件配置管理）中所描述的。

活动 2 SQA 小组的活动的进行与 SQA 计划一致。

计划包括：

1. SQA 小组的责任和权利。
2. SQA 小组的资源需求（包括人员，工具和设备）。
3. 项目的 SQA 小组活动的进度和资金。
4. SQA 小组参与建立项目的软件开发计划、标准和程序。
5. SQA 小组进行的评估。

受评估的产品和活动的例子包括：

- 操作软件和支持软件
- 可提交的和不可提交的产品
- 软件性的和非软件性的产品（如文档）
- 产品开发和产品验证活动（如测试）
- 制造产品过程中的活动

6. 审核由 SQA 小组执行。
7. 项目标准和程序用作 SQA 小组审核的基础。
8. 文档化并跟踪不符点问题直至结束的程序。

这个程序可作为计划的一部分或包含于其他文档的参考当中。

9. SQA 小组应制订的文档。
10. 为软件工程部门和其他软件相关部门提供关于 SQA 活动的反馈的方法和频率。

活动 3 SQA 小组参与项目软件开发计划、标准和程序的制定和审核。

1. SQA 小组提供关于如下的计划、标准和程序的咨询和审核：
 - 与组织性原则的一致性
 - 与外部强制标准和需求的一致性（如工作陈述中要求的标准）。
 - 适于项目运用的标准
 - 应在软件开发计划中确定的主题
 - 项目指定的其他区域。

活动 4 SQA 小组审核软件工程活动，验证其一致性。

1. 根据软件开发计划和指定的软件标准和程序评估活动。

参阅其他 KPA 中验证实施这一公共特性中涉及 SQA 小组进行的专门审核的实践。

2. 鉴定、存档和跟踪偏离直至结束。
3. 对修改进行验证。

活动 5 SQA 小组审核指定的软件工作产品，验证其一致性。

1. 可提交的软件产品在提交给客户之前经受评估。
2. 根据指定的软件标准、程序和合同要求评估软件工作产品。

3. 鉴定、存档和跟踪偏离直至结束。
4. 对修改进行验证。

活动 6 SQA 小组定期向软件工程部门报告其活动结果。

活动 7 根据文档化的程序对软件工作产品和软件活动中鉴定出的偏离存档和处理。

这一程序代表性地指明：

1. 协同适当的软件任务负责人、软件经理或项目经理解决与软件开发计划和指定的软件标准和程序的偏离问题并存档。
2. 上述过程无法解决的偏离存档并呈交指定接受不符点问题的高级经理。
3. 定期审核 2 中问题直至其解决。
4. 不符点问题的文档是受管理和控制的。

活动 8 SQA 小组适当地定期与客户的 SQA 成员审核其活动和发现。

度量和分析

度量 1 进行度量，用来确定 SQA 活动的成本和进度情况。

度量的例子包括：

- SQA 活动里程碑的完成与计划的对比
- SQA 活动中完成的工作、花费的努力和消耗的资金与计划的对比
- 产品审核和活动审核的数量与计划的对比。

验证实施

验证 1 定期与高级经理审核 SQA 活动。

高级经理定期审核的主要目的是在适当的抽象层次上及时获知和洞察软件过程活动。审核的间隔应满足机构的需要，可是长期的，只要具备足够的异议报告机制。

参阅 KPA（软件项目跟踪和监督）的验证 1 中涉及高级经理监督审核的典型内容的实践活动。

验证 2 定期或事件驱动下协同项目经理审核 SQA 的活动。

参阅 KPA（软件项目跟踪和监督）的验证 2 中涉及项目经理监督审核的典型内容的实践。

验证 3 独立于 SQA 小组之外的专家定期审核 SQA 小组的活动和软件工作产品。

第六节 软件配置管理

目标是在整个软件生命周期中建立和维护软件项目中的产品的完整性。

包括整个软件生命周期中按时在给定节点上鉴定软件（如所选的软件工作产品及其描述）的配置，系统地控制配置的改动及维护配置的完整性和可追溯性。软件配置管理下的工作产品包括提交给客户的软件产品（如软件需求文档和代码）以及生产这些软件产品所需要的东西（如编译器）。

建立一个包含软件基线的软件基线库。通过改动控制和软件配置管理的配置审核功能系统地控制对于基线的改动和从软件基线库中生成的软件产品的释放。

这一 KPA 包含了执行软件配置管理功能的实践。鉴定特定配置的实践包含在描述了每个配置项/模块的开发和维护的 KPA 中。

目标

- 目标 1 软件配置管理活动是有计划的。
- 目标 2 所选的软件工作产品经鉴定，受控制，是可利用的。
- 目标 3 对鉴定过的工作产品的改动是受控的。
- 目标 4 受影响的部门和个人了解软件基线的状况和内容。

行为的责任

责任 1 项目依照书面的组织性的原则进行软件配置管理（SCM）。

这一原则代表性地指明：

1. 清晰地分配每一项目的 SCM 的责任。
2. SCM 的进行贯穿整个项目生命周期。
3. SCM 是针对可对外提交的软件产品，指定的内部软件工作产品和用于项目内部的指定的支持工具（如编译器）。
4. 建立到达配置项/模块的存储库的通路。

这个库的内容即这些实践中提到的“软件基线库”。

访问这个库的工具和程序即“配置管理库系统”。

放在配置管理之下作为单一实体的工作产品被称为配置项目。

配置项目被典型地分解成配置元素，配置元素被典型地分解为模块。在一个硬件/软件系统中，所有的软件可被看作一个配置项目，或可将软件分解成多个配置项目。这里“配置项/模块”用于表示配置管理下的元素。

5. 定期审核软件基线和 SCM 活动。

行为的能力

能力 1 建立一个有权管理项目的软件基线的委员会（即软件配置控制委员会 SCCB）。

SCCB：

1. 授权建立软件基线及配置项/模块的识别。
2. 代表项目经理和所有可能受软件基线改变的影响的部门的利益。

受影响部门的例子包括：

- 硬件质量保证
- 硬件配置管理
- 硬件工程
- 制造工程
- 软件工程（包括所有子部门，如软件设计）
- 系统工程
- 系统测试
- 软件质量保证
- 软件配置管理
- 合同管理
- 文档支持

3. 审核和授权对软件基线的改动。
4. 授权由软件基线库中生成产品。

能力 2 建立一个负责协调和执行 SCM 的小组。

SCM 小组协调或执行：

1. 建立和管理项目的软件基线库。
2. 制定、维护和分配 SCM 计划，标准和程序。
3. 放在 SCM 下的一系列工作产品的鉴定。

工作产品是从定义、维护和利用一个软件过程中得到的任何人工产物。

4. 管理对软件基线库的访问。
5. 更新软件基线库。
6. 由软件基线库中生成产品。
7. 记录 SCM 的活动。
8. SCM 报告的生成和发布。

能力 3 为进行 SCM 活动提供充足的资源和资金。

1. 指派一个专门负责 SCM 的经理。
2. 开发支持 SCM 活动的工具。

支持工具的例子包括：

- 工作站
- 电子数据表程序
- 配置管理的工具

能力 4 在进行 SCM 活动的目标、程序和方法方面对 SCM 小组的成员进行培训。

培训的例子包括：

- SCM 标准、程序和方法
- SCM 工具

能力 5 软件工程部门和其他软件相关部门的成员接受进行 SCM 活动的培训。

其他软件相关部门的例子包括：

- 软件质量保证
- 文档支持

培训的例子包括：

- 在软件工程部门和其他软件相关部门内进行的 SCM 活动所要遵循的标准、程序和方法
- SCM 小组的任务、责任和权利

进行的活动

活动 1 根据文档化的程序为每一个软件项目制定 SCM 计划。

这一程序代表性地指明：

1. SCM 计划是在全部项目计划的初期进行并与之同步发展。
2. 由受影响的部门审核 SCM 计划。
3. SCM 计划是受管理和控制的。

活动 2 文档化的经批准的 SCM 计划用于进行 SCM 活动的基础。

计划包括：

1. 要进行的 SCM 活动，活动的进度，分配的责任和需要的资源（包括员工，工具和计算机设备）。

2. 软件工程部门和其他软件相关部门进行的 SCM 需求和活动。

活动 3 建立一个配置管理库系统作为软件基线库。

这个库系统：

1. 支持 SCM 的多个控制层次。

产生多个控制层次的情况的例子包括：

在生命周期中的不同时间所需的控制层次不同（如产品成熟时地较紧控制）。

只具软件系统和包括软、硬件的系统所需的控制层次不同。

2. 提供配置项/模块的存储和回复。

3. 提供受影响部门间和不同控制层次间的配置项/模块的库内共享和传递。

4. 帮助使用配置项/模块的产品标准。

5. 提供配置项/模块的存储和存档版本的恢复。

6. 帮助确保由软件基线库中正确生成产品。

7. 提供 SCM 记录的存储、更新和回复。

8. SCM 报告的支持生产。

9. 提供库结构和内容的维护。

库维护功能的例子包括：

库文件的备份/修复

库错误的回复。

活动 4 识别放在配置管理下的软件工作产品。

1. 根据文档化的标准选择配置项/模块。

可鉴定为配置项/模块的软件工作产品的例子包括：

过程相关的文档（如，计划、标准或程序）

软件需求

软件设计

软件代码模块

软件测试程序

为软件测试活动而进行的软件系统构建

为向客户或最终用户提交进行的软件系统构建

编译器

其他支持工具

2. 配置项/模块指定唯一标识符。

3. 规定每一个配置项/模块的特性。

4. 规定每一个配置项/模块从属的软件基线。

5. 规定每一个配置项/模块放在配置管理下的节点。

6. 规定负责每一个配置项/模块的人（即所有人，从配置管理的角度来看）。

活动 5 根据文档化的程序开始、记录、审核、批准和跟踪所有配置项/模块的改动要求和问题报告。

活动 6 根据文档化的程序控制对基线的改动。

这一程序代表性地指明：

1. 进行审核或逆行测试以确保改动未对基线产生未预期的影响。

2. 只有经 SCCB 批准的配置项/模块进入软件基线库。
3. 以维持软件基线库正确性和完整性的方式录入/出配置项/模块。

录入/出的例子包括：

- 验证更改是经授权的。
- 建立更改日志
- 保持更改复制
- 更新软件基线库
- 存档替换后的软件基线

活动 7 根据文档化的程序从软件基线库生成产品及控制其释放。

这一程序代表性地指明：

1. SCCB 授权由软件基线库中生成产品。
2. 由软件基线库生成的产品，无论是为内部所用还是外部所用的，都是仅从软件基线库中的配置项/模块中构建。

活动 8 根据文档化的程序记录配置项/模块的状况。

这一程序代表性地指明：

1. 充分详细地记录配置管理活动以便了解每个配置的内容和状况及恢复先前的版本。
2. 维护每个配置项/模块的当前状况和历史（即改动和其他活动）。

活动 9 开发存档 SCM 活动和软件基线内容的标准报告并使受影响的部门和个人能利用它。

报告的例子包括：

- SCCB 会议记录
- 改动需求总结和状况
- 问题报告汇总和状况（包括解决情况）
- 软件基线改动汇总
- 配置项/模块的改动历史
- 软件基线状况
- 软件基线审核结果

活动 10 根据文档化的程序进行软件基线审核。

这一程序代表性地指明：

1. 对于审核有充分的准备。
2. 评估软件基线的完整性。
3. 审核配置管理库系统的结构和设备。
4. 验证软件基线库内容的完备性和正确性。
5. 验证与适用的 SCM 标准和程序的一致性。
6. 向项目软件经理报告审核的结果。
7. 跟踪审核的活动项目直至结束。

度量和分析

度量 1 进行度量，确定 SCM 活动的状况。

度量的例子包括：

- 每一单位时间处理的改动需求的数目
- SCM 活动里程碑的完成与计划的对比
- SCM 活动中完成的工作，花费的努力和消耗的资金。

验证实施

验证 1 定期协同高级经理审核 SCM 活动。

验证 2 定期或事件驱动下协同项目经理审核 SQA 的活动。

参阅 KPA（软件项目跟踪和监督）的验证 2 中涉及项目经理监督审核的典型内容的实践活动。

验证 3 SCM 小组定期审核软件基线，验证软件基线符合对其定义的文档。

验证 4 软件质量保证部门审核 SCM 的活动和工作产品并报告结果。

参阅 KPA（软件质量保证）

作为最低限度，这些审核要验证：

1. 以下符合 SCM 的标准和程序：

- SCM 小组
- SCCB
- 软件工程小组
- 其他软件相关小组

2. 定期审核软件基线的进行。