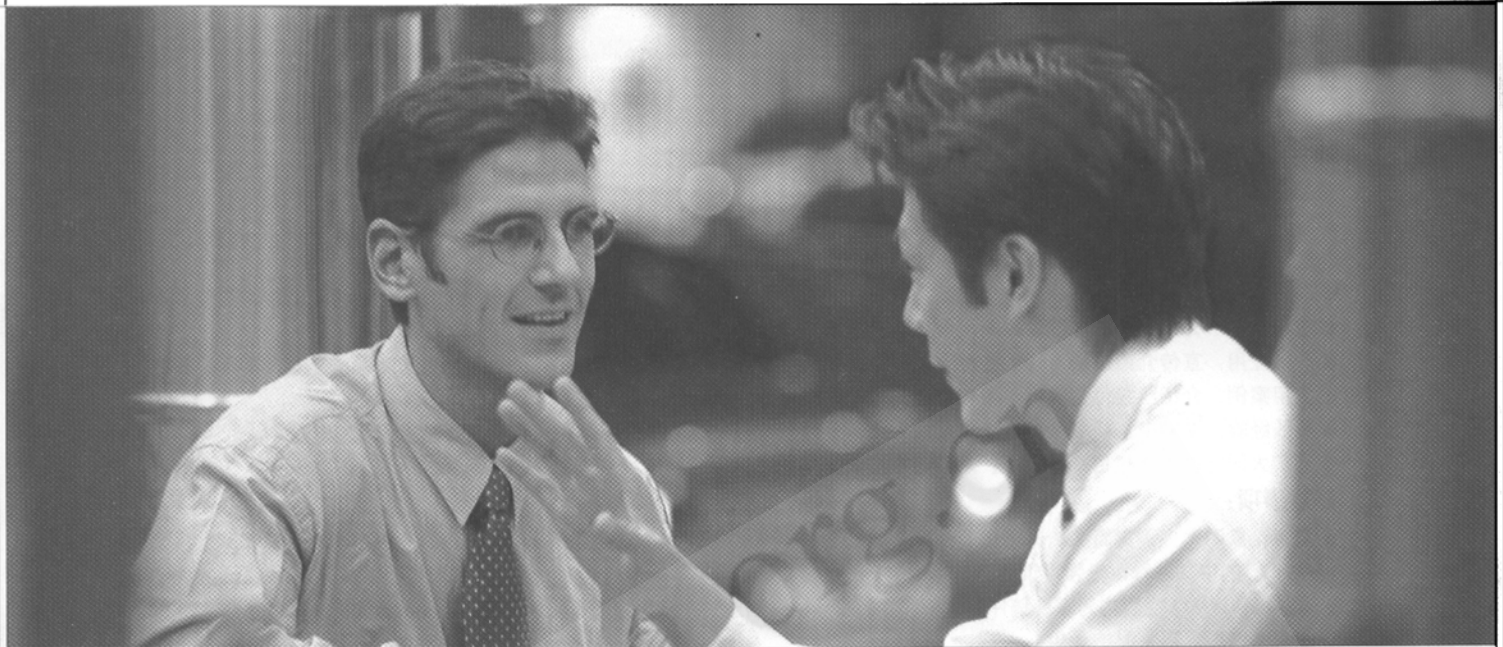


SW-CMM 同级评审 KPA 软件管理流程的设计



张效民 武磊 (北京中国科技大学研究生院 100039)

摘要: 同级评审是 SW-CMM 三级的一个关键过程域, 其目的是尽早消除软件工作产品中的缺陷。本文针对同级评审关键过程域的需求, 结合本软件开发组织的特点, 设计了该关键过程域的软件管理流程。

关键词: 软件能力成熟度模型 关键过程域 软件过程 同级评审

1 软件能力成熟度模型简介

SW-CMM 描述了一个有效的软件过程的各个关键元素, 指出了软件企业如何摆脱杂乱无章的、不成熟的软件过程, 形成一个成熟的、有纪律的软件过程所必经的进化、提高的途径--判断企业当前的过程成熟状况, 找出在改进过程中急需解决的若干问题, 然后依据 SW-CMM 选择过程改进策略, 提高企业的软件过程能力。

SW-CMM 由 5 个成熟度等级组成 (见表 1) 每个成熟度等级有着各自的功能。在总共 18 个 KPA 中, 共有 52 个目标 (Goal), 和 316 个关键过程 (KP)。虽然可以用 52 句话 (52 个 Goal) 及 500 多页的文档描述 CMM 的需求, 但这些只描述了我们应该“做什么”, 而没有描述“如何”去达到这些目标。

2 什么是同级评审

同级评审是指与软件工作产品生产者处于

同一级别的人员, 系统的检测软件工作产品, 找出其中的错误并确定需要更改的领域。

同级评审关键过程域的目的是尽早消除软件工作产品中的缺陷, 并使缺陷的消除过程变得更加有效。同时, 通过同级评审会对软件工作产品和能够预防的缺陷有一个更好的理解。

同级评审工作要纳入软件项目计划的活动中, 需要进行同级评审的软件工作产品, 在项目定义软件过程中加以标识。该关键过程域仅说明执行同级评审的实践, 其目标是:

- 有计划的进行同级评审;
- 识别并消除软件工作产品中的缺陷。

3 同级评审关键过程域管理流程的设计

3.1 同级评审流程的定义和类型

同级评审 (以下简称评审) 工作是针对同事的软件产品进行的有组织的检查, 目的是发现缺陷并评估与以前阶段文档的一致性。一般

有 3--7 人参加, 每人具有以下角色中的一个或多个。

- 主持人: 管理整个流程并批准评审的结果的角色;
- 作者: 生成或修改软件产品的角色;
- 读者: 负责向参加评审的人员展示被审资料, 根据自己的理解, 解释每一条语句或每一段文档的含义的角色;
- 记录者: 负责记录在评审的过程中发现的所有与缺陷相关的信息的角色;
- 测试者: 从测试的角度检查评审资料的角色;
- 其他参与者: 检查评审资料, 寻找缺陷的角色

其中, 一个参与者可以担当多个角色, 但作者、主持人和读者三个角色是互斥的, 即一个参与者只能同时担当这三个角色中的一个角色。另外, 由于某种特殊原因 (作者个人的性

Design of Software Process in Peer Review KPA

格过于保守或过于自信), 作者可以不参加评审会议, 为防止会议的进程难以控制, 会议可只有主持人、记录者和读者三人参加。

3.1.1 判定是否需要评审的标准

(1) 如果是进行代码的合并 (merge), 下面情形不需要进行评审:

- 只是去除一个分支 (branch);
- 合并中未出现代码冲突, 且所有代码都已经过审。

(2) 在重用代码时, 要根据上下文来判定是否需要评审, 下面情形需要进行评审:

- 重用的代码被改变时, 将按新代码处理;
 - 接口重用。
- (3) 数据表的改动需要进行评审。

(4) 所有新工作和新改动需要进行评审。

3.1.2 评审流程的类型

根据评审任务量的大小, 评审流程可分为两类: 正式评审和微型评审。

正式评审: 不适合应用微型评审的评审, 遵循 4.4 节的流程进行。

微型评审: 对于一些改动很小且低风险的文档或代码变化, 将采用微型评审, 它由至少二名熟悉产品的人员参加, 遵循 4.5 节的流程进行。

3.1.3 进行微型评审的标准

只有同时满足以下 8 个条件时, 才能进行微型评审:

(1) 代码或文档改动量小于等于 6 行 (不含注释行), 并在小于等于 3 个文件中; 如果代码和文档都有改动, 两者相加的行数要小于等于 6 行 (不含注释行);

(2) 如果代码改动涉及到多个文件版本, 则每个版本的改动要小于等于 3 行 (不含注释行), (所有改动相加的行数要小于等于 6 行);

(3) 改动只涉及到一个子系统或一个系统分层;

(4) 对于改动的分析, 用一段文字就可以描述清楚;

(5) 在改动中, 没有类型定义或非局部变量定义的变更;

(6) 没有增加新的功能;

(7) 没有增加新的代码文件;

(8) 作者和评审者一致同意不需要正式的评审。

如果以上任一条件不满足, 都需要进行正式评审。

3.2 评审的开始条件

软件产品必须满足以下条件才能进行评审:

文档:

- (1) 必须通过电子的拼写检查;
- (2) 必须使用正确的文档模版;
- (3) 必须归档 (check in);
- (4) 为便于纪录, 文档材料必须具有行号和页号;

代码:

- (1) 所有文件必须无编译错;
- (2) 所有改动必须符合编码规范;
- (3) 所有文件必须归档 (check in);
- (4) 为便于纪录, 所有打印的文件必须具有行号和页号;

表 1 CMM 的分级结构

级别	工作重点(focus)	关键过程域(KPA)
5 优化级	不断过程改进	缺陷防范 (DP, Defect Prevention) 技术改革管理 (TCM, Technology Change Management) 过程变更管理 (PCM, Process Change Management)
4 已管理级	软件过程和软件产品的定量管理	定量过程管理 (QPM, Quantitative Process Management) 软件质量管理 (SQM, Software Quality Management)
3 已定义级	项目和组织的策略	组织过程焦点 (OPF, Organization Process Focus) 组织过程定义 (OPD, Organization Process Definition) 培训程序 (TP, Training Program) 集成软件管理 (ISM, Integrated Software Management) 软件产品工程 (SPE, Software Product Engineering) 组间协调 (IC, Intergroup Coordination) 同级评审 (PR, Peer Review)
2 可重复级	建立软件项目管理的控制流程	需求管理 (RM, Requirements Management) 软件项目计划 (SPP, Software Project Planning) 软件项目跟踪与监控 (SPTO, Software Project Tracking and Oversight) 软件转包合同管理 (SSM, Software Subcontract Management) 软件质量保证 (SQA, Software Quality Assurance) 软件配置管理 (SCM, Software Configuration Management)
1 初始级	专业人士与个人能力	

3.3 评审的结束条件

- (1) 所有评审记录已被录入到数据库中;
- (2) 对软件流程的改进建议已提交。

3.4 正式评审的流程

评审流程如下图 1 所示:

评审流程包括七个阶段, 其中必须经过计划、准备、会议、修改和后续工作五个阶段, 背景知识介绍和第三小时会议两个阶段为可选阶段。

评审流程中用到了一些自定义的专有名词, 定义如下:

- 缺陷(Fault): 软件产品中发现的失误;
- 错误(Error): 当前开发阶段引入的缺陷;
- 过错(Defect): 以前开发阶段引入的缺陷。

3.4.1 计划阶段

在此阶段中, 需要完成以下工作 (除特殊标明, 以下工作均由作者完成):

(1) 选择一名受过培训的人员作为评审的主持人;

(2) 评审小组的组成: 作者或主持人选择参加评审的其他人员, 一个人可担当多个角色, 但作者、主持人和读者三个角色是互斥的; 测试者这个角色在需求评审中是必须的, 在其他评审中是可选的; 需要时, 可与项目经理协商选出适当的评审小组; 若此修改涉及多个子系统, 则每一个子系统都要有人参加评审;

(3) 统计修改、新加或删除的行数 (代码) 或页数 (文档), 根据一次评审会议 2 小时, 每小时

125 行代码或 6-8 页文档的约定, 计算所需的评审会议的次数。如果需要多于一次的评审会议, 作者将决定一次性的分发或以每一次会议为一单元分次分发会议资料, 若一次性分发则需要清楚的标明哪一部分资料是为哪一次会议准备的;

(4) 评审的主持人根据评审的开始条件检查评审资料, 决定是否可以进行评审;

(5) 评审的主持人根据评审的内容和评审小组的人员组成, 决定是否需要背景知识介绍。如果评审小组成员都对评审内容不熟悉, 则需要举行一个背景知识介绍会议来帮助大家了解相关知识以提高评审质量;

(6) 安排背景知识介绍会议或评审会议的时间并发出邀请信 (e_mail)。一般来说, 在背景知识介绍会议或发出评审资料 2--3 天后才能举行评审会议, 若要提前, 需征得所有参加者的同意;

(7) 以电子或书面的方式分发评审资料;

(8) 准备会议纪录表格;

(9) 评审的主持人要确认每一个评审小组成员都具有至少 24 小时的准备时间, 否则有权推迟评审会议。

3.4.2 背景知识介绍阶段

如果需要背景知识介绍, 作者将介绍相关的背景知识并提供所有的相关资料。

3.4.3 准备阶段

每一个评审小组成员, 根据各自的角色进

行如下准备:

作者:

- 收集所有与评审相关的文档, 如需求、设计说明书等;

- 准备检测其他参与者对评审资料理解程度的提问。

主持人:

- 理解评审资料并准备能够揭示缺陷和吸引参与者的提问;

- 检查与会者的准备情况, 如果需要, 可以推迟评审会议。

读者:

- 充分理解评审资料并能够清楚的向与会者进行解说;

- 准备向与会者逐行逐段地讲解评审资料。

记录者:

- 充分理解评审资料, 争取发现缺陷;

- 准备足够的记录表格。

测试者:

- 理解评审资料, 评价文档或源代码的可测试性。

其他参与者:

- 检查评审资料, 寻找缺陷, 检查此次修改是否与以前的文档一致。

3.4.4 会议阶段

评审主持人召集会议并询问每人的准备时间, 如果评审主持人认为准备时间不足以使与会者理解评审资料, 他/她将推迟会议并重新安排会议时间。如果准备时间是充足的, 记录者将在记录表中记录每人的准备时间。

之后, 读者将开始向与会者逐行逐段地讲解评审资料, 为便于大家的理解, 作者可以在需要时辅以必要的说明, 此处要特别注意的是: 所有的提问将只针对评审资料, 不针对作者本人。测试者也可以提出与测试相

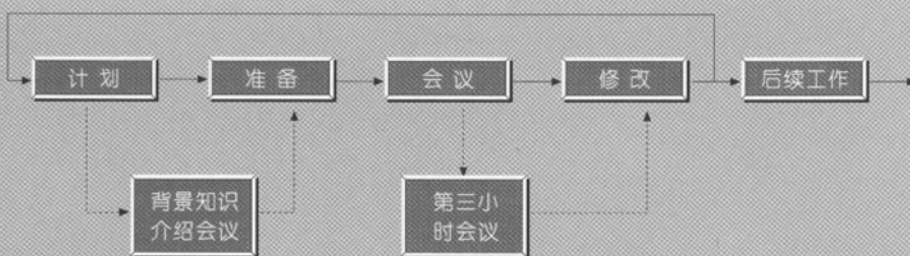


图 1 正式评审流程

关的问题。

对于每一个被发现的缺陷,将做以下处理:

- 评审小组判定引入的阶段;
- 评审小组判定缺陷的严重级别、类型和产生原因;
- 记录者将缺陷及相关信息记录在记录表上;
- 如果缺陷不易在本次改动中修复,提请一个新的改动请求。

为了使会议进行顺利,评审主持人要使会议做到:

- 会议的目的是发现缺陷而不是解决方案;
- 讨论的议题只限于专业,不要涉及个人;
- 每一个与会者都有机会发言;
- 记录者有充足的时间进行记录;
- 会议时间不要超过2小时。

如果2小时内不能结束会议,要另安排时间进行剩下的工作。根据发现的缺陷类型和严重程度,评审主持人与评审小组成员协商决定是否需再次评审。如果需要,评审主持人将以“需再次评审”为结论,结束本次评审。作者将在更改工作完成后,申请新的评审。

评审结论可分为:

- 通过:表明评审发现的缺陷可由作者自行改正,不需要评审主持人的审核,评审结束。
- 有条件通过:表明评审发现的缺陷可由作者自行改正,但必须由评审主持人审核后,评审才能结束。

• 需再次评审:所有缺陷将记录在案,此次评审结束,新的评审将遵循正常的流程。

• 取消:评审会议被取消,需要提供取消的原因,例如:人员不齐,发现了太多的缺陷,资料不全,没有足够的时间等等。

3.4.5 探讨解决缺陷方案阶段(第三小时会议)

在最后一次评审会议之后,如果需要,评审小组将讨论产生缺陷的原因和最佳的解决办法。同时,评审小组将讨论如何通过软件流程的改进来防止发生类似的缺陷;如果形成了对软件流程改进的建议,由评审主持人将此建议以书面的形式提交给软件流程控制部门进行进一步分析。

3.4.6 修改阶段

作者负责将发现的缺陷一一改正,未确认的问题找到答案。如果有在本次改动中无法解决的缺陷,作者负责提交一个新的改动请求。

3.4.7 后续工作

后续工作包括:

- (1) 作者与评审主持人共同复查修改情况,作者将介绍修改的内容,评审主持人将负责确认修改的正确性和全部的缺陷已被改正。
- (2) 如果不需要重新评审,评审主持人在确认所有评审记录已被录入到数据库后,以“接受”或“有条件接受”为结论结束本次评审;如果需要重新评审,评审主持人则以“需重新评审”为结论结束本次评审。

3.5 微型评审的流程

如果作者认为评审符合所有微型评审的入口条件,则应按以下流程进行微型评审:

(1) 作者选择另一个人作为审查者,审查者必须熟悉被修改的部分;

(2) 作者向审查者说明修改的内容,这项工作可以在计算机前或以书面形式进行;

(3) 如果在本次修改内发现了缺陷,作者将立即改正;如果在本次修改外发现了缺陷,作者根据情况的不同,必须:

- 如果缺陷与本次修改直接相关,作者将在本次修改中将其纠正,如果因此而使本次修改不能满足微型评审的入口条件,则中止微型评审,提请一个正式的评审;

- 如果缺陷与本次修改无直接关系,提请一个新的改动请求。

(4) 评审者将评审数据录入数据库并结束本次评审,如果本次评审因发现缺陷而被中止,结论为“需重新评审”,否则,结论为“接受”。

4 结束语

本文首先介绍了SW-CMM的分级结构,然后解析了同级评审关键过程域的作用和目标,最后根据同级评审关键过程域的要求,设计了同级评审关键过程域的软件管理流程,希望能起到抛砖引玉的作用,为提高我国的软件开发过程的标准化水平贡献一份力量。 ■

参考文献

- 1 Mark Ginsberg and Lauren Quinn, "Process Tailoring and the Software Capability Maturity Model," Software Engineering Institute, CMU/SEI-94-TR-024, November 1995.
- 2 Rita Hadden, "How Scalable are CMM Key Practices?" Crosstalk: The Journal of Defense Software Engineering, Vol.11, No.4, April 1998, pp. 18-20, 23.
- 3 Mark C. Paulk, "Questions and Answers on the CMM, Issue #1" Software Engineering Institute, 5 April 1994.
- 4 Mark C. Paulk, "Questions and Answers on the CMM, Issue #2" Software Engineering Institute, 2 August 1994.
- 5 杨一平等, 软件能力成熟度模型CMM方法及其应用, 人民邮电出版社, 2001.