

软件配置管理

摘要: 软件配置管理(SCM)是CMM第二级的一个关键过程域。它是软件过程的关键要素,是在开发和维护各个阶段管理软件演进过程的方法和规程。现在国内的很多程序员甚至项目经理对软件配置管理的基本概念并不清晰,也很少有项目用到软件配置管理。本文从软件配置管理的基本概念入手,逐步介绍软件配置管理的功能和活动,使读者对软件配置管理有一个整体的认识。

关键词: 软件配置管理 更改控制 CMM SCM

1 引言

如果你是一个软件项目经理,是否遇到这种问题:当软件产品要交付给客户时,原先可以正常运行的程序现在却发现了莫名其妙的错误。经过几天的浴血奋战,终于发现原来是由于程序员的失误,在重新编译的时候使用了过去有错误的旧程序。这是典型的没有应用配置管理而造成的问题。现在国内的很多程序员甚至项目经理对软件配置管理的基本概念并不十分清晰,也很少有项目用到软件配置管理。

实际上软件配置管理在整个软件生产过程中具有非常重要的地位。虽然采用软件配置管理会带来一定的开销,但不采用配置管理会引起许多问题而且使生产效率低下。在SEI的CMM中,软件配置管理是第二级(可重复级)的一个关键过程域。其目的是在整个项目的软件生命周期中,建立、维护软件项目产品的完整性。本文从软件配置管理的基本概念入手,逐步介绍软件配置管理的功能和活动,使读者对软件配置管理有一个整体的认识。

2 软件配置管理的定义

软件配置管理(SCM)是软件过程的关键要素,它是在开发和维护的各个阶段,管理软件演进过程的方法和规程。它控制软件产品的演进;保证其配置的完整性和可跟踪性。经典的配置管理定义在IEEE-STD-729中给出:配置管理是识别和定义系统中配置项的过程,它在这些配置项的生命周期中控制它们的变更,记录并报告配置项及变更需求的状态,检验配置项的完整性和正确性。

软件配置管理使整个软件产品演进过程处于一种透

明状态。开发人员、测试人员、项目管理者、质量保证组以及客户可以轻易地从配置管理中得到有用的信息。配置管理可以告诉他们:软件产品由什么组成,处于什么状态;对软件产品做了什么变更;谁做的变更;什么时间做的变更;为什么要做此变更。

3 软件配置管理的活动和功能

SCM涉及软件组织中每天的开发和维护工作。在软件产品的整个生命周期中软件的变更都要被识别、控制和管理。SCM的活动可以说十分广泛和琐碎。以下是几个在一般开发环境中关键的活动:

- 软件的存取和检索
- 在生命周期中完成变更
- 管理编译和构造过程
- 管理变更的分布
- 管理软件变更需求
- 各个组之间的沟通与协调
- 获得项目状态
- 跟踪 bug 并修正

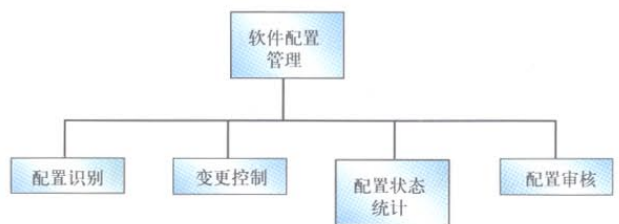


图1 配置管理的功能

SCM的活动可以归结为四个主要功能:配置识别、

变更控制、配置状态统计、配置审核(如图1)。

3.1 配置识别

配置识别包括识别软件系统的结构,唯一标识独立的组件并使其可以某种方式存取。配置识别的目的是获得在软件整个生命周期中识别其组件的能力。配置识别告诉我们系统中的配置是什么;系统的各个组件是什么;文件的版本是什么。

配置识别的主要活动是:选择配置项、制定配置项标识方案和存取方案。

配置项是配置管理的最小单元,它一般由一个或多个文件组成。软件组织可以根据不同的原则选择配置项,典型的配置项有:需求规格说明,模块源代码或对象源代码,设计和界面规格说明,测试计划和测试数据,软件质量保证计划,软件配置管理计划,软件工具等。

选择好配置项后就要为其选择适当的标识方案。配置项的标识使配置项被唯一识别,并且标识方案可以显示软件演进的层次结构。常用的方案是数字方案,即配置项由名称和数字版本号标识(如图2)。最左面的数字代表基线(基线的概念见1.2 变更控制)的版本,右边的数字代表由较小更改产生的版本号。

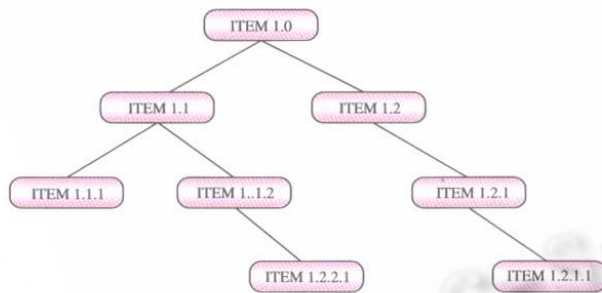


图2 数字标识方案

软件组织必须建立软件配置库,存放软件配置,并且随管理配置项的版本变化。这个配置库应使软件项目组的所有成员都可存取其中的配置项,但必须协调各成员之间的关系,使每个成员所能执行的权限不超过其应有的范围。例如:如果某个程序员只负责一个模块的开发那么他就不应该有对其他模块原码的修改权。

有效的配置识别是其他配置管理活动的前提,如果配置项和相关的配置文档没有被很好的确定,想要控制这些配置的变更、建立准确的记录和报告、审核配置的有效性是不可能的。不准确或不完整的配置项识别和配置文档

可能会导致有缺陷的产品、延期交付和很高的维护费用。

3.2 变更控制

变更控制是软件配置管理的核心,它通过创建产品基线,在产品的整个生命周期中控制它的发布和变更。变更控制的目的是建立一个帮助保证生产符合质量标准的软件和保证每个版本的软件包含所有必要的元素以及工作在同一版本中的各元素可以一起正常工作的机制。变更控制告诉我们控制什么;产品的变更是如何被控制的;谁来控制变更;变更在什么时候被接收、批准和检验的。

许多软件组织建立软件配置控制组(SCCB)控制配置项或整个系统的变更。这个组审查每个被提议的变更,对其进行批准或否认。如果批准变更,它与相关的组进行协调实施变更。

变更控制的另一个关键要素是基线的使用。基线是被正式审查和认可了的产品或规格说明。它是进一步开发的基础,对其进行变更必须通过正式的变更程序。如果某一产品被定为基线,那么它就被冻结,要想对它进行修改必须建立一个新的版本。

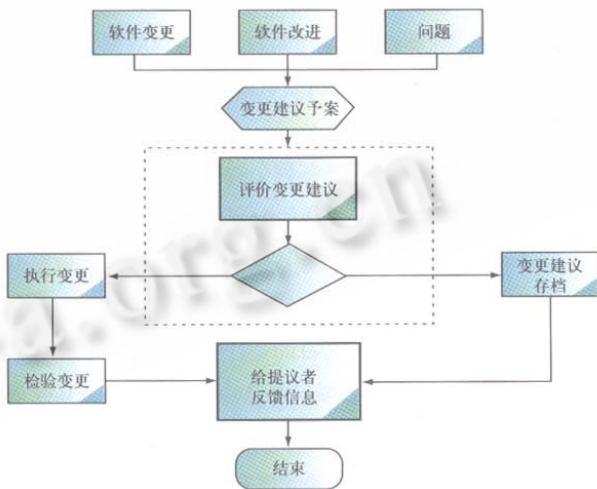


图3 变更控制过程

一种一般的变更控制过程如图3所示。由用户或开发人员提出的变更和改进要求形成变更建议预案提交给SCCB。SCCB对变更建议进行讨论评价。如果认为变更正确且可行,则批准其执行。变更被执行后,要对执行的结果进行检验。在执行变更和检验变更的过程中要将执行和检验的情况形成文档存于配置库中。检验通过后将变更信息反馈给提出变更要求的用户或开发者。至此变更结束。如果变更没有被批准,则要将变更建议存档并给提一

个反馈信息。

3.3 配置状态统计

配置状态统计涉及记录和报告变更过程的状态。其目的是为了持续地记录配置的状态以及保持基线产品和其变更建议的历史,并使相关人员了解配置和基线的状况。它包含在整个软件生命周期中对基线所有变更的可跟踪性报告。配置状态统计告诉我们配置项的状态是什么;变更要求是否被SCCB批准;配置项的什么版本执行了一个被批准的变更;新版本系统与旧的有什么不同;每个月查出了多少错误;有多少被改正了;问题的原因是什么。

项目和配置项的关键信息可通过状态统计传递给项目成员。软件工程师可以看到都做了哪些修改,或者每个文件都包含在哪个基线中。项目经理可以跟踪详细的问题报告和各種其他维护活动。最简单的报告应包括:事务日志、变更日志、配置项增量报告。其他典型的报告有:资源使用、所有配置项状态、过程中的变更。

3.4 配置审核

配置审核根据需求、标准或合同协议检验软件产品。配置审核告诉我们系统是否满足需求;是否对上一版本所有变更都已加入当前版本。配置审核的目的是检验所有的软件产品都已产生并且被正确的识别和描述,以及所有的变更要求可以根据确定的SCM过程和程序解决。

配置审核分为正式审核和非正式审核。在软件生命周期的关键阶段采取非正式的审核。如:再开始系统设计前,一般要进行配置审核,检验需求规格配置的完整性和正确性。在软件交付客户前采取正式审核。正式审核有功能型和物理型两种类型。功能型配置审核检验软件功能是否满足系统需求中定义的软件需求。也就是说,功能型配置审核根据需求验证系统。物理型配置审核确定软件产品和设计文档是否符合软件合同的要求。也就是说,物理型配置审核根据合同验证系统。

4 结束语

在企业中实施软件配置管理不仅仅要项目经理和配置管理人员熟悉软件配置管理,还应该对所有程序员进行SCM方面的培训。因为软件配置管理存在于软件开发和维护的整个过程之中。在软件的生命周期中程序员应该与配置管理人员积极配合,按照配置管理的规程开发和维护产品,这样才能很好的完成配置管理工作。■

参考文献

- 1 The Capability Maturity Model of Software Paulk,C.Mark
- 2 The Capability Maturity Model:Guideling for Improving the Software Process Mark C.Paulk Charles V.Weber Bill Curtis Mary Beth Chrissis.