

软件配置管理 - 团队开发的基石 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/186/2021_2022__E8_BD_AF_E4_BB_B6_E9_85_8D_E7_c41_186045.htm 1概述随着现代软件技术的发展，对于软件项目的需求日益复杂而且变更频繁，项目的开发模式已经由昔日的手工作坊式的开发转变为规模化、团队式的开发，当开发团队发展到一定规模时，会越来越强调开发过程规范化和成熟度。软件项目的成败在很大程度上取决于对其开发过程的控制，这包括对质量、源代码、进度、资金、人员等的控制。要进行有效的过程控制，仅仅依靠人的力量是不够的，还需要有相应的管理工具。"软件配置管理"是一套规范、高效的软件开发管理方法，同时也是提高软件质量的重要手段。软件配置管理可以帮助开发团队对软件开发过程进行有效的变更控制，高效地开发高质量的软件。在质量体系的诸多支持活动中，配置管理处处在支持活动的中心位置，它有机地把其它支持活动结合起来，形成一个整体，相互促进，相互影响，有力地保证了质量体系的实施。软件配置管理并不是一个新的概念，早在二十世纪70年代，加州大学圣巴巴拉分校（University of California at Santa Barbara）的Leon Presser 授就已经提出变更和配置控制的理论，并随着软件产业的逐渐壮大，在国外成熟的软件企业中逐渐得到重视和普及，在一些国外的大中型软件企业中，不仅设置专职的配置管理人员，而且有些公司还设有公司级的变更控制委员会（CCB：Change Control Board）对公司的软件配置管理工作进行统一的管理，配置管理对于软件开发的重要性由此可见一斑。国内的企业真正认识到配置管理的重要性，却是

近几年的事情，随着CMM（Capability Maturity Model）的概念和理论的普及，配置管理作为CMM2级的一个关键过程域，其重要性逐渐为人们认同。“进行管理和控制”意味着在给定时间（过去或现在）使用的工作产品的版本是已知的（即版本控制），而且以受控的方式引进变更（即变更控制）。如果希望有比“进行管理和控制”所蕴含的更高程度的控制，则工作产品可置于配置管理的完备的纪律之下

。CMM Level 2 KPA3其实在日常的开发中，我们已经在自觉或不自觉的进行着软件配置管理的工作。最简单的如使用软件开发集成环境提供的Backup功能，它可以使我们回溯到最近的修改；再进一步，就是使用文件夹或者压缩工具，把实现了一定功能的程序版本按日期进行备份，这也已经实践了配置管理的部分必要功能，但这仅仅是配置管理众多功能重的一小部分。真正的配置管理系统应该具有以下功能：并行开发支持：因开发和维护的原因，要求能够实现开发人员同时在一个软件模块上工作，同时对同一个代码部分作不同的修改，即使是跨地域分布的开发团队也能互不干扰，协同工作，而又不失去控制；修订版管理：跟踪每一个变更的创造者、时间和原因，从而加快问题和缺陷的确定；版本控制：能够简单、明确地重现软件系统的任何一个历史版本；产品发布管理：管理、计划软件的变更，与软件的发布计划、预先定制好的生命周期或相关的质量过程保持一致；项目经理能够随时清晰地了解项目的状态建立管理：基于软件存储库的版本控制功能，实现建立（build）过程自动化；过程控制：贯彻实施开发规范，包括访问权限控制、开发规则的实施等；变更请求管理：跟踪、管理开发过程中出现的缺陷

(Defect)、功能增强请求 (RFE) 或任务 (Task) ，加强沟通和协作，能够随时了解变更的状态；代码共享：提供良好的存储和访问机制，开发人员可以共享各自的开发资源。2缺乏配置管理造成的常见问题由于历史和现实的原因，配置管理流程和工具在我国软件行业中的应用并不普及，由于缺乏必要的配置管理流程和工具，很多软件企业在日常的开发工作中都会或多或少的遇到如下的问题：组织的知识和过程财富流失现代的社会竞争激烈，人员流动频繁，如果由于没有必要的配置管理流程和工具，大量的文档和代码等知识财富必然缺乏统一的管理，可能随意地保存在项目经理和软件工程师各自的机器里，往往会因为硬盘的故障或人员的离职而永远的消失，软件组织的数字财富就这样因为缺乏必要的配置管理而白白的流失不能及时了解项目的进展状况现代软件工程思想认为越早发现缺陷和风险，采取相应措施的代价越小。CMM的一个重要作用就是要提高软件开发过程中的可视性，使得问题能够被及时的发现。然而由于缺乏配置管理的流程和工具的支持，部门主管无法确切得知项目的进展情况，即便是项目经理也不知道各个开发人员的具体工作，项目进展随意性很大。所有的问题往往都会集中到项目里程碑时一起出现，这必然会造成巨大的开销，其结果往往是容忍部分缺陷存在或者延误开发周期。所有问题只能寄希望于最终实施时再解决，项目的实施工作因此变成了无法汇报、无法理清、无休止的维护。缺乏实现并行开发的手段在日常的开发工作中，经常会出现并行开发的需求，比如：对于一个项目可能要在开发新版本的同时继续对先前的版本进行必要的维护，或者针对某个特定的版本需要针对不同的客户同时进

行客户化的修改等等。在并行模式下，不同开发人员可以同时编辑修改某一文件，并行开发有可能产生冲突，但是却能够提高开发效率。如果没有配置管理工具的支持，进行并行开发将十分困难，单单通过人工操作，往往会造成修改过的bug重复出现或者几个人进行相同的工作，产生不必要的浪费。软件复用率低

软件复用是现代软件工程中的重要思想，是提高软件产品生产效率和质量的重要手段。软件产品是一个公司的宝贵财富，代码的可重用性是相当高的，如何建好知识库，用好知识库将对公司优质高效开发产品产生重大的影响。但如果没有良好的配置管理流程，软件复用的效率将大打折扣，比如对于复用的代码进行了必要的修改或改进，却只能通过手工的方式将发生的变更传递给所有复用该软件的项目，效率如何可想而知。另外由于缺乏进行沟通的必要手段，各个开发人员各自为政，编写的代码不仅风格迥异，而且编码和设计脱节，往往会导致开发大量重复的难以维护的代码。无法开展规范化的测试工作在传统的开发方式中，由于缺乏必要的配置管理和变更控制，测试工作只是人们的一种主观愿望，根本无法提出具体的测试要求，加之开发人员的遮丑，测试工作往往是走走过场，测试结果既无法考核又无法量化，当然就无法对以后的开发工作起指导作用。

对软件版本的发布缺乏有效的管理因为缺乏有效的管理手段，往往会在产品发布时却无法确定该版本所有的组件，或者向用户提供了错误的版本。对于特定客户出现的问题，无法重现其使用的版本，只能到用户的现场才能进行相应的调试工作。由于应用软件的特点，各个不同的客户会有不同的要求，开发人员要手工地保持多份不同的拷贝，即使是相同的

问题，但由于在不同地方提出，由不同人解决，其做法也不尽相同，程序的可维护性越来越差。这些都会延长实施的周期，同时意味着人力物力的浪费。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com