

提高软件开发的质量和效率

—基于 JBCM 系统的软件过程典型活动模式

国内大多数软件企业目前正处在成长阶段，在软件开发过程的选择上，需要考虑企业自身的具体情况。有条件的企业，应该按照 ISO/IEC12207 国际标准：“信息技术、软件生存周期过程”实施，在过程改善进度方面，往往选择 CMM 软件过程成熟度模型及其过程改进模式。

然而，这种标准的过程在实施中有大量的文档要求和严格的过程执行纪律，可能会使软件人员感到负担很重，不一定能够适应许多情况下快节奏的软件开发要求。

另外，为了实施这种标准的过程改进，企业要专门投入人力、资金和时间，并且可能要忍受一个难以避免的过度期，在此过度期中不能立竿见影地收到过程改进的成效。那么，是不是说，对我们某些暂时没有条件进行标准过程改进的企业来说，过程改进就无法实施了呢？回答是否定的！过程改进的意义在于提高软件开发的质量和效率，任何有利于提高质量和效率的措施都是在进行过程改进。因此，软件企业完全可以根据自身的具体情况，提出切实可行的过程改进目标，采取在当前条件下实际可操作的措施，来改进软件过程。下面，我们提出一种基于配置管理的软件过程典型活动模式，该模式可以运用在任何规范或不够规范的软件过程中，通过配置管理系统的具体支持，改进现有过程中关键环节的典型活动，从而达到过程改进的目标。

无论是采用 ISO/IEC12207 国际标准或 CMM 过程改进模式，还是在现有条件下进行过程改进，都是对过程中活动的安排或重新安排，为了突出软件过程活动的重点，有必要对软件过程的关键环节和典型活动进行分析。

1. 软件过程的关键环节和典型活动分析

ISO/IEC12207 软件过程结构提出三类过程，分别为基本过程、支持过程和组织过程，这三类过程又包含了 17 个具体的过程。其中，基本过程中的开发过程是活动最密集的过程，企业的主要的人力、资金和时间的相当部分都投入这一过程，因为它是直接生产软件产品的过程，产品的质量和效率与该过程的活动密切相关，当然，其它过程也是与开发过程相协调，以形成一个完整的软件过程结构。

对于当前条件下的软件企业来说，可能在其它过程活动的选择上各不相同，但开发过程则大同小异，无论采用什么样的软件生命周期模型，都要进行计划、需求获取、分析、设计、编码、测试、交付和维护等活动，而其它的任何活动，都是直接或间接地保证开发过程能够高质量高效率地进行。因此，开发过程活动及与开发直接相关的管理活动是过程改进的关键活动，应该关注这些活动中的关键环节。

分析软件过程中的关键环节，必须站在一定的角度，从不同的层次进行观察，根据软件企业一般的结构，可以从三个角度进行分析，分别为企业管理者的角度、项目负责人的角度和开发人员的角度。如果在这三个方面的过程活动都得到强有力的支持，则企业的软件过程能力必然会有显著提高，软件的质量和效率也能得到有效的保证。

2. 基于 JBCM 的软件过程典型活动模式

JBCM 是北京大学软件工程国家工程研究中心、北京北大青鸟软件工程有限公司研发的一个基于构件的软件配置管理系统，提供对企业软件资源的组织和使用的支持，特别是从软件

的构造性和演化性出发，全面支持开发过程管理。为了使软件企业能快速地达到用好配置管理系统，改善软件过程的目的，下面总结了基于该系统的典型软件开发管理模式。这里所说的模式，是针对软件过程管理中的一项具体任务或要解决的一个具体问题，定义的一个可操作的活动序列，活动中特别的人员角色都已指明。

表-1 基于 JBCM 的编码人员活动模式

| 标识 | 模式 | 活动 | 人员 |
|-----|--------|---|------|
| 1.1 | 代码重用模式 | 通过构件标识和注释找到构件； 通过版本标识、注释或版本比较找到版本； 从配置库中获取构件指定版本； 获取版本文件中需要重用的信息。 | 编码人员 |
| 1.2 | 共享协作模式 | 二人按共享写方式分别检出同一构件的同一文件； 二人在各自的工作区内对构件内容进行更新； 一人检入更新后的构件新版本； 另一人在系统提供的合并注释功能支持下进行再更新后检入构件的另一新版本。 | 编码人员 |
| 1.3 | 分支协作模式 | 二人明确对同一构件版本协作开发的各自任务； 二人从同一版本出发进入不同分支（需要时创建）； 二人在各自的分支上开发构件新的版本（一至多个）； 二人开发结束后，通过比较版本差异确认冲突； 二人共同解决冲突合并不同分支版本。 | 编码人员 |
| 1.4 | 重用提取模式 | 获取指定构件的可重用版本； 对版本进行接口标准化改造，以及必要的逻辑重组； 建立并标识新的可重用构件及检入第一个版本； 根据需要对可重用构件进行扩充； | 编码人员 |
| 1.5 | 构件组装模式 | 根据系统需求，在设计过程中规划对已有构件的重用方式； 根据构件标识和注释查找可重用的构件； 获取可重用的构件，进行功能和接口测试； 根据需要对构件功能和接口进行扩充，并生成重用构件的新版本； 根据设计方案开发新的构件； 利用配置功能构造系统的结构； | 编码人员 |
| 1.6 | 程序员组模式 | 构件管理员确定程序员组成员； 构件管理员设计构件结构； 构件管理员向组员分配任务； 组员利用共享和分支功能并行开发； 构件管理员组织开发过程中的交流； 构件管理员采用文档、代码审查及测试等手段确认构件质量。 | 编码人员 |

| | | | |
|-----|--------|---|------|
| 1.7 | 开发回滚模式 | 对准备放弃的构件最新版本进行标识； 通过构件历史版本选择和比较确定合适的版本； 对该版本进行分支，并在新的分支上进行继续开发，或者，在原分支上先检出，再检入新版本并进行标识。 | 编码人员 |
|-----|--------|---|------|

表-2 基于 JBCM 的项目负责人活动模式

| 标识 | 模式 | 活动 | 人员 |
|-----|---------------|--|-------|
| 2.1 | 资源组织模式 | 项目管理员建立项目资源包 在项目资源包中，建立需求构件和概要设计构件； 完成初始阶段需求分析和软件体系结构概要设计； 根据概要设计建立项目下的子项目-构件结构并确定子项目-构件管理员； 根据概要设计建立软件的初始结构配置； | 项目管理者 |
| 2.2 | 责任分配模式 | 根据开发活动需要，项目管理员重新指定子项目管理员； 根据开发活动需要，项目管理员重新指定构件管理员； 根据开发活动需要，项目或构件管理员重新调整用户权限； | 项目管理者 |
| 2.3 | 系统结构模式 | 建立：/计划/需求/分析/设计/源代码/执行体/用户文档/测试/安装/等子项目-构件结构； 建立源代码结构中的系统构造结构配置； 建立执行体结构中的系统组装结构配置； 建立提交配置，包括需求、分析、设计、系统组装结构配置； 建立测试配置：包括提交配置和测试结构； 建立产品配置：包括测试配置和用户文档结构； | 项目管理者 |
| 2.4 | 计划控制模式 | 根据系统构造结构配置的层次结构，按照计划进度，定期导出配置最新版本； 对导出版本进行构造、测试和质量审查； 若审查不通过，则反馈信息调控下一步的开发活动； 对审查通过的配置建立构造基线； 按照构造基线，对系统或部分系统进行构造，将结果检入可执行体结构； | 项目管理者 |
| 2.5 | workflow 支持模式 | 1、开发负责人组织开发活动并建立构造基线； 2、构造负责人导出构造基线，构造可执行体，检入可执行体； 3、安装负责人导出可执行体，制作安装系统，检入安装系统并建立组装基线； 4、项目负责人建立提交基线； 5、测试负责人导出提交基线，进行测试，提交测 | 项目管理者 |

| | | | |
|-----|---------|---|-------|
| | | 试报告，建立测试基线。 6、评审负责人导出测试基线，评审测试结果，若不通过则转 1； 7、发布负责人确定是否发布，若是，则建立产品基线。 | |
| 2.6 | 工作量审查模式 | 通过构件版本视图审查构件小组成员以版本为单位的工作量； 通过构件两个版本（不一定连续）比较，检查构件小组成员在一定阶段的代码工作量（参考）； 通过查询操作，检查指定用户的工作情况记录； | 项目管理者 |
| 2.7 | 质量审查模式 | 获取需求、设计、代码的对应版本，进行内容检查和一致性检查； 通过某一阶段的所有构造基线相对于计划的偏离程度估计开发活动进展情况； 通过某一阶段的组装基线相对于计划的偏离程度，估计组装活动的进展情况； 通过某一阶段的提交基线相对于计划的偏离程度，估计提交活动的进展情况； 通过某一阶段的测试基线相对于计划的偏离程度，估计测试活动的进展情况； 通过某一阶段的产品基线相对于计划的偏离程度，估计发布活动的进展情况； | 项目管理者 |
| 2.8 | 任务移交模式 | 针对任务确定需要移交的资源内容； 改变移交资源的用户访问权限； 接收者导出资源，与移交者共同确认资源的完整性和正确性（必要时对文档进行审查，并对代码进行构造和测试，评测结果也可作为对移交者的一种评价）； 接收者根据任务重组资源。 | 项目管理者 |

表-3 基于 JBCM 的企业管理人员活动模式

| 标识 | 模式 | 活动 | 人员 |
|-----|--------|---|-------|
| 3.1 | 管理层次模式 | 根据企业行政管理结构，建立项目-子项目结构； 根据企业配置管理职责分配，设立系统配置管理员； 系统配置管理员根据访问企业软件资源的需要，设立配置管理系统用户并分配职责； 各级项目-子项目管理员根据企业部门划分，建立项目，确定项目用户及权限，并根据部门规模和需要指定子项目管理员； 负责软件项目开发的项目-子项目管理员进行软件项目资源管理。 | 企业管理者 |
| 3.2 | 信息发布模式 | 发布者建立一个构件； 发布者将接收者全部指定为构件的只读用户； | 企业管理者 |

| | | | |
|-----|---------|--|-------|
| | | 发布者定期更新构件内容； 各接收者定期导出构件内容获取发布信息； 发布者通过对构件的审计活动获得各接收者是否按时获取信息。 | |
| 3.3 | 信息集中模式 | 接收者建立一个项目（子项目）； 接收者为每个信息提交者建立一个构件，并指定该提交者为构件一般用户； 接收者建立一个配置包含全部构件； 提交者按期更新各自的构件内容； 接收者按期导出配置的最新版本获取全部提交者的最新信息； 必要时，接收者建立配置的基线，标识特殊的信息集合。 | 企业管理者 |
| 3.4 | 活动审计模式 | 系统管理员通过审计查询指定用户对资源的操作活动情况； 系统管理员通过不同条件的查询获取所需的资源情况； | 企业管理者 |
| 3.5 | 报告分析模式 | 系统管理员通过生成报告，获取企业资源总体状况； 可将报告或其某一部分直接在企业网站上公布，做为企业内部信息发布的一部分（例如：公布配置活动执行情况）； 通过报告中各级资源相关的活动日志，审查企业人员的活动； 通过报告中配置和基线情况，审查相关项目的进展情况； 通过报告中版本信息，获取开发活动中资源的被使用程度； | 企业管理者 |
| 3.6 | 里程碑控制模式 | 通过检查系统-子系统对应项目下的配置基线，获得项目进展过程中里程碑记录； 导出指定里程碑对应的基线的内容，可对里程碑的资源状态进行审查； 通过实际里程碑与计划的比较，获得实际活动偏离计划的程度，并通过对以下各级里程碑的检查，获取主要影响进度的活动，并以此为依据调整计划或采取控制措施。 | 企业管理者 |

以上活动模式只反映了 JBCM 系统在实际应用中一部分的经验总结，软件企业根据自身的软件过程特点、管理和开发人员素质、过程改进的步骤，可以对这些模式进行改造，也可以根据经验扩充新的活动模式，以适应企业软件过程改进的需要。

JBCM 产品部.张晨东
2002 年 10 月

联系方式：Marketing@jbcase.com