

应用软件行业企业质量管理核心过程实施指南

本指南是基于江苏省南京市雨花台区信息技术行业的小微企业质量管理实际情况编写,用于解决小微企业质量管理中的“痛点”、“难点”问题,指导小微企业质量管理提升,也可供其他地区类似小微企业参考、借鉴使用。

ISO9001质量管理体系虽适用于所有组织,具有通用性,目前通过质量管理体系认证小微软件企业不在少数,但由于ISO9001标准缺少针对软件行业具体实施的方法,通过认证的软件企业同时存在文件化体系不健全、组织结构不适应企业发展、质量管控过程不符合实际、生搬硬套等问题,影响了质量管理体系的有效运行,

根据应用软件行业小微企业特点,针对上述问题,结合ISO9001标准和CMMI要求,给出了质量管理控制指南,供应用软件行业小微企业参考。

一、需求分析过程管理

(一) 典型问题

多数软件企业缺少明晰的需求分析过程,习惯把用户的需求直接作为软件需求,在开发的过程中没有设立需求建模和抽象化的过程,也没有考虑性能、安全、易用性、可维护性和扩展性等非功能性需求,需求的表达不够结构化,导致对需求一致性的理解产生偏差,企业需求分析人员能力欠缺,最终影响软件产品质量。

（二）控制要点

需求阶段的工作在开发过程中是至关重要的，它明确了软件产品要实现的每个功能点和开发内容。需求阶段主要工作有需求收集、需求分析、需求描述和需求验证。需求收集主要是收集所确定的功能点，需求分析过程中，应充分讨论需求的可实现性，包括各个功能点的接口、约束条件、相应标准和限制等，分析后将所有的需求功能点描述在文档中，建议以《需求说明书》的形式展现，在完成需求分析后，应进行评审验证。所有已确定的需求应记录在《需求说明书》中，对于基线化后的需求不得随意修改，所有变更须经评审，未经基线化时不得进入下个阶段的开发工作。

（三）实施指南

需求管理就是软件项目的范围管理，是整个项目的源头，项目的估算、计划、后续的跟踪控制、验证和确认等都和需求相关。需求管理保证了项目的进度、质量和成本目标的实现，以及项目计划的严肃性和可执行性。

企业应对收集到的用户需求进行分类和优化，采用软件工程的语言结构和文档对用户需求和产品需求进行描述。需求收集要能很好地描述现状，了解用户的期望，还需考虑需求复用，可扩展和配置等方面的问题。应对需求进行动态行为分析和静态数据分析，并在需求文档和总体设计方案文档中进行体现。

需求分析阶段还有一个重点输出就是原型和DEMO，通过原

型和DEMO将理解后的想法更加形象地表达给用户，会减少后期需求变更引起的返工时间，软件原型是降低需求变更风险的有效方法。

（四）检查改进

利用需求跟踪矩阵对需求分析进行检查与改进，可在需求跟踪矩阵中记录每个需求的相关属性，明确每个需求的关键信息，需求跟踪矩阵中记录的典型属性包括唯一标识、需求的文字描述、收录该需求的理由、所有者、来源、优先级别、版本、当前状态（如进行中、已取消、已推迟、新增加、已批准、被分配和已完成）和状态日期等相关内容。为确保相关方满意，也可能需要增加如稳定性、复杂性和验收标准待补充属性。

二、软件开发过程管理

（一）典型问题

软件开发过程中质量管控不充分，缺少对编码规范、需求变更，代码审查等方面的有效控制，造成软件上线后系统BUG出现频繁，维护困难、改造升级成本大等问题，导致软件维护成本和交接成本的增加。

（二）控制要点

应对软件设计全生命周期进行质量管理和控制，主要在数据质量、编程质量、文档质量等三个方面进行管理和控制。在项目开始应充分了解项目目标、交付和范围，编写项目概要设计，加强需求变更控制，同时邀请业务人员参与设计过程，保证软件的

业务架构，通过人工比对、程序比对、统计分析等方法保证数据质量的完整性。

（三）实施指南

在概要设计中要根据需求说明书中描述的各个功能点进行分解，同时按照功能模块来组织分析。在概要设计中，需要描述模块的数据结构、模块间的消息接口定义、系统所有的全局变量、每个处理的详细过程和相关的流程图等内容，建议输出《概要设计说明书》，所有的数据结构、消息内容和接口定义都应在《概要设计说明书》中描述，该过程应充分分析模块间的耦合性和接口一致性，对《概要设计说明书》进行评审并将其基线化，对于基线化后的所有过程实现、消息内容、数据结构和接口定义不得随意修改或变更，所有变更须经评审或交叉验证。在未经过基线化时，不应该进入下个阶段的开发工作。

详细设计是对概要设计描述的过程进行细化，由于项目时间一般都较紧，一般企业会将概要设计和详细设计合并进行，在概要设计中将模块设计、消息描述、接口定义和函数实现等内容一并描述，但这种方式可能对后期单元测试和集成测试产生一定的影响，使单元测试的效果比较差，导致软件质量低下。所以要关注函数的处理逻辑描述是否清楚，以确保测试用例有较强的针对性，保证测试有较好的效果。建议输出《详细设计说明书》文档，在详细设计时可根据实际情况将概要设计和详细设计合并进行，但是单元测试的用例设计要加强管理，避免单元测试的效果低劣。

企业应制定《软件编码规范》，并按照编码规范进行编码，编码规范应能对代码的健壮性和良好维护性提供帮助。可行时，应每天进行代码审查，目的是检查出代码实现上和接口上出现的问题。在编码过程中，如果发现系统设计或接口定义存在问题，应提出变更需求，经评审后进行修改，不得擅自修改设计或接口定义。

（四）检查改进

软件开发过程的质量管理从质量保证计划开始。在项目启动之前，应制定详细的《软件质量保证计划》，为后期的质量管理和控制做出规划，有效地提供质量保证。

质量保证计划主要包括质量要素和质量目标，技术评审计划、软件测试计划、质量保证计划、缺陷（问题）跟踪工具等要求，以及可能影响软件产品质量的技术要点。质量保证计划应明确项目和软件产品质量的可接受水平，描述如何确保可交付成果和过程能够达到这一质量水平，质量保证计划还应描述不合格的处理方式以及需采取的纠正措施。编制质量保证计划通常采用流程图、因果分析图等方法进行分析，确定需要监控的关键元素，设置合理的关键节点，质量保证计划的完整性应得到保证。

应在项目启动阶段，通过相应的配置工具软件，根据分类方法，对目录结构进行策划，建立相应的配置目录结构，设定目录访问和存取权限。每个具体的配置项，应标识作者、时间、版本号、状态等信息，对版本进行实时监控。

建立配置库后，应制订适用的配置流程，应对项目配置情况进行分析，定期提供配置报告，发布最新配置项，提出改进建议并跟踪执行情况，避免出现因文档或代码版本不一致导致质量故障发生。为保证配置项的可靠性，应制订备份策略，定期进行备份，避免因文件损坏导致配置项丢失的情况出现。

三、软件测试过程管理

（一）典型问题

代码审查过程中敷衍了事，编码人员不进行交叉审查，审查时也未使用相关工具对代码进行审查，使得软件中存在BUG，致使上一环节的问题被带入到下一个环节。软件测试流程不合理/不充分，在产品交付给用户使用后，问题会暴露出来，造成公司成本大幅增加。测试人员对典型软件缺陷分析不足，对开发难以发挥指导作用。测试骨干力量不足，各部门专兼职测试人员流动性大，测试质量大打折扣。

（二）控制要点

采取对源代码逻辑、属性、命名规范、代码布局等检查，避免复制代码，坚持设计回溯，对冗余代码及时重构，保证代码和软件外观风格的统一。通过黑盒测试、灰盒测试、白盒测试等确保代码质量，利用阶段集成，自动化测试等增强缺陷发现能力，建立缺陷跟踪流程。对于开发过程中产生的各种文档，制定文档规范，统一文档语法、语义和逻辑要求，确保软件的完整性、设计合理性、过程可追溯性。

（三）实施指南

单元测试是白盒模式，是通过设计具有针对性的测试用例，对程序所有的逻辑路径进行测试，测试用例的质量决定了单元测试的效果。通过单元测试可以有效发现存在的逻辑错误和功能错误，同时单元测试的所有测试用例也可以在集成测试和维护阶段进行回归测试，保证后期代码修改的正确性。建议使用合适的单元测试工具，提高测试效率，同时有利于后期的回归测试。

集成测试是一种灰盒测试，也可以当成黑盒测试进行，是通过测试用例，对模块中的接口进行测试。一般是在单元测试结束后进行，对模块间的接口进行测试，集成测试应关注模块间消息接口测试情况，注意不要与单元测试混淆，否则难以有效发现接口上的问题。

系统测试是对系统功能的验证，是根据需求设计文档中对描述的各个功能点进行验证，理论上系统测试进行是比较快的，发现问题也是比较少的。但是一般情况下，由于在开发中前期各个阶段工作的不完善、不规范，系统测试反而发现的缺陷最多。所以应在修复系统测试中发现的缺陷后，对所有的单元测试用例进行回归测试，防止由于修改引出新的缺陷。

（四）检查改进

软件数据质量反映了软件产品的质量，数据质量可采用人工或程序进行比对，对转换前后的数据进行直接的比对，或通过统计分析的方法，对新旧数据转换的正确程度进行量化的分析，发

现数据或者统计结果的不一致性，并采取纠正措施。

应检查源代码的逻辑、属性、命名标准、代码布局等内容，代码的编译、链接、集成和构建应进行验证和确认。

应使用开发工具所带的编译功能或专门程序对软件源代码进行检查，发现源代码存在的问题。也可通过人工检查判断是否符合所制定的编码规范。制定编程规范，关注源代码是否无矛盾或遗漏、无定义变量、变量无赋值、死循环等情况。应通过人工或软件检查判断如内存管理、任务并行性、关键算法、接口对性能的影响、模块间通信等。

应开展软件质量评估工作，评估应包括对项目各个关键点的质量评估，以客户满意度为主要指标，包含软件质量特性的内容，即使用性、可测试性、正确性、维护性、可靠性、移植性、效率、重用性、完整性、互操作性和适应性等。

应确定评估范围和目标，制定评估计划，进行问卷调查并进行分析汇总，分析归纳所发现的弱点，将发现的弱点进行修正。

测试机制应包括丰富的软件测试经验、强大的测试工具、优秀的测试管理水平等。应组建测试团队，使用完整的自动化软件测试工具，完成全方位的软件质量验证。软件测试应贯穿于整个软件产品生命周期，软件测试要经历测试计划、测试用例的设计和实现，可选择合适的测试工具进行单元测试、功能测试。对于小微软件企业来说，可将研发出来的软件产品交给第三方专业测试公司，在提高软件产品质量的同时，也节约了产品的测试成本。

对于质量保证计划中设置的关键节点，应按规定及时进行检测与检查，确定项目成果是否符合相关的质量标准，同时进行趋势分析，对一些偏向于不合格的趋势及早进行控制。质量分析时，对于已发现的不合格或潜在不合格，应制定相应的纠正措施，对质量保证计划进行相应的调整，保证项目的顺利实施。