开发质量控制

为了健全和完善武汉凯迪信息化系统设计开发的质量管理体系，促进质量管理活动系统化、规范化，以确保所交付的武汉凯迪信息化系统能够满足规定的各项具体需求，项目中需要进行开发质量控制。

* 1. **项目各个阶段中的开发质量控制**

我们小组在项目各个阶段（需求分析阶段、设计阶段、编码阶段、测试阶段、工程实施阶段）中采取了一系列的措施对开发质量进行控制，确保项目的质量。

* + 1. **需求分析阶段**

信息系统需求分析为整个信息系统的开发指明方向和奠定基础。据分析，信息系统中有15%的错误起源于错误的需求。需求分析不明确，会引发开发中的一系列更改。这些更改导致浪费大量资源、信息系统开发无法按时完成等严重问题。因此，信息系统需求分析的质量对整个信息系统开发的质量与成本有重要影响，它是关乎信息系统开发成败的重要因素，研究需求分析的质量控制方法显得十分必要。

在该阶段要确定用户所提出的各种功能性或非功能性的需求，并完成需求规格需求说明以及各类文档。至此期间，可能还需要组内组织进行对需求的评审工作，在此落实项目的范围，对用户进行反馈，对于确实难以实现的功能，可以和用户协商分批实现或者变通解决。特别要注意的是，多与客户进行交流，并且在交流的时候尽量使用用户专业领域的语音来描述需求，少使用计算机领域的术语，以便能够更好的达成共识，有利于接下来项目工作的进行。同时，把有关的确定好的文档，例如需求规格说明书等文档存入资料库。各成员每天先更新SVN至最新版本，然后再进行其他操作。

* + 1. **设计阶段**

在此阶段，小组成员需要就功能设计，界面设计以及数据库设计等方面做出决断。为提高项目设计质量，应做到以下几点：加强工程设计的标准化工作；严把设计方案的选择与审核关；做好设计接口控制；建立设计成果校审制度；建立设计文件的会签制度；鼓励设计人员进行设计创新；认真进行设计文档交底与会审；做好工程设计变更控制。我们小组成员在设计阶段，由质量控制小组严格把控质量控制关，如采取建立以上措施的部分措施，并且召集项目小组成员召开相关会议对设计方案进行商讨。

* + 1. **编码阶段**

在该阶段，小组成员按要求完成所分配的代码相关的工作，组长做好管理工作，时刻秉承制定的项目开发计划和目标，把开发的进度及时和用户沟通，调动起开发团队的积极性，对工作结果和计划进行检查，要求所有成员按时完成周报，日报，并上传相关的报告，完成的代码或其他形式的成果上传到SVN个人文件夹内，时刻了解和掌握各项目成员的工作状态，及时发现，掌握项目中的各种问题，包括进度，技术以及人员对项目开发的一些看法等，并且及时解决为。使用版本控制工具，以及严格版本控制规程。

同时，尤其要注意的是源程序文档化，数据说明标准化，语句结构简单化，输入输出规范化。建立编码标准，对项目编码人员进行相关培训，建立review机制确保提交到SVN中心库的代码符合编码规范。

* + 1. **测试阶段**

在该阶段，制定测试计划，协调成员对各个测试用例进行测试，并最终验收测试报告，要求把测试用例设计，测试报告，测试工具源代码及软件，测试通过的软件上传到SVN指定文件夹内。

* + 1. **项目实施阶段**

在项目实施阶段，验收实施计划、实施报告、用户使用手册、用户培训资料、用户培训记录以及软件问题反馈表等，并将最终确定的文档存入SVN指定文件夹找。

* 1. **项目中用到的一些措施**
     1. **软件产品工程**

软件产品工程的目的是坚定地实施界限清晰的，结合所有软件工程活动的工程过程，以高效准确地生产正确的、一致的软件产品。

软件产品工程专注于执行通过使用项目中已经定义的软件过程和恰当的方法和工具来构建并处理软件工程任务。通用软件研制采用软件工程方法，自软件需求分析开始，经设计、编码、测试、集成、试验、验收直至交付止。软件产品开发涉及系统创意、构思、策划和论证等工作,软件开发完成后须通过试验和验收方可交付使用。

文档（例如软件需求规格说明书，软件设计文档，测试计划和测试过程）展示了软件工程过程，这些文档被编写和审查以确保每一项任务能够说明之前的任务对紧接着的任务（包括这些任务的执行和维护软件）是合适的。当变更被认可时，受影响的软件产品、计划、承诺、过程和活动能够被修改以适应被认可的变更。

* + 1. **组间协调**

组间协调的目的是建立一种针对软件工程小组积极参与到另一个工程小组中，以使得工程能够更好的有效且高效地满足客户需求。

组间协调专注于软件工程小组在其他软件工程小组的参与以处理系统级的需求、目的和争议。软件工程小组的代表参与到建立系统级的需求、目的和计划通过与尽可能地客户、用户共同协作。这些需求、目的和计划成为了所有工程活动的基础。

策划和管理组间的技术性交流和互动以确保整个系统的质量。技术性的审查和交互通常由工程组的代表门主导以确保所有的工程小组能够意识到所有项目组的状态和计划，也确保系统级的和组间级的争议被妥善处理。

与工程任务相关的软件的细节在需求管理和软件产品工程这一关键过程域中有描述。

* + 1. **同行评审**

同行评审的目的是尽早高效地将不合格品从软件工作产品中移除。一个重要的必然性是能够开发出更容易被理解的软件工作产品，而且不良品能够被避免。

关键过程域覆盖了执行同行评审的举措。这些时间定义了一个特定的被同行审查过了的软件产品被包含在描述开发和维护每一个软件产品的关键过程域中。

* + 1. **软件质量管理**

软件质量保证管理对于提高软件工程和管理是非常必要的，因此构成了软件工程过程的一部分。FLUIDS项目中采用的特定方法建立在能力成熟度模型之上。这一评价过程和改进计划的方法论是专用于软件处理单元和软件过程。能力成熟度模型定义了五个级别的过程成熟度。过程成熟度又被划分为若干个关键过程域。每个关键过程域描述了关键实践过程，并且有利于时间这些目标。软件成熟度模型更佳的专业，并且与通用质量标准ISO 9000并不冲突，也同样被纳入欧盟的评价方法和认证过程——BOOTSTRAP项目中。

接下来的部分提简要介绍了对于提供了项目级可用性措施的那些关键过程域。在FLUID项目中，这些关键措施将引导软件质量控制步骤。这些信息已经被从能力成熟度模型的一般性文档和该关键实践的说明中提取。

* + 1. **缺陷预防**

毫无疑问，对待缺陷最佳的举措就是消除它。然而，这可能太过理想化，在现有的科技条件下很难实现。与此同时，开发者们需要快速查找缺陷并且尽可能小的消除它的影响的策略。定义和实现最佳缺陷预防的技术（这是最佳软件开发过程的一个大部分）可能在缺陷管理计划有着较高的优先级。

缺陷的预防应该开始于系统的临界风险评估。获取关键性的风险允许人们了解到那些最可能发生的和对系统具有重大影响的风险类型。然后，可以找出预防风险的策略。预防风险的步骤主要有以下几个步骤：

1. 识别关键风险
2. 估计预期的影响
3. 最小化预期影响
   1. **软件质量保证SQA中的7个方面的要求详细解释**
      1. **定义恰当的质量目标**

软件最终是要交付给用户使用的，因此应从用户的角度来定义软件质量目标，软件应满足用户的业务需求，实现令人满意的用户体验。这样做的好处：既不将质量目标定得太高，任由你付出百般努力也无法实现，也不将目标定的过低，那样你无法给用户交差，根据时间，资源和预算客观情况定义合适的软件质量标准最好，既不让开发团队感觉痛苦，又能让用户满意。

* + 1. **让每个人都知道质量的重要性**

尽量在软件开发生命周期的前段时间减少软件缺陷，避免在后期来消灭缺陷，那样耗费的时间和精力更多。好处：让每个人都知道质量的重要性后，他们就会从心理上更注重代码质量，就会更用心写出高质量的软件。

* + 1. **调整团队个人的目标，纳入质量考核体系**

根据业务需求调整团队和个人的工作目标，并纳入质量考核体系，实施严格的奖惩措施，刺激开发人员的工作效率和工作质量。好处：根据团队成员的执行表现给予适当奖励，让他们知道改善软件质量是一种奋斗目标，逐渐发展成为一种习惯。

* + 1. **获取正确的需求**

确保从需求获取开始，项目就朝正确的方向迈进，需求偏离或需求错误是让开发人员最头痛的事，大量的返工和修改会熄灭本已燃起的激情，而正确的需求会给开发人员带来愉快的心情。好处：减少返工和重新测试周期，减少总体工作量。

* + 1. **将测试重点放在最关键和风险很高的点**

在时间有限的情况下，不可能将方方面面的缺陷通过测试全部暴露出来，这时只有抓住重点，做到有的放矢，将核心功能点重点测试，避免重大缺陷成为漏网之鱼。好处：杜绝关键缺陷，即便有其它缺陷未被发现，也不至于影响到软件的整体质量。

* + 1. **提高设计质量**

开发人员会根据架构师的设计文档进行编码的，如果设计描述得含混不清，那开发人员可能会根据自己的理解编写代码，或许就会造成南辕北辙的结果。好处：参照简明清晰的设计编写出来的代码也会更简单，更干净，也更容易测试和返工，代码中包含的错误也会更少，也更容易诊断和修复缺陷。

* + 1. **合理使用自动化测试工具**

传统的手工测试很难覆盖软件的全部功能点，某些后台功能只能借助工具来测试，此外，手工测试的效率低，反复单调的测试更是对测试人员心理素质的极大考验，容易造成对测试工作的懈怠，降低测试质量。好处：通过自动化测试工具的合理使用，可以缩短测试周期，提高测试的可重复性。