

# 如何保证软件设计的质量

学号：14126080

姓名：丁泽宇

软件质量是软件的生命，其重要性越来越为人们所认识。近 10 年来，软件质量研究一直是软件研究发展较快的新方向、新问题。这是因为在软件实践积累了经验，需要加以概括，总结为知识，抽象为科学。另一方面，人们一些实践活动失败，乃软件质量低劣所至，这迫使人们不得不进入这一领域，研究软件质量概念和模型，研究影响软件质量的因素及软件质量控制方法。

软件质量是一复杂系统工程问题，它必需用系统方法来研究。软件质量问题主要来源于软件的开发过程，而影响开发过程质量的主要特性包括：开发所采用的技术、开发者个人的业务经历水平及开发所使用的工具，常见影响软件质量的因素归纳总结有以下几个方面：

(1) 需求分析因素。对用户的提出的需求沟通不到位，转换为软件需求规格说明不完整；用户需求变更管理不到位，用户需求可能存在变更，而软件开发者没有及时更改软件或软件更改引入新的问题等。

(2) 软件设计方法因素。软件开发者往往是先编码后写设计方案，或者按照自己的编程习惯开发软件，导致软件不符合用户需求，或未考虑用户对特殊情况处理和错误处理情况等。

(3) 编码过程因素。主要表现在软件编码不规范，软件容错能力不够等。

(4) 内部测试过程因素。如内部测试未开展或内部测试不充分等。

(5) 软件文档因素。如文档描述存在二义性、文文之间描述不一致，文档版本不一致等。

针对这些影响软件质量的因素，需要建立以软件质量为核心的质量体系来保证软件质量。建立树立为软件质量为核心的软件质量管理体系，如依据软件能力成熟度模型集成(CMMI)建立管理体系，明确在软件开发的各个阶段必须控制的质量指标。软件项目组严格按照质量体系开发软件，质量保证人员必须全程跟踪监督。软件质量体系度量包括以下方面：

## (1) 需求分析质量度量

软件需求分析是否充分、需求是否完整、需求定义是否准确，是否存在需求理解不同的情况，软件需求规格说明文档是否覆盖软件任务书或合同中所有要求。

由于需求分析工作对整体的软件工程都会产生较大的影响，所以在进行此阶段的工作时应当尽量让此项工程的参与人员加入进来，使整体工作人员对客户的需求均能有第一手的了解，避免复述这一过程中可能产生的任何错误。通常在软件公司中程序开发人员与程序测试人员在工作时是不会产生任何交集的，这就决定了许多问题不能得到协商，通过此项共同参与的协商过程能让开发人员与测试人员产生良好的交流，更好的完成问题的发现以及问题的解决。

## (2) 软件设计质量度量

需求开发完成以后，根据需求要求进行软件架构和设计。软件设计根据设计的层次不同，分为架构设计，概要设计和详细设计。

软件设计决定软件产品的结构，层次，模块划分和逻辑接口。软件设计要考虑软件的可用性、健壮性、可扩展性、一致性、以及安全性。可用性要求按设计实现的软件产品是满足基本业务需求的，这是软件设计的最基本要求。健壮性需要考虑软件运行时的容错处理、异常处理；在软件运行出错时能有自我修复能力，并且需要保证尽可能不影响用户的正常业务。

可扩展性要求设计要考虑软件后续可能的需求要求，这有个“度”的权衡。考虑可扩展性既不要过度设计，把一些遥远的、理想的、可能根本不会去实现的需求考虑进去，无谓增加了设计的复杂性；也不能做一天和尚敲一天钟，设计的软件结构无法较好适应需求的变化。软件设计的一致性体现两个方面，一方面是和需求规格能对应，保持需求可追溯；另一方面软件设计包括的三个层面设计接口上需要保持一致。安全性是指在设计阶段就需要考虑软件的安全问题，在软件结构上和规范上需要避免引入安全漏洞。

良好的软件设计应该是结构清晰、逻辑分明的。软件设计虽然比较复杂，设计人员需要有一定的积累和知识，但好在软件行业已经发展这么多年，先辈们也积累总结了若干软件架构设计的方法和模式，比如软件架构的 4+1 视图法，若干设计模式。软件设计要学会利用好这些成熟的方法和模式。

### （3）文档编码质量度量

在进行软件工程需求设计工作的过程中，进行软件需求分析文档的编写是十分重要的。在进行软件需求编写的过程中，需要用简洁明了的语言阐述客户对软件功能的预期。尽量避免过量专业术语的使用，通俗的语言能让程序员更好的理解；避免模糊性词句的出现以防程序员对文档的理解出现任何偏差影响其编写的程序，造成软件的计算方式变化；文档中使用的语句最好为实词，形容词以及有一些描述性的词语在不同人脑中存在的印象通常是不同的，为了防止产生误差应尽量避免。比如说在进行结合数据库软件开发的过程中需要提出的要求应当为具体的存储量数据以及可能会产生波动的范围，而不能说客户需要较大或者较小的存储量，准确的说法会让程序员在工作过程中更有把握，使其编写出来的程序质量更有保障。

### （4）测试结果质量度量

内部测试如单元测试、单元集成和测试、配置项测试、系统合格性测试是否都逐层开展，发现的问题是否都进行了处理，测试是否依据了软件需求文档等。

系统测试需要注意测试的全面性，测试需要有个全面的指导性文档，这个文档叫测试大纲，测试大纲规定了测试需要覆盖的范围和衡量指标，测试大纲根据需求说明书来编写。测试大纲的内容需要覆盖需求说明书里规定的功能性需求和非功能性需求，保证系统测试的完整性。

同时如何选择合适的自动化测试方案测试自动化的引进和实施，不仅涉及测试工作本身流程上、组织结构上的调整与改进，甚至也包括需求、设计、开发、维护及配置管理等其他方面的配合。如果对这些必要的因素没有考虑周全的话，必然在实施过程中处处碰壁，既定的实施方案也无法开展。

### （5）验收结果质量度量

是否组织开展软件验收工作，对软件完成的功能数量，各项性能指标等是否进行了评审，各项功能性能质量是否满足用户需求等。

为了保证软件设计的质量，也可以在项目中采用软件重用技术。软件重用是指在开发新软件的过程中重复使用已有的软件成分，该软件成分可能是已存在的软件，也可能是专门设计的可重用的软件构件。广义的软件重用还包括软件的开发思想方法、文档、环境、数据等。

由于封装和继承的特性，面向对象方法比其他软件开发方法更适于支持软件复用。封装意味着可以将表示构件的类看作黑盒子。用户只需了解类的外部接口，即了解它能够响应哪些消息，相应的对象行为是什么。继承是指在定义新的子类时，可利用可复用构件库中已有的父类的属性和操作。当然，子类也可以修改父类的属性与操作，或者引进新的属性与操作。构件的用户不需要了解构件的实现细节。最大限度地采用软件重用技术，不仅能缩短开发周期，提高开发效率，也能提高软件的可维护性和可靠性。

高质量的软件设计同时应该注意加强软件的容错设计。容错是指系统或软件出现有限数目的硬件或软件故障的情况下，系统仍具有连续正确执行任务的固有能力和能力。其主要目的是提供足够的冗余信息和合适的算法程序，使系统在实际运行时能够及时发现程序设计错误，采取补救措施，以提高软件可靠性。

软件容错设计有冗余设计与非冗余设计两类方法。冗余设计有软件冗余、硬件冗余、时间冗余和信息冗余等。其中软件冗余又有表决器、N 版本和恢复块等方法在软件应用较少；非冗余设计有保护处理、异常处理、容差处理看门狗等技术等。

在软件产品开发生命周期每个阶段都需要重视本阶段的质量控制，在开发周期各个阶段的流转时，对前阶段的输出结果可以做常规性评审，阶段流转要有明确的质量要求，在评审时需要评估符合要求后才允许进入下一阶段。严格的质量控制利于尽可能把质量问题都在本阶段发现和解决，避免缺陷放大效应带来的人力和成本浪费，从而提高软件开发效率，开发出高质量并符合用户需求的软件产品。