

如何保证软件设计的质量

软件在没有发布之前的开发过程主要分为需求分析、设计、编码和验证四个阶段，最终的软件质量与这四个阶段的各自质量之间的关系是：

最终的软件质量 = 需求分析质量 + 设计质量 + 编码质量 + 验证质量

即，最终的质量来自于各阶段质量之总和，只要其中一个环节质量是差，则产品的整体质量都将是差。由此看来每一个阶段的质量都起着决定性的作用。

对于如何保证软件设计的质量来说，可以先从软件设计意义来说。设计是什么？有人说设计是人类的思想，也有人说设计是模型，也有人说设计是规划人们完成工作的步骤。在敏捷开发社区里直接指出设计就是源代码！其实上面所说的设计概念都只提到设计本身的某一个方面。总结定义如下：设计是人类为了完成某一项任务而对该任务的实现进行不同程度的抽象，这样使人类在有限的智力范围内更容易、更优美、更便捷的实现任务目标。“思想”是一种对任务实现方法的抽象，“模型”是对任务实现结果的抽象，“规划的步骤”是对任务实现过程的抽象，“程序源代码”也是任务实现结果的一种抽象（这种抽象度比 UML 模型更低）。其实只有“可以部署到客户环境中的可执行系统”才是任务的主要结果，也即软件产品。因而，我们主要保证质量的产品就是：“可以部署到客户环境中的可执行系统”。设计的目的是降低任务的复杂度，对目标系统进行不通层次抽象，把系统易变、复杂的部分进行分离，解除不必要的耦合，使系统在满足功能需求的同时保证系统的可修改性、可重用性、可靠性、易用性、性能等非功能需求。

对于实现的系统很小、很简单，并且已经有成功的实现经验和案例，可以不需要做更高抽象层次的设计，源代码本省就是一种很精确的设计；对于中大规模的系统，或拥有复杂逻辑的系统，一般需要采用比源代码更高一级抽象层次的设计，比如 UML 类设计图等等来描述高层设计；如果系统非常复杂庞大、开发周期很长、开发成员众多，那就需要一个更加有条理、更加规范和严谨的设计抽象来保证工作的有序和协调了，同时需要更多的设计层次来降低系统的复杂性。是不是只要做了设计，就能确保软件产品的质量达到要求呢？大家知道这是不可能的！但设计能从架构或结构方面促使系统的质量达到要求，系统细节问题是不能靠设计保证的，这需要严格的测试过程。良好的设计能促进软件产品的质量提高，低劣的设计直接造成软件产品的质量低劣。所以说，为了保证开发出来的软件产品达到既定的质量要求或指标，我们需要根据这些质量要求，选择促进软件产品到达这些质量要求的设计策略，采用业界多年已经验证的设计原则和模式（比如大家熟知的著名的‘开闭原则’等等），进行

恰当的抽象，创造出优良的设计结果。同时在做这些设计的时候，也需要对设计进行复审，使其保证设计的方向不会偏离目标。世界没有包治百病的灵丹妙药，软件也没有万能的解决办法，只有不断改进我们解决问题的思想、方法和过程，良好软件设计就是我们在提升软件质量方面一个重要的法宝！

软件设计是从软件需求规格说明书出发，根据需求分析阶段确定的功能设计软件系统的整体结构、划分功能模块、确定每个模块的实现算法以及编写具体的代码，形成软件的具体设计方案。软件设计把许多事物和问题抽象起来，并且抽象它们不同的层次和角度。将问题或事物分解并模块化使得解决问题变得容易，分解的越细模块数量也就越多，它的副作用就是使得设计者考虑更多的模块之间耦合度的情况。

设计阶段是通过设计方法找出软件实现更好的方法，注意这里是“更好”两个字，而不是强调最好。不良设计并不会像需求分析失误那样很容易暴露出其本质，相反，它所暴露出的更多是表象，比如逻辑复杂、维护时举步为艰等等。如果参与者不具备一定的洞察力以发现隐藏在现象背后的不良设计本质，则很有可能身受其害却不能自拔，还以为“本来就有那么复杂”。

项目的开发是一个逐步演进的过程，项目组成员对于需求的理解也是逐步加深的，一开始合适的设计到后面看来很有可能就不够全面或显得力不从心，如果仍沿用以前的设计则自然将暴露出它的不足，进而会出现需要更高的维护成本。重构思想的提出，就是用于帮助项目演进设计的，当然，在运用重构方法时，应尽可能保证项目有足够的单元测试用例，以预防重构时又引入新的缺陷。重构不只是一个词，其核心应当是一个方法论，一个用于优化设计的方法论。

1、思想上重视

充分认识设计阶段的重要性，从思想上强调设计阶段质量保障工作的必要性与重要性。关于软件设计的重要性前文已从几个方面作了总结，不再赘述。项目团队成员与甲方都要充分理解并一致认同设计规范与设计评审等质量管理措施对整个项目的意义与重要性。

2、选用合适的设计思想、设计方法

设计开始，在充分了解需求与项目背景的前提下，结合项目情况采用恰当的设计思想与设计方法，从设计的指导思想与方法上避免设计阶段的质量瑕疵。我们在做软件设计时还要根据项目的具体情况与应用场景选用合适的设计思想作指导，选用合适的建模方法帮我们尽快理清系统的业务逻辑并理出思路。

从方法学的角度来讲，软件的设计与开发从最初的机器语言-汇编语言发展到面向过程的结构化设计方法，到现在应用较多的面向对象、面向组件发展到面向服务，每一步都体现了不断抽象、更加贴近业务实务的发展趋势。

不管采用什么样的设计方法进行架构设计，设计都需要以充分满足项目需求为目的，任何分析与设计方法只有针对具体问题才有实际意义。另一方面要考虑的是，采用的方法要侧重满足项目或产品的质量需求，也就是非功能性需求。确保设计阶段的质量无忧。

3、项目管理上避免

项目管理是贯穿整个项目生命周期的，80%的软件项目质量问题是由项目管理造成的。软件设计阶段作为软件项目的一个重要环节，要做好质量保障自然离不开好的项目管理。从设计团队组建到角色分工与权责确定，到设计规范的制定与流程梳理，所有这些工作都需要一个好的团队负责人去把控。设计团队负责人还要重视设计评审，通过设计评审不断发现问题，逐步完善细化设计架构与详细设计说明书，作为后期代码实现与测试用例编写的指导。要重视项目经理的作用，项目经理的职责是进行沟通，促进沟通并建立沟通的渠道。只有通过沟通才能在项目成员间建立起认同与理解，从而将设计思路有效实现。

4、引入专业的第三方质量保障服务机构指导

一般的项目建设，乙方自己充当质量保障的角色，部分软件企业为了降低成本，尽可能的减少质量保障环节的资源支出，致使设计质量无法保障，即使有部分软件企业视质量为生命，建立了良好的质量管理体系，但是囿于精力所限或赶工期或质量保障经验上的限制，设计质量也是不能令人满意。而从甲方看，一般囿于人员、技术、精力的限制，甲方很难有精力或技术能力去对项目的质量进行深入的关注。更何况软件本身并不可见，充满复杂的逻辑关系，模块之间的耦合关联度不易把握。第三方质量保障服务机构靠技术与服务来赢得客户信任，因而更加重视项目的质量与最终用户体验。从而会更加专业的对待项目过程中的质量管理。