# 软件工程产品质量 第1部分：质量模型阅读报告

131220064 刘闯

* GitHub账号：ChuangLiu27（[15850538707@163.com](mailto:%31%35%38%35%30%35%33%38%37%30%37@%31%36%33.%63%6f%6d)）

## 阅读内容总结

这次阅读的软件工程标准是“GB-T 16260.1-2006 软件工程 产品质量 第1部分：质量模型”。

软件产品质量可以通过测量内部属性（典型地是对中间产品的静态测度），也可以是通过测量外部属性（典型地是通过测量代码执行时的行为）。一般意义上来说，广义的软件产品质量应该是有递进关系的三部分所组成的，它们是过程质量，软件产品质量，使用质量。这个划分是符合人的一般认知的，前者有助于提高后者水平，而后者也可以为改进前者过程提供反馈。

标准中描述了关于软件产品质量的两部分模型：a）内部质量和外部质量，b）使用质量。

内部质量是基于内部视角的软件产品特性的总体，外部质量是基于外部视角的软件产品特性的总体。使用质量是基于用户观点的软件产品用于指定的环境和使用环境时的质量。

**外部和内部质量的质量模型可以划分为六个特性及细分后的子特性。它们是：**

**功能性：**当软件在指定条件下使用时，软件产品提供满足明确和隐含要求的功能的能力

①适当性：软件产品为指定的任务和用户目标提供一组合适的功能的能力

②准确性：软件产品提供具有所需精度的正确或相符的结果或效果的能力

③互操作性：软件产品与一个或更多的规定系统进行交互的能力。

④软件保密性：软件产品保护信息和数据的能力，以使未授权的人员或系统不能阅读或修改这些信息和数据，而不拒绝授权人员或系统对它们的访问

⑤功能性的依从性：软件产品遵循与功能性相关的标准、约定或法规以及类似规定的能力

**可靠性：**在指定条件下使用时，软件产品维持规定的性能级别的能力

①成熟性：软件产品为避免由软件中故障而导致失效的能力

②容错性：在软件出现故障或者违反其指定接口的情况下，软件产品维持规定的性能级别的能力

③易恢复性：在失效发生的情况下，软件产品重建规定的性能级别并恢复受直接影响的数据的能力

④可靠性的依从性：软件产品遵循与可靠性相关的标准、约定或法规的能力

**易用性：**在指定条件下使用时，软件产品被理解、学习、使用和吸引用户的能力

①软件产品使用户能理解软件是否合适以及能将软件用于特定的任务和使用条件的能力

②易学性：软件产品使用户能学习其应用的能力

③易操作性：软件产品使用户能操作和控制它的能力

④吸引性：软件产品吸引用户的能力

⑤易用性的依从性：软件产品遵循和易用性相关的标准、约定、风格指南或法规的能力

**效率：**在规定条件下，相对于所用资源的数量，软件产品可提供适当性能的能力

①时间特性：在规定条件下，软件产品执行其功能时，提供适当的响应和处理时间及其吞吐率的能力

②资源利用性：在规定条件下，软件产品执行其功能时，使用合适数量和类别的资源的能力

③效率依从性：软件产品遵循与效率相关的标准或约定的能力

**维护性：**软件产品可被修改的能力。修改可能包括纠正、改进或软件对环境、需求和功能规格说明变化的适应

①易分析性：软件产品诊断软件中的缺陷或失效原因或识别待修改部分的能力

②易改变性：软件产品使指定的修改乐意被实现的能力

③稳定性：软件产品避免由于软件修改而造成意外活动的能力

④易测试性：软件产品使已修改软件能被确认的能力

⑤维护性的依从性：软件产品遵循与维护性相关的标准或约定的能力

**可移植性：**软件产品从一种环境迁移到另外一种环境的能力

①适应性：软件产品毋需采用额外的活动或手段就可适应不同指定环境的能力

②易安装性：软件产品在指定环境中被安装的能力

③共存性：软件产品在公共环境中同与其分享公共资源的其他独立软件共存的能力

④易替换性：软件产品在同样的环境下，替代另一个相同用途的指定软件产品的能力

⑤可移植性的依从性：软件产品遵循与可移植性相关的标准或约定的能力

**使用质量的的属性分类为四个特性：**

**有效性：**软件产品在指定的使用周境下，使用户能达到与准确性和完备性相关的规定目标的能力

**生产率：**软件产品在指定的使用周境下，使用户为达到有效性而消耗适当数量的资源的能力

**安全性：**软件产品在指定的使用周境下，使用户为达到有效性而消耗适当数量的资源的能力

**满意度：**软件产品在指定的使用周境下，使用户满意的能力

## 阅读之后结合本组目前状况的分析

目前本组正在做需求分析和开发工具的学习。同步工具是github，开发语言是C++。需要注意到的质量问题：

（1）：应使得需求分析达到准确性和适当性。需求管理的准确性关乎到软件产品能否提供具有所需精度的正确或相符的结果或效果的能力，目前的需求已经改了两次，不知道会不会有第三次，对需求的深度挖掘也在进一步加深。根据已经从助教手里拿到的parameters.xml和militia.xml以及字段，可以很肯定知道编队系统的显性要求。但是，用户所描述的要求并不总是反映实际客户的需要，因为：（1）用户并不是经常可以意识到自己的实际需要，（2）要求在被说明之后还可能会继续变化，（3）不同用户可能具有不同的操作环境，（4）针对现在的软件，咨询所有可能类型的用户显然是做不到。

基于目前我们只能从助教和老师那里得到具体的项目要求，更深层次的实现需求还需要我们自己去分析。因此，质量需求不可能在开始设计之前就得到完整的定义。然而，尽可能详细地理解实际用户的需要并在需求中表示它们是必须的。就比如民兵编队用户虽然没有提出要实现批量处理的功能但是现实中为了处理多达50万的数据项，批量处理显然是必须的。

（2）：在本项目中使用质量的度量考虑。结合使用质量的定义，对最终用户来说使用质量主要是功能性、可靠性、易用性和效率性。在此基础上，最终用户会使用直观的反映来度量我们做的产品，例如功能要求中的“最终省级合并所有数据的冲突检测时间不大于5分钟”，对最终用户来说4分45秒和5分钟在具体使用过程中的差异不会明显，但是5分15秒和5分钟会有较明显的差别。

对软件维护人员来说使用质量是维护性的结果。维护人员在维护过程中对文档、代码、使用说明等进行再阅读，使用维护性的度量来评价软件产品。

作为项目的开发人员，我们可以根据使用任一质量特性的内部测度的准则值（代码行数、复杂度测试、在走查和Fog索引中发现的故障数）来评价软件产品。

## 阅读中值得思考的地方

在一个软件产品的生存周期内，内部质量、外部质量和使用质量的观点是时刻变化的。在开发的最早阶段，仅能测量资源和过程。当中间产品（规格说明、源代码等）可用时，就可以在选择的内部度量层次上对它们进行评价。

作为程序员，绝大部分包括我，注意到的产品质量还仅仅停留在使用质量这个层次上，思维上觉得只要编程实现出来的功能足以交付使用就足够了，往往这样的思维会使程序的内部质量（可靠性、易用性、效率等）被掩盖在满足功能的急切心理之下。

事实上作为一个大的项目，代码仅仅只是内部质量的一部分。