如何提高代码质量

**1.软件产品质量**

软件产品质量通常可以从以下六个方面去衡量(定义) :

         功能性(Functionality)，即软件是否满足了客户业务要求；

         可用性(Usability)，即衡量用户使用软件需要付出多大的努力；

         可靠性(Reliability)，即软件是否能够一直处在一个稳定的状态上满足可用性；

         高效性(Efficiency)，即衡量软件正常运行需要耗费多少物理资源；

         可维护性(Maintainability)，即衡量对已经完成的软件进行调整需要多大的努力；

         可移植性(Portability)，即衡量软件是否能够方便地部署到不同的运行环境中；

由此可见，软件产品的质量有其明显的特殊性。而目前提高软件产品质量的主要方法是软件过程质量控制。



软件产品质量关注点

**2.软件过程质量控制**

目前软件过程改进的主要方式有：

         自顶向下的方式：能力成熟度模型集成CMMI，其层次分为初始级、可重复级、已定义级、已经管级、优化级。

         自底向上的方式：个体软件过程（Personal   Software   Process,  PSP），向每个软件工程师阐明软件过程改进的原则，使他们能够明白如何有效地生产出高质量的软件。

**3.代码质量**

    软件是交付给用户，并由用户体验的产品；代码则是对软件正确且详细的描述，所以代码质量关系到软件产品的质量。虽然[软件](http://software.it168.com/)质量不等于代码质量，但是代码上的缺陷会严重的影响到软件产品的质量。因此，为提高代码质量的投入是值得的。

**4.代码质量的关注点**

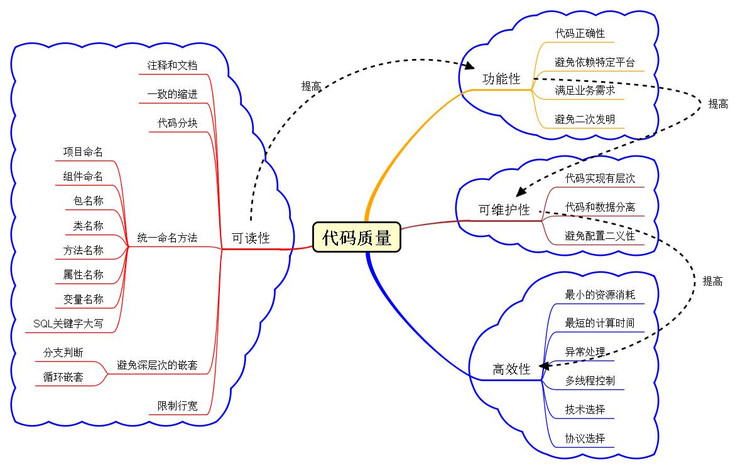
围绕软件质量的可度量特性，代码质量的关注点主要有：

         可读性：代码是否可读易读，对于一个团队来说，编码标准是否一致，编码风格是否一致；

         功能性：代码正确得实现了业务逻辑；

         可维护性：代码逻辑是有层次的，是容易修改的；

         高效性：代码实现在时间和空间的使用上是高效的；



代码质量关注点

**5.代码质量控制**

**5.1分阶段模型**

由于代码质量与开发人员息息相关，而开发人员的水平又有差异，所以在提高代码质量的改进过程中建议分阶段进行。

       借鉴CMMI的分级模型，代码质量的提高过程也是分阶段的：

**初始阶段，**开发人员认识到代码质量的重要性；

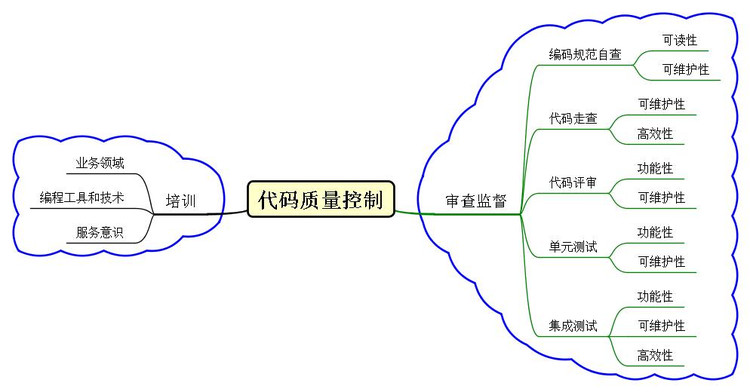
**提高阶段，**开发人员能够严格遵循开发规约；

**优化阶段，**开发人员能够自觉提高代码质量，能够为改进代码质量添砖添瓦；

很显然，我们已经处于初始阶段，为了进一步提高，必须具备几个条件，完整的规范、执行流程、检查机制和辅助工具。

**5.2提高代码质量的方法**

       提高代码质量的方法主要有：培训和审查监督。培训主要是针对业务领域、编程工具和技术、服务意识等基础方面的增强；而审查监督是针对编码规范、代码走查、代码评审、单元测试、集成测试、持续集成、执行流程等编程方面的增强。



代码质量控制方法关注点

**5.3质量控制过程**

