## 质量属性

质量是衡量卓越水平或无缺陷或无缺陷状态的一种度量。质量属性是与系统功能分开的系统属性。

实施质量属性使区分好系统和坏系统变得更加容易。属性是影响运行时行为，系统设计和用户体验的总体因素。

他们可以分类为静态质量属性和动态质量属性。

静态质量属性反映与体系结构，设计和源代码直接相关的系统和组织的结构。它们对于最终用户是不可见的，但是会影响开发和维护成本，例如：模块化，可测试性，可维护性等。

动态质量属性反映系统在执行过程中的行为。它们与系统的体系结构，设计，源代码，配置，部署参数，环境和平台直接相关。它们对于最终用户是可见的，并且在运行时存在，例如吞吐量，健壮性，可伸缩性等。

## 具体策略

可修改性：

如果应用程序的需求不清楚，则最大程度地减少前期的大型设计。如果有可能修改需求，则应避免对整个系统进行大型设计。我们对应用需求进行分析，着重开发重点功能，力求小而精。

性能：

在性能方面，我们将采用若干优化措施，比如减小前端的视图绘制耗时、进行数据交互的双向绑定保证数据正确，避免UI线程的长时间堵塞等。

安全性：

通信或系统服务（如日志记录，概要文件和配置）相关的代码应在单独的组件中抽象出来。我们在开发时，不将此代码与业务逻辑混合使用，因为扩展设计和维护很容易。

此外，我们将会在线程安全方面做工作，保证视图和数据双向绑定的稳定和正确的数据协同。

可重用性：

定义了组件和子系统的功能，使其适合在其他应用程序中使用。重用功能时要优先考虑组合（而不是继承），因为继承会在子类和父类之间建立依赖关系，因此会阻止子类的自由使用。相反，该组合提供了很大的自由度并减少了继承层次结构。我们将着重考虑这一点。