1. 测试框架，如JUnit（原则上鼓励使用自己项目适合的测试框架）
2. 覆盖率(<http://www.cnblogs.com/coderzh/archive/2009/03/29/1424344.html>)，例如可以要求行覆盖80%，分支覆盖50%
3. 语句覆盖（StatementCoverage）
4. 判定（分支）覆盖(DecisionCoverage)

#### 条件覆盖(ConditionCoverage)

#### 路径覆盖(PathCoverage)

#### 其他

#### 单元测试标准

#### http://cplusplus.wikidot.com/cn:unit-test-good-way

1. 开发模式
2. 领域驱动设计(DDD)

http://www.infoq.com/cn/articles/ddd-in-practice

1. 测试驱动开发TDD

<https://www.zhihu.com/question/37623307>

1. 为什么要写单元测试

我的实践告诉我：

1. 简化原来的方法（将复杂的拆成简单的再组合，算是重构）
2. 减少逻辑性错误（例如数学中的分母为0）
3. 防止别人修改后破坏了原有逻辑（决定的逻辑修改需要修改测试代码，主要是防止那些保持原有逻辑却不小心修改的事件发生）

总体上我觉得通过不同的重构代码，使得代码逻辑清晰，易读，重用性强。对于新手可以增强对设计模式6原则的理解和应用

附：设计模式六大原则

[设计模式六大原则（1）：单一职责原则](http://www.uml.org.cn/sjms/201211023.asp#1)

[设计模式六大原则（2）：里氏替换原则](http://www.uml.org.cn/sjms/201211023.asp#2)

[设计模式六大原则（3）：依赖倒置原则](http://www.uml.org.cn/sjms/201211023.asp#3)

[设计模式六大原则（4）：接口隔离原则](http://www.uml.org.cn/sjms/201211023.asp#4)

[设计模式六大原则（5）：迪米特法则](http://www.uml.org.cn/sjms/201211023.asp#5)

[设计模式六大原则（6）：开闭原则](http://www.uml.org.cn/sjms/201211023.asp#6)

特别注意：不要为了测试而测试 .

网上关于为什么要写单元测试的见解

<http://www.cnblogs.com/herbertchina/p/4306797.html>

<https://www.zhihu.com/question/20417403>

<http://blog.turn.tw/?p=2821>

<http://www.techug.com/post/twelve-benefits-of-writing-unit-tests-first.html>

写单元测试可能遇到的问题：

如果没有使用TDD 或是 项目从来没有写过单元测试，在写单元测试过程中，会大幅度增加代码重构，对原有逻辑产生冲击，这即需要程序员持有谨慎的态度，同时也需要回归测试，建议循环渐进，从简单到负责，从小类到大类……