1. 单元测试：测试程序构件中的所有方法和对象类。
2. 测试函数（方法）
3. 按照不同的方式调用函数。通过不同的输入样例，正在测试的构件能否做出希望的事情。尤其要注意在具有刁难性的测试用例时构件做出的事情。如果程序正常运行，构件可以工作并且没有缺陷，测试成功。
4. 可以通过规格说明或用户文档，并且根据预测哪种类型的输入值最有可能发现错误的经验来识别划分。
5. 测试对象类：模拟状态变更的所有事件

设计测试来实现对该对象所有特征的覆盖，也就是测试与该对象相关的所有操作，设置并且检查与该对象相关的所有属性的值，并且将该对象置于所有看待状态中。

3、对源程序进行分析。比如用代码审查方法。

1. 构件测试

构件由相互交互的对象组成，构件之间存在不同类型的接口，可以通过构件接口访问这些对象的功能。这里需要检查接口错误，可能发生在以下几种类型：

1. 参数接口
2. 共享存储接口
3. 过程式接口
4. 消息传递接口

接口错误包括下面几种类型：

1. 接口误用
2. 接口误解
3. 时间性错误
4. 系统测试

与构件测试存在重叠，但是有区别。此处应该关注构件和对象之间的交互，还可能会在可复用构件与新的构件集成时对它们进行测试，检验它们是否像如期一样运作。比如可以在任何提供了用户输入的地方，所有的功能必须同时使用正确和不正确的输入进行测试。

1. 发布测试
2. 基于需求的测试：确认需求是否满足，为每一个需求创建一组测试。
3. 场景测试：设想典型的使用场景，也就是描述系统可能的使用方式，并且是现实的。需要运行整个场景，看系统对不同的输入进行的响应和表现，检查系统对于错误的响应。
4. 性能测试：确保系统可以处理预计承受的负载。观察系统在满负荷或者是超负荷运转时是否出现缺陷，出现什么缺陷。
5. 用户测试

找人作为软件系统的用户，包含6个阶段。

1. 定义验收准则。
2. 计划验收测试。
3. 设计验收测试用例。
4. 运行验收测试。
5. 协商测试结果。
6. 接受或拒绝系统。