1. 软件生命周期 (SDLC) 的六个阶段

1、问题的定义及规划

此阶段是软件开发方与需求方共同讨论，主要确定软件的开发目标及其可行性。

2、需求分析

在确定软件开发可行的情况下，对软件需要实现的各个功能进行详细分析。需求分析阶段是一个很重要的阶段，这一阶段做得好，将为整个软件开发项目的成功打下良好的基础。 "唯一不变的是变化本身。 "，同样需求也是在整个软件开发过程中不断变化和深入的，因此我们必须制定需求变更计划来应付这种变化，以保护整个项目的顺利进行。

3、软件设计

此阶段主要根据需求分析的结果，对整个软件系统进行设计，如系统框架设计，数据库设计等等。软件设计一般分为总体设计和详细设计。好的软件设计将为软件程序编写打下良好的基础。

4、程序编码

此阶段是将软件设计的结果转换成计算机可运行的程序代码。在程序编码中必须要制定统一，符合标准的编写规范。以保证程序的可读性，易维护性，提高程序的运行效率。

5、软件测试

在软件设计完成后要经过严密的测试，以发现软件在整个设计过程中存在的问题并加以纠正。整个测试过程分单元测试、组装测试以及系统测试三个阶段进行。测试的方法主要有白盒测试和黑盒测试两种。在测试过程中需要建立详细的测试计划并严格按照测试计划进行测试，以减少测试的随意性。

6、运行维护

软件维护是软件生命周期中持续时间最长的阶段。在软件开发完成并投入使用后，由于多方面的原因，软件不能继续适应用户的要求。要延续软件的使用寿命，就必须对软件进行维护。软件的维护包括纠错性维护和改进性维护两个方面。

2、软件生命周期模型

从概念提出的那一刻开始，软件产品就进入了软件生命周期。在经历需求、分析、设计、实现、部署后，软件将被使用并进入维护阶段，直到最后由于缺少维护费用而逐渐消亡。这样的一个过程，称为 "生命周期模型 "（Life Cycle Model ）。

典型的几种生命周期模型包括瀑布模型、快速原型模型、迭代模型。

瀑布模型的特点 （文档是主体），很多的问题在最后才会暴露出来。迭代模型比瀑布模型问题暴露的要早；快速原型法比瀑布模型直观。

3. 软件测试概念

广义概念：指软件生存周期中所有的检查、评审和确认工作，其中包括了对分析、设计阶段，以及完成开发后维护阶段的各类文档、代码的审查和确认

狭义概念：识别软件缺陷的过程，即实际结果与预期结果的不一致

4. 软件测试目的

测试的目的就是发现软件中的各种缺陷
测试只能证明软件存在缺陷，不能证明软件不存在缺陷
测试可以使软件中缺陷降低到一定程度，而不是彻底消灭
以较少的用例、时间和人力找出软件中的各种错误和缺陷，以确保软件的质量

5．软件测试原则

Good-enough:　　一种权衡投入 / 产出比的原则
保证测试的覆盖程度，但穷举测试是不可能的
所有的测试都应追溯到用户需求
越早测试越好，测试过程与开发过程应是相结合的
测试的规模由小而大，从单元测试到系统测试
为了尽可能地发现错误，应该由独立的第三方来测试
不能为了便于测试擅自修改程序
既应该测试软件该做什么也应该测试软件不该做什么

6．软件测试的的重点

测试用例的设计
- 测试用例的设计是整个软件测试工作的核心
- 测试用例反映对被测对象的质量要求，决定对测试对象的质量评估

测试工作的管理
- 尤其是对包含多个子系统的大型软件系统，其测试工作涉及大量人力和物力，有效的测试工作管理是保证有效测试工作的必要前提

测试环境的建立
- 测试环境应该与实际测试环境一致

7．黑盒测试

什么是黑盒测试

- 又称功能测试或数据驱动测试，是针对软件的功能需求 /实现进行测试，通过测试来检测每个功能是否符合需求，不考虑程序内部的逻辑结构

黑盒测试方法

- 功能划分
- 等价类划分
- 边界值分析
- 因果图
- 错误推测等

8．什么是白盒测试

- 白盒测试也称结构测试或逻辑驱动测试，必须知道软件内部工作过程，通过测试来检测软件内部是否按照需求、设计正常运行

- 白盒测试的主要方法
  - 对应于程序的一些主要结构：语句、分支、逻辑路径、变量；白盒测试的主要方法是：
    - 语句覆盖方法
    - 分支覆盖方法
    - 逻辑覆盖方法

9. 什么是动态测试

动态测试需要在开发 / 测试环境或实际运行环境中运行软件，并使用测试用例去查找软件缺陷；动态测试包括功能确认与接口测试、覆盖率分析、性能分析、内存分析等

10. 什么是静态测试

静态测试不实际运行软件，主要是对软件的编程格式、结构等方面进行评估 .静态测试包括代码检查、程序结构分析、代码质量度量等。它可以由人工进行，也可以借助软件工具自动进行

11. 手工测试和自动测试

a. 手工测试缺点在于测试工作量大，重复多，回归测试难以实现

b. 自动测试利用软件测试工具自动实现全部或部分测试工作：管理、设计、执行和报告；节省大量的测试开销，并能够完成一些手工测试无法实现的测试

手工完成测试的全部过程无法保证测试的科学性与严密性 :

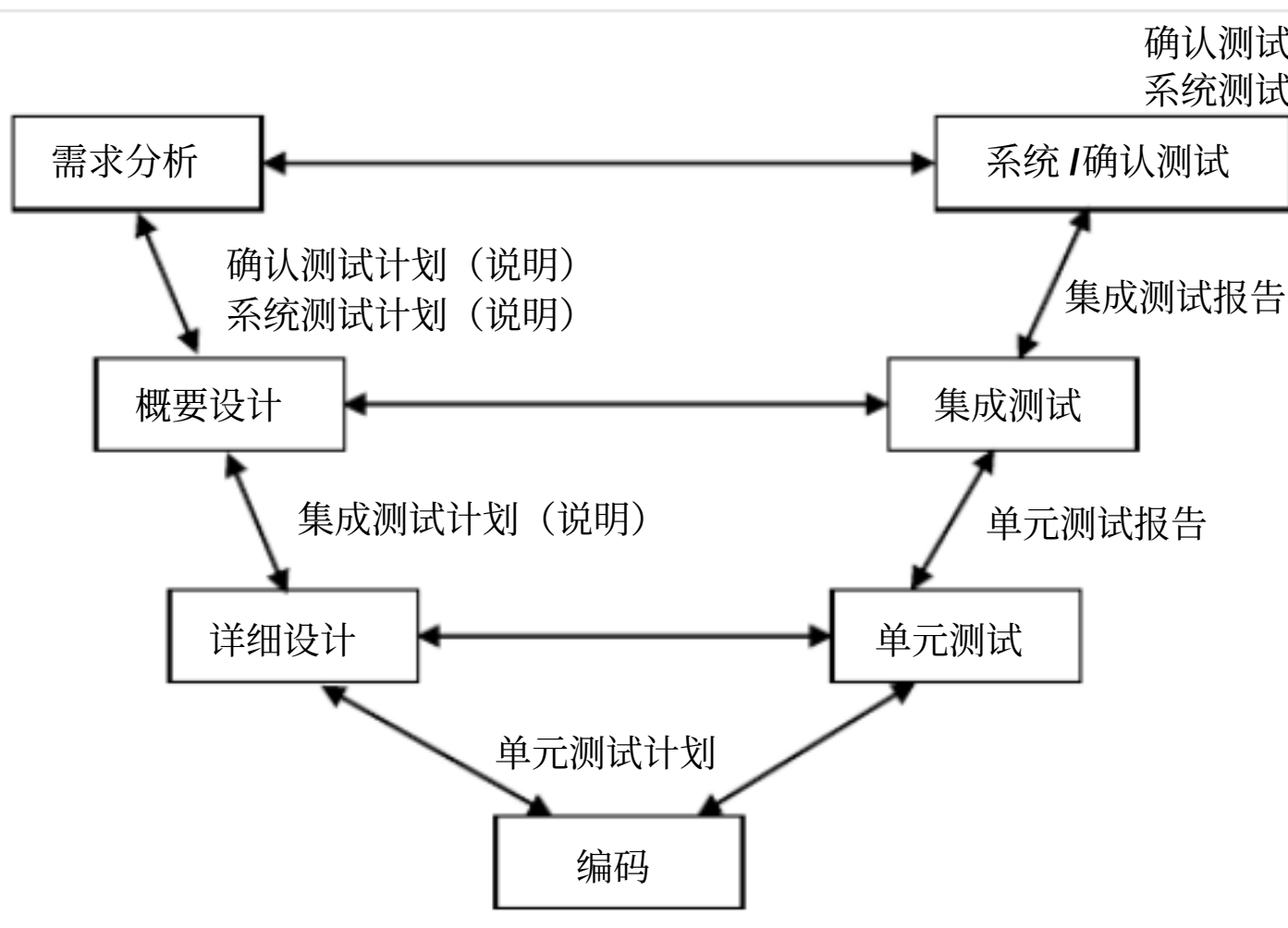
- 修改的缺陷越多，回归测试越困难

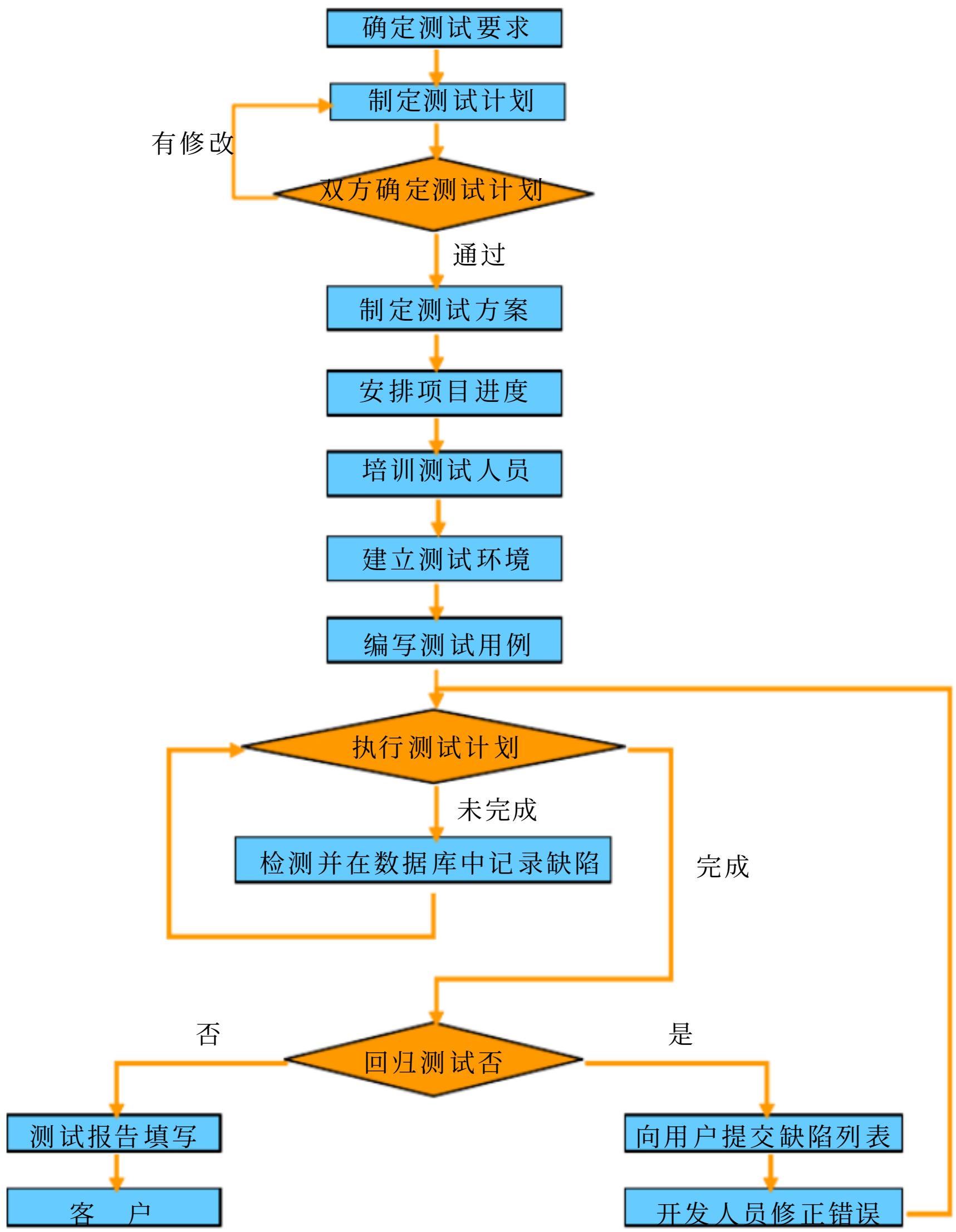
- 没有人能向决策层提供精确的数据以度量当前的工作进度及工作效率
- 反复测试带来的倦怠情绪及其他人为因素使得测试标准前后不一
- 测试花费的时间越长，测试的严格性也就越低

自动测试将测试人员从反复、烦杂的测试执行中解放出来，用更多的时间进行测试设计和结果分析
软件测试不可能完全自动化
不能完成所有手工测试任务
无创造性且灵活性差，不能改进测试的有效性
过程中可能会遇到许多意想不到的问题，特别是当软件不稳定时
测试脚本的维护高

12. 测试流程

单元测试
集成测试
系统测试
用户验收测试
回归测试





13. 单元测试

完成对最小的软件设计单元　—模块的验证工作
目标是确保模块被正确地编码

使用过程设计描述作为指南，对重要的控制路径进行测试以发现模块内的错误
通常情况下是面向白盒的
对代码风格和规则、程序设计和结构、业务逻辑等进行静态测试，及早地发现和解决不易显现的错误

单元测试的内容

- 接口测试
- 内部数据结构
- 全局数据结构
- 边界
- 语句覆盖，错误路径

14. 集成测试

通过测试发现与模块接口有关的问题
目标是把通过了单元测试的模块拿来，构造一个在设计中所描述的程序结构
应当避免一次性的集成（除非软件规模很小），而采用增量集成

集成测试主要内容

API
API/ 参数组合

15 ． 系统测试

根据软件需求规范的要求进行系统测试，确认系统满足需求的要求
系统测试人员相当于用户代言人
在需求分析阶段要确定软件的可测性，保证有效完成系统测试工作
系统测试主要内容
所有功能需求得到满足
所有性能需求得到满足
其他需求（例如安全性、容错性、兼容性等）得到满足

16. 用户验收 / 确认测试

Alpha 测试

- 是由用户在开发者的场所来进行的， Alpha 测试是在一个受控的环境中进行的

Beta 测试

- 由软件的最终用户在一个或多个用户场所来进行的，开发者通常不在现场，用户记录测试中遇到的问题并报告给开发者

17 ． 压力测试 VS 性能测试

性能测试的目的不是去找 bugs，而是排除系统的瓶颈，以及为以后的回归测试建立一个基准。而性能测试的操作，实际上就是一个非常小心受控的测量分析过程。在理想的情况下，被测软件在这个时候已经是足够稳定了

性能测试是为了检查系统的反映，运行速度等性能指标，他的前提是要求在一定负载下，如检查一个网站在 100 人同时在线的情况下的性能指标，每个用户是否都还可以正常的完成操作等。
概括就是：在不同负载下（负载一定）时，通过一些系统参数（如反应时间等）检查系统的运行情况；

压力测试是为了发现系统能支持的最大负载，他的前提是要求系统性能处在可以接受的范围内，比如经常规定的叶面 3 秒钟内响应；概括就是：在性能可以接受的前提下，测试系统可以支持的最大负载。

举例说明：针对一个网站进行测试，模拟 10 到 50 个用户就是在进行常规性能测试，用户增加到 1000 乃至上万就变成了压力 / 负载测试。如果同时对系统进行大量的数据查询操作，就包含了强度测试。

18. 主流 测试工具 的测试流程

========winrunner
1 启动时选择要加载的插件
2 进行一些设置（如录制模式等）
3 识别应用程序的 GUI，即创建 map(就是学习被测试软件的界面）
4 建立测试脚本（录制及编写）
5 对脚本除错及调试（保证能够运行完）
6 插入各种检查点（图片，文字，控件等）
7 在新版应用程序中执行测试脚本
8 分析结果，回报缺陷

=========quicktestpro========
1 准备录制
打开你要对其进行测试的应用程序，并检查 QuickTest 中的各项设置是否适合当前的要求。
2 进行录制
打开 QuickTest 的录制功能，按测试用例中的描述，操作被测试应用程序。
3 编辑测试脚本
通过加入检测点、参数化测试，以及添加分支、循环等控制语句，来增强测试脚本的功能，使将来的回归测试真正能够自动化。
4 调试脚本
调试脚本，检查脚本是否存在错误。
5 在回归测试中运行测试
在对应用程序的回归测试中，通过 QuickTest 回放对应用程序的操作，检验软件正确性，实现测试的自动化进行。

6　分析结果，报告问题
查看 QuickTest　记录的运行结果，记录问题，报告测试结果。

＝＝＝＝ TestDirect==========
安装好后，先进入站点管理
1　创建域及工程
2　添加用户
3　编辑 licenses　及本服务器
4　编辑数据库
－－ TD
1　选择新建的工程进行定制　(列表，用户，组，版本等　)
2　在 require　中增加需求
3　把需求转化为　plan
4 在 testlab　中由计划新建测试具体用例与执行

5　发现 bug，在 defect　中提交 bug
（每一部分都可以相对独立地使用）

======loadrunner
1　制定负载测试计划
(分析应用程序，　确定测试目标，计划怎样执行　LoadRunner）
2　开发测试脚本
（录制基本的用户脚本，完善测试脚本）
3　创建运行场景
（选择场景类型为　Manual Scenario ，选择场景类型，理解各种类型，场景的类型转化）
4　运行测试
5　监视场景
（MEMORY 相关， PROCESSOR 相关，网络吞量以及带宽，磁盘相关，　WEB 应用程序 ， IIS5.0 ， SQL SERVER, NETWORK DELAY

爱人者，人恒爱之；敬人者，人恒敬之；宽以济猛，猛以济宽，政是以和。将军额上能跑马，宰相肚里能撑船。

最高贵的复仇是宽容。有时宽容引起的道德震动比惩罚更强烈。

君子贤而能容罢，知而能容愚，博而能容浅，粹而能容杂。

宽容就是忘却，人人都有痛苦，都有伤疤，动辄去揭，便添新创，旧痕新伤难愈合，忘记昨日的是非，忘记别人先前对自己的指责和谩骂，时间是良好的止痛剂，学会忘却，生活才有阳光，才有欢乐。

不要轻易放弃感情，谁都会心疼；不要冲动下做决定，会后悔一生。也许只一句分手，就再也不见；也许只一次主动，就能挽回遗憾。

世界上没有不争吵的感情，只有不肯包容的心灵；生活中没有不会生气的人，只有不知原谅的心。

感情不是游戏，谁也伤不起；人心不是钢铁，谁也疼不起。好缘分，凭的就是真心真意；真感情，要的就是不离不弃。

爱你的人，舍不得伤你；伤你的人，并不爱你。你在别人心里重不重要，自己可以感觉到。所谓华丽的转身，都有旁人看不懂的情深。

人在旅途，肯陪你一程的人很多，能陪你一生的人却很少。谁在默默的等待，谁又从未走远，谁能为你一直都在？

这世上，别指望人人都对你好，对你好的人一辈子也不会遇到几个。人心只有一颗，能放在心上的人毕竟不多；感情就那么一块，心里一直装着你其实是难得。

动了真情，情才会最难割；付出真心，心才会最难舍。

你在谁面前最蠢，就是最爱谁。其实恋爱就这么简单，会让你智商下降，完全变了性格，越来越不果断。

所以啊，不管你有多聪明，多有手段，多富有攻击性，真的爱上人时，就一点也用不上。

这件事情告诉我们。谁在你面前很聪明，很有手段，谁就真的不爱你呀。

遇到你之前，我以为爱是惊天动地，爱是轰轰烈烈抵死缠绵；我以为爱是荡气回肠，爱是热血沸腾幸福满满。

我以为爱是窒息疯狂，爱是炙热的火炭。婚姻生活牵手走过酸甜苦辣温馨与艰难，我开始懂得爱是经得起平淡。

爱人者，人恒爱之；敬人者，人恒敬之；宽以济猛，猛以济宽，政是以和。将军额上能跑马，宰相肚里能撑船。

最高贵的复仇是宽容。有时宽容引起的道德震动比惩罚更强烈。

君子贤而能容罢，知而能容愚，博而能容浅，粹而能容杂。

宽容就是忘却，人人都有痛苦，都有伤疤，动辄去揭，便添新创，旧痕新伤难愈合，忘记昨日的是非，忘记别人先前对自己的指责和谩骂，时间是良好的止痛剂，学会忘却，生活才有阳光，才有欢乐。

不要轻易放弃感情，谁都会心疼；不要冲动下做决定，会后悔一生。也许只一句分手，就再也不见；也许只一次主动，就能挽回遗憾。

世界上没有不争吵的感情，只有不肯包容的心灵；生活中没有不会生气的人，只有不知原谅的心。

感情不是游戏，谁也伤不起；人心不是钢铁，谁也疼不起。好缘分，凭的就是真心真意；真感情，要的就是不离不弃。

爱你的人，舍不得伤你；伤你的人，并不爱你。你在别人心里重不重要，自己可以感觉到。所谓华丽的转身，都有旁人看不懂的情深。

人在旅途，肯陪你一程的人很多，能陪你一生的人却很少。谁在默默的等待，谁又从未走远，谁能为你一直都在？

这世上，别指望人人都对你好，对你好的人一辈子也不会遇到几个。人心只有一颗，能放在心上的人毕竟不多；感情就那么一块，心里一直装着你其实是难得。

动了真情，情才会最难割；付出真心，心才会最难舍。

你在谁面前最蠢，就是最爱谁。其实恋爱就这么简单，会让你智商下降，完全变了性格，越来越不果断。

所以啊，不管你有多聪明，多有手段，多富有攻击性，真的爱上人时，就一点也用不上。

这件事情告诉我们。谁在你面前很聪明，很有手段，谁就真的不爱你呀。

遇到你之前，我以为爱是惊天动地，爱是轰轰烈烈抵死缠绵；我以为爱是荡气回肠，爱是热血沸腾幸福满满。

我以为爱是窒息疯狂，爱是炙热的火炭。婚姻生活牵手走过酸甜苦辣温馨与艰难，我开始懂得爱是经得起平淡。