1、成熟软件企业的特性

软件开发和维护，遵循计划规定的过程；

严格的项目管理 按合理估算的计划监控；

贯彻软件质量保证；

重视需求分析和设计，强调客户满意；

有变化控制和配置管理；

进度和成本控制有保证，项目的成功依靠组织架构的支持，有较好重复性；

提高软件产品质量的有力手段是进行测试

测试是为发现错误（缺陷）而执行程序的过程

软件测试目的是发现软件中的错误

有效的测试用例：能有效地发现别的测试用例未发现的错误

成功测试是发现了未曾发现的错误

软件测试的原则：

1、尽早地和不断地进行软件测试；

2、避免测试自己的程序，要交叉测试；

3、要按测试计划来执行；

4、增量测试；

5、要设计好周密的测试用例

6、缺陷统计和分析

2、软件测试对象：程序+文档=软件

3、软件测试应贯穿于软件定义与开发的整个期间

4、软件的种类：

新开发软件：关注软件的内部质量和外部质量特性

现货软件：关注软件的使用质量特性

定制软件：关注软件的使用质量特性

现存软件：关注现存软件与将来业务目标的差距

基于计算机系统的软件：关注软件与基于计算机系统其它成份之间的接口

5、测试软件文档：完整性、正确性、一致性、易理解、易浏览。

6软件错误的程度：从低到高 错误 缺陷 故障 失效 事故

7、软件的可测性：被测试的容易程度

8、软件的测试方法：白盒测试和黑盒测试

白盒测试：测试软件结构、分支、代码进行测试，又称结构测试

黑盒测试：输入一个测试用例得到一个可接受的结果，又称功能测试

黑盒测试的错误类型：

①是否有不正确或遗漏了的功能？

②在接口上，输入能否正确地接受？能否输出正确的结果？

③是否有数据结构错误或外部信息（例如数据文件）访问错误？

④性能上是否能够满足要求？如：是否能导入导出数据等

⑤是否有初始化或终止性错误？

白盒测试的错误类型：

1．对程序模块的所有独立的执行路径至少测试一次。

2.对所有的逻辑判定，取“真”与取“假”的两种情况都能至少测试一次；

3.在循环的边界和运行界限内执行循环体；

4.测试内部数据结构的有效性等

软件测试的流程（阶段）

单元测试：各个模块内部结构的独立测试

集成测试：经过单元测试之后，各模块集成时各接口的测试

确认测试：经过集成测试之后，针对用户的需求进行的测试

系统测试：经过确认测试之后，结合用户使用环境（软硬件，网络设备）来进行的测试

α测试：在开发环境中进行的测试

β测试：在用户环境中进行的测试

验收测试：按合同要求测试使客户接受此测试

通过验收测试的软件产品应定义发布的软件版本号

回归测试：修改后重新走一遍以前做过的测试，用于验证修改后是否会引起不必要的负面影响

软件测试的生命周期:

风险评估: 能力、领域、目标、工具、经验等是否具备

测试计划: 制定需求、过程、估算、进度、方法等

测试设计: 设计测试用例

测试开发: 开发驱动模块、桩模块代码

测试执行:记录测试的过程

测试评估:书写测试报告

缺陷等级:

5级：灾难性的—系统崩溃、数据被破坏

4级：很严重的—数据被破坏，

3级：严重的—特性不能运行，无法替代

2级：中等的—特性不能运行，可替代，

1级：烦恼的—提示不正确，报警不确切，

0级：轻微的—表面化的错误，拼写错等，

从哪些方面去设计用例:

方法1：应用输入强制产生所有错误信息—测错误信息

方法2：质疑软件的默认值—测变量默认值

方法3：使用特殊字符输入测试允许的字符集合– 测特殊字符

方法4：使输入缓冲区上溢—测允许最大字符长度

方法5：测试输入值的组合逻辑--测组合表达式的最大范围

方法6：多次重复同样的输入序列—测系统资源（速率和空间）情况

方法7：强制每个输入产生不同的输出—测同输入异输出

方法8：强制产生无效输出—测领域知识内的无效输出

方法9：强制改变输出属性—测异常的输出属性,尽量设置默认值

方法10：强制屏幕刷新—测屏幕刷新的最佳点

**白盒测试常用的逻辑覆盖测试方法**：

1.语句覆盖－使程序中的每个可执行语句至少执行一次。

2.判定覆盖－使程序中的每个判定至少都获得一次“真”和“假”值。

3.条件覆盖－使每个判定中每个条件的可能取值至少满足一次。

4.判定/条件覆盖－使判定中的每个条件的可能结果和每个判定本身的判定结果均至少出现一次。

5.条件组合覆盖－使得每个判定中条件的各种可能组合都至少出现一次。

6.路径覆盖－覆盖程序中所有可能的路径

测试工程师的职业道德:

测试的公正性

报告结果的客观性

测试专业能力的提升－科学性

单元测试

1. 在开发阶段进行
2. 是软件设计的最小单位
3. 是程序员的一项基本职责

合格代码：

1. 正确性:代码逻辑必须正确，能够实现预期的功能
2. 清晰性:代码必须简明、易懂，注释准确没有歧义
3. 规范性:代码必须符合企业或部门所定义的共同规范
4. 一致性:代码必须在命名上风格上都保持统一
5. 高效性:尽可能降低代码的执行时间

单元测试的步骤：

1. 人工静态检查：通过人工检查发现代码的逻辑正确性、清晰性、规范性、一致性、算法高效性
2. 动态执行跟踪：通过设计测试用例，执行待测程序来跟踪比较实际结果与预期结果来发现错误

单元测试的任务：

1. 对模块内所有独立的执行路径至少测试一次
2. 对所有的逻辑判定，取“真”与“假”的两种情况都至少执行一次
3. 在循环的边界和运行界限内执行循环体
4. 测试内部数据的有效性等等

好的测试用例的原则：

1. 一个好的测试用例在于能够发现至今没有发现的错误
2. 测试用例应由测试输入数据和与之对应的预期输出结果这两部分组成
3. 在测试用例设计时，应当包含合理的输入条件和不合理的输入条件

路径覆盖法测试用例设计难度低，但它的覆盖程度高，常选该方法

语句覆盖法测试用例设计难度高，但它的覆盖程度低，很少选该方法

测试类有两个模块： 驱动模块和桩模块

JUnit两个*设计模式*：*命令模式 （ TestCase 类）、集成模式 （ TestSuite 类）*

性能测试的内容：

* + 确定应用的反映时间
  + 确定系统能够支持多少用户
  + 确定优化的系统配置
  + 验证不同的硬件和软件能够良好的运行
  + 找出当系统在高负载的情况下发生的问题

性能测试的目的：性能优化、最小化成本、最小化风险

* 性能测试的好处：从用户的角度改进质量
* 减少变更的成本
* 减少系统的成本
* 增加利润
* 尽早的发现主要的应用缺陷和架构上的问题

性能测试的理念：建模和经验主义，常用的是经验主义

三种经验主义方法：

1. 基于硬件的测试：

要求

1. 需要很多台实际的计算机
2. 需要一名测试人员
3. 不适合于大量用户系统的测试
4. 基于软件测试

要求：1）至少一台物理计算机模拟很多个用户

1. 需要一个测试人员
2. 适合于大量用户系统的测试

3、基于软件和硬件合并的测试: 利用软件方法能够的优势来测试服务器的相应时间，通过硬件方法的能力测试端对端客户的相应时间

获取用户需求的方法：访谈；工作的缩影/观察；调查；系统度量和日志记录

性能测试的类型：

负载测试：在强负载的和变化的用户级别的测试情况下测试服务器的相应时间

压力测试：测试应用或者服务器的崩溃点，确保系统在产品环境下的生存能力

竞争测试：模拟一个真实世界的用户环境来识别复杂的问题

配置测试：测试兼容性的问题，确定最小的和最优化的硬件和软件配置

性能测试的原则：

* 精确的建模用户负载量
* 为了比较测试使用相同的负载量
* 在可控的环境下进行测量
* 当比较结果时一次只更改一个变量

Web测试

Web测试的功能测试的内容：

1、链接测试 :是否己链、是否存在、是否孤立无被链；

2、表单测试：提交操作的完整性和校验提交的正确性；  
3、Cookies测试： Cookies是否起作用，刷新对Cookies有什么影响等；

4、设计语言测试 ：设计语言版本的差异可以引起客户端或服务器端严重的问题

5、数据库测试：是否连库，数据输出是否一致或正确；

6、数据校验：需要对用户输入进行校验

7、应用程序特定的功能需求：如下订单、更改订单、取消订单等等

Web测试中性能测试

1、响应速度测试：不超过5秒钟  
2、负载测试：多用户同时在线或同时对同一页面请求

3、压力测试 ：测试系统的崩溃极限和故障恢复能力

Web测试中可用性测试

1、导航测试:是否易于导航?导航是否直观？

2、图形测试：页面字体的风格是否一致、颜色是否搭配。

3、内容测试：正确性、准确性、相关性

4、整体界面测试：整体感、舒适感。

客户端兼容性测试

1、平台测试:操作系统取决于用户系统的配置

2、浏览器测试：测试不同厂商的IE与宿主语言是否兼容；

3、视频设置:分辨率模式

4、Modem/连接速率

5、打印机:打印是否正常

6、组合测试

安全性测试

1、目录设置: 设置index.html 或 main.html 页面

2、系统日志:验证服务器日志是否工作正常

3、SSL 测试:进行安全传送

4、登陆:验证身份,阻止非法的用户名/口令登录

5、服务器端的脚本是否构成安全漏洞

6、超时限制:超时(15分钟后)是否需要重新登陆才能正常使用

基本路径覆盖法

第一步：根据源程序画出该代码的程序流程图

提示：有条件分支的代码有菱形表示，其它语句用矩形表示，源代码的起点用圆表示，每个图形中写该代码行的行号

第二步：由程序流程图画控制流图

提示：在程序流程图的基础上，转化成控制流图，所有图形（菱形、矩形）都转成圆形，直线转成曲线或直线，省略起点图形。

第三步：计算圈复杂度

1、流图中区域的数量对应于环型的复杂性；

2、给定流图G的圈复杂度V(G)，定义为V(G)=E-N+2，E是流图中边的数量，N是流图中结点的数量；

3、给定流图G的圈复杂度V(G)，定义为V(G)=P+1，P是流图G中判定结点的数量。

第四步：确定基本路径集合

上述圈复杂度就是该程序基本路径的个数

自动化测试

手工测试与自动化测试的测试时间比较:总的测试时间有差距,自动化测试所用时间远小于手工测试所用时间，尤其是在执行测试和产生测试报告方面。

自动化测试的步骤  
第一步.录制脚本：录制测试过程成为自动化测试脚本  
第二步.编辑脚本：增强和改进录制的自动化测试脚本  
第三步.执行脚本：执行自动化测试脚本完成自动化测试

理解：

1、不是在时间紧迫时选择自动化测试。

2、测试自动化不只是捕获和回放

3、执行自动化测试的员工需要不断地培训

4、没有100% 的自动化测试。

自动化测试的 5 个级别 ：

级别 1：捕获和回放

级别 2：捕获、编辑和回放

级别 3：编程和回放

级别 4：数据驱动的测试

级别 5：使用动作词的测试自动化

自动化测试的场合：

1、单元测试/组件测试

2、冒烟测试

3、功能/集成测试

4、系统测试 － 用例测试

5、回归测试

6、性能测试

自动化测试的好处：

1、降低风险

2、快速的反馈

3、能在多个平台上同时进行测试

4、更好的估算测试进度

5、能集中更多的优秀人员

6、喜悦

第十章：测试过程

测试过程

1、制定测试计划：确定和描述要实施和执行的测试。

2、设计测试：确定、描述和生成测试模型及测试用例。

3、实施测试：实施（记录、生成或编写）设计测试中定义的测试过程

4、在集成测试阶段执行测试

5、在系统测试阶段执行测试

6、评估测试：生成并交付测试评估摘要。

测试角色

1. 测试设计人员：主要角色，负责对测试进行计划、设计、实施和评估
2. 测试实施人员：设置和执行测试，评估测试执行过程并修改错误

测试设计人员要提供的产物：

测试指南、测试计划、工作量分析文档、测试过程、测试用例、测试脚本、测试评估摘要

测试实施人员要提供的产物 ：测试结果、测试评估摘要

面向对象测试与传统测试的不同点：

1、测试的定义必须扩大包括用于 OOA 和 OOD 模型的错误发现技术

2、单元和集成测试策略必须有很大的改变

3、测试用例的设计必须考虑OO软件的独特特征

面向对象的测试内容：

测试类模型在语法、语义和语境的正确性、完整性和一致性

面向对象测试策略：

1. 单元测试：测试单位是类或对象，即类或对象的属性和操作（行为）
2. 集成测试：采用基于线程或类的测试策略来进行集群测试。
3. 确认测试（有效性测试）：用户根据需求来进行可见的动作和可识别的系统输出

OO软件的测试用例设计的技术和方法适用于传统测试的白盒和黑盒的测试方法