从软件开发者的角度出发,则希望测试成为表明软件产品中不存在错误 目标 的过程,验证该软件已正确地实现了用户的要求,确立人们对软件质量 的信心。 1.测试的目标和衡量标准 功能、安全、用户体验、兼容性、性能、可靠 衡量标准 从技术角度 黑盒测试(需求规格说明书)、白盒测试、灰盒测试、ALAC测试 2.测试的分类(或组织),各种类型的测试的主要任务及所依赖的文 单元测试(详细设计说明书、源程序代码) 集成测试 (概要设计说明书、详细设计说明书) 从执行阶段角度 确认测试 (需求设计说明书、概要设计说明书) 系统测试(系统的需求规格说明书) 已知产品的功能设计规格,可以通过测试证明每个实现了的功能是否符 黑盒测试 合要求 已知产品的内部工作过程,可以通过测试证明每种内部操作是否符合设 计规格要求, 所有内部成分是否已经过检查 3.机器测试:在设定的测试数据上执行被测程序的过程。又称动态测试 语句覆盖:每一可执行语句至少执行一次。(最弱的逻辑覆盖) 判定覆盖(分支覆盖):每个判断的取真分支和取假分支至少经历一次 条件覆盖:每个判断的每个条件的可能取值至少执行一次。(满足条件 白盒测试 (结构测试、逻辑驱动测试) 覆盖,并不一定能满足分支覆盖) 逻辑覆盖 判定—条件覆盖:每个条件的所有可能取值至少执行一次,同时每个 6.测试 判断本身的所有可能判断结果至少执行一次。 条件组合覆盖:每个判断的所有可能的条件取值组合至少执行一次。 测试用例设计 路径覆盖:覆盖程序中所有可能的路径 是在程序控制流图的基础上,通过分析控制构造的环路复杂性,导出基 本可执行路径集合,从而设计测试用例的方法。 基本路径覆盖 设计出的测试用例要保证在测试中程序的每一个可执行语句至少执行一 人工测试: 采用人工方法进行, 目的在于检查程序的静态结构, 找出编 译不能发现的错误。 4. 单元测试的主要内容。 5. 集成测试的类型及主要的测试策略。 6. 确认测试的内容。 7.了解测试计划的主要内容。 8.测试系统中的测试过程:功能测试、性能测试、验收(或确认)测 试、安装测试,以及它们的内容。 1. 维护活动的类型:改正性、适应性、完善性、预防性。 为了识别和纠正软件错误、改正软件性能上的缺陷、排除实施中的误使 用,应进行的诊断和改正错误的过程。 改正性维护 随着信息技术的飞速发展,软件运行的外部环境(新的硬、软件配置) 或数据环境(数据库、数据格式、数据输入/输出方式、数据存储介 质)可能发生变化,为了使软件适应这种变化,而修改软件的过程叫做 适应性维护 适应性维护 2. 各种维护活动的主要内容和目标 为了满足新的功能与性能要求,需要修改或再开发软件,以扩充软件 功能、增强软件性能、改进加工效率、提高软件的可维护性。 完善性维护 7.系统维护 是指把今天的方法学用于昨天的系统以满足明天的需要。也就是说,采 用先进的软件工程方法对需要维护的软件或软件中的某一部分 (重新) 进行设计、编码和测试。 预防性维护 软件再工程是指通过对目标系统的检查和改造,其中包括设计恢复(库 存目录分析)、再文档、逆向工程、程序和数据重构以及正向工程等一 系列活动,旨在将逆向工程、重构和正向工程组合起来,将现存系统重 新构造为新的形式,以开发出质量更高、维护性更好的软件。 再工程 是指通过分析目标系统来标识系统的部件及其交互关系,然后使用其他 3.软件再生: 文档重构、重组、逆向工程、再工程, 以及它们各自的内 形式或更高层的抽象创建系统表现的过程。凡是在软件生命周期内将软 逆向工程 件的某种形式描述转换成更为抽象形式的活动都可称为逆向工程 容和含义。 就是在不改变代码外在行为的前提下,修改代码来改进程序的内部结 软件工程 构,提高程序的可理解性和可维护性,进而帮助尽早发现缺陷,提高编 重构 程速度 SRP, OCP, LSP, DIP, ISP, LoD 指一个类的功能要单一,不能包罗万象。 SRP (单一职责原则) 一个模块在扩展性方面应该是开放的而在更改性方面应该是封闭的。 OCP (开放封闭原则) 子类应当可以替换父类并出现在父类能够出现的任何地方。 LSP (里式替换原则) 高层次的模块不应该依赖于低层次的模块,他们都应该依赖于抽象。 抽象不应该依赖于具体实现,具体实现应该依赖于抽象。 1. 主要的面向对象设计原则及各自的原理: 8.其他 DIP (依赖倒置原则) 高层模块就是调用端,底层模块就是具体实现类。 (应该让底层模块定 义抽象接口并且实现,让高层模块调用抽象接口,而不是直接调用实现 模块间要通过抽象接口隔离开,而不是通过具体的类强耦合起来,即面 向接口编程。 ISP (接口分离原则) 类间解耦 一个类对自己依赖的类知道的越少越好。软件编程的总的原则: 低耦 LoD (迪米特法则) 合, 高内聚。无论是面向过程编程还是面向对象编程, 只有使各个模块 之间的耦合尽量的低,才能提高代码的复用率。 1. 白盒测试主要包括哪几种?各自的测试内容是什么? 2.软件项目管理中用于估计工作量的指标主要包括哪些? 表达逻辑判断的工具 判定表 以词条方式定义在数据模型、功能模型和行为模型中出现的数据对象及 控制信息的特性,给出它们的准确定义,包括数据流、加工、数据文 数据字典 件、数据元素,以及数据源点、数据汇点等。 是在层次结构图的基础上推出的一种描述系统结构和模块内部处理功能 的工具,用来说明每个模块的输入、输出数据和数据加工的重要工具。 3. 判定表、数据字典、IPO图、PDL、DFD图、甘特图等常用的图形的 IPO图 (输入/处理/输出图) 主要作用是什么?哪些经常在结构化方法中使用? PDL 一种语言 是描述系统中数据流程的一种图形工具,以图形的方式描绘数据在系统 中流动和处理的过程,它标志了一个系统的逻辑输入和逻辑输出,以及 DFD 把逻辑输入转换逻辑输出所需的加工处理。 表示工作进度计划以及工作实际进度情况最为简明的图示方法。通过条 状图来显示项目、进度和其他时间相关的系统进展的内在关系随着时间 甘特图 进展的情况。 软件过程 软件也有一个孕育、诞生、成长、成熟和衰亡的生存过程, 我们称这个 过程为软件生命周期或软件生存期。 9.补充提纲 软件生命周期 概括地说,软件工程是指导计算机软件开发和维护的工程学科。采用工 程的概念、原理、技术和方法来开发与维护软件, 把经过时间考验而证 明正确的管理技术和当前能够得到的最好的技术方法结合起来,以经济 地开发出高质量的软件并有效地维护它,这就是软件工程。 软件工程 瀑布模型是一个项目开发架构,开发过程是通过设计一系列阶段顺序展 4. 软件过程、软件生命周期、软件工程、瀑布模型、原型法、里程碑 开的,从系统需求分析开始直到产品发布和维护,每个阶段都会产生循 等基本概念。 环反馈,项目开发进程从一个阶段"流动"到下一个阶段。 瀑布模型 原型法是指在获取一组基本的需求定义后,利用高级软件工具可视化的 开发环境,快速地建立一个目标系统的最初版本,并把它交给用户试 用、补充和修改,再进行新的版本开发。反复进行这个过程,直到得出 系统的"精确解",即用户满意为止。 快速原型是快速建立起来的可以在计算机上运行的程序,它所能完成的 原型法 功能往往是最终产品能完成的功能的一个子集

基线是软件生存期各开发阶段末尾的特定点,也被称为里程碑

里程碑

5. 常用的质量保障措施和方法有哪些?

从用户的角度出发,普遍希望通过软件测试检验软件中隐藏的错误和缺

陷,以考虑是否可以接受该产品。