x 可靠性设计

历年真题

试题三 论软件的可靠性设计

现代军事和商用系统中，随着系统中软件成分的不断增加，系统对软件的依赖性越来越强。软件可靠性已成为软件设计过程中不可或缺的重要组成部分。实践证明，保障软件可靠性域有效、最经济、最重要的手段是在软件设计阶段采取措施进行可靠性控制，由此提出了可靠性设计的概念。可靠性设计就是在常规的软件设计中，应用各种方法和技术，使程序设计在兼顾用户的功能和性能需求的同时，全面满足软件的可靠性要求。

请围绕“软件的可靠性设计”论题，依次从以下三个方面进行论述。

1.概要叙述你参与管理和开发的软件项目以及你在其中所担任的主要工作。

2.简要说明目前比较主流的软件可靠性设计技术，结合项目实际情况，阐述所选择的可靠性设计技术及其原因。

3.结合你具体参与管理和开发的实际项目，举例说明所选取的软件可靠性技术的具体实施过程，并详细分析实施效果。

1. 概要叙述你参与管理和开发的软件项目以及你在其中所承担的主要工作。

2. 结合项目实际，论述你在项目开发过程中，进行软件可靠性设计时遵循的基本原则；论述你在该项目中所采用的具体可靠性设计技术。

3. 阐述你在具体的可靠性设计工作中，为了分析影响软件可靠性的主要因素，所采用的可靠性分析方法。

可靠性的主要影响因素

运行环境（软件可靠性的定义是相对于运行环境的)

软件的可靠性投入

软件的开发方法和开发环境

软件规模

软件内部结构（内部结构越复杂，包含的缺陷数就可能越多）

可靠性设计需要遵循的原则

1. 软件可靠性设计是软件设计的一部分，必须在软件的总体设计框架中使用，并且不能与其他设计原则相冲突。

2. 软件可靠性设计在满足提高软件质量要求的前提下，以提高和保障软件可靠性为最终目标。

3. 软件可靠性设计应确定软件的可靠性目标，不能无限扩大，并且排在功能、用户需求、开发费用之后考虑。

可靠性设计技术

一般来说，被认可的且具有应用前景的软件可靠性设计技术主要有容错设计、检错设计和降低复杂度设计等技术。

1.容错设计技术

对于软件失效后果特别严重的场合，如尤机的飞行控制系统、空中交通管制系统及核反应堆安全控制系统等，可采用容错设计方法。常用的软件容错技术主要有恢复块设计、N版本程序设计和冗余设计三种方法。

(1)恢复块设计

恢复块设计就是选择一组操作作为容错设计单元，从而把普通的程序块变成恢复块。一个恢复块包含若干个功能相同、设计差异的程序块文本，每一时刻有一个文本处于运行状态。一旦该文本出现故障，则用备份文本加以替换，从而构成“动态冗余”。

(2)N版本程序设计

N版本程序的核心是通过设计出多个模块或不同版本，对于相同初始条件和相同输入的操作结果，实行多数表决，防止其中某一模块/版本的故障提供错误的服务，以实现软件容错。

(3)冗余设计

软件冗余设计技术实现的原理是在一套完整的软件系统之外，设计一种不同路径、不同算法或不同实现方法的模块或系统作为备份，在出现故障时可以使用冗余的部分进行替换，从而维持软件系统的正常运行。

2.检错设计

在软件系统中，对无需在线容错的地方或不能采用冗余设计技术的部分，如果对可靠性要求较高，故障有可能导致严重的后果。这时一般采用检错技术，在软件出现故障后能及时发现并报警，提醒维护人员进行处理。

采用检错设计技术需要着重考虑几个要素：检测对象、检测延时、实现方式和处理方式。

(1)检测对象，即检测点和检测内容。在设计时应考虑把检测点放在容易出错的地方和出错对软件系统影响较大的地方，检测内容选取那些有代表性的、易于判断的指标。

(2)检测延时，在软件检错设计时要充分考虑到检测延时，如果延时长到影响故障的及时报警，则需要更换检测对象或检测方式。

(3)实现方式，最直接的一种实现方式是判断返回结果，如果返回结果超出正常范围，则进行异常处理。计算运行时间也是一种常用的技术，如果某个模块或函数运行超过预期的时间，可以判断出现故障。另外，还有置状态标志位等多种方法，自检的实现方式要根据实际情况来选用^

(4)处理方式。大多数检测采用“查出故障一停止软件系统运行一报警”的处理方式，但也有采用不停止或部分停止软件系统运行的情况，这一般由故障是否需要实时处理来决定。

3.降低复杂度设计

降低复杂度设计的思想就是在保证实现软件功能的基础上，简化软件结构，缩短程序代码长度，优化软件数据流向，降低软件复杂度，从而提高软件可靠性。

4. 分析

除了容错设计、检错设计和降低复杂度设计技术外，人们尝试着把硬件可靠性设计中比较成熟的技术，如故障树分析（FTA）、失效模式与影响分析（FMEA）等运用到软件可靠性设计领域，这些技术大多数运用一些分析、预测技术，在软件设计时就充分考虑影响软件可靠性的因素，并采取一些措施进行优化。

可靠性分析方法

在软件可靠性设计之前和软件可靠性设计过程中，都需要采用软件可靠性分析和预测方法，来确定当前系统中的主要可靠性因素和目标。常见的软件可靠性分析方法包括故障树分析方法、失效模式与效应分析方法等。

故障树分析方法：一种自顶向下的软件可靠性分析方法，即从软件系统不希望发生的事件（顶事件），特别是对人员和设备的安全及可靠性产生重大影响的事件开始，向下逐步追查导致顶事件发生的原因，直至基本事件（底事件），从而确定软件故障原因的各种可能组合方式和（或）发生概率。基本的步骤是软件故障树的建立、定性分析和定量分析。

失效模式与影响分析方法：在产品设计阶段和过程设计阶段，对构成产品的子系统、零件，对构成过程的各个工序逐一进行分析，找出所有潜在的失效模式，并分析其产生原因和可能的后果，从而预先采取必要的措施，以提高产品的质量和可靠性的一种系统化的活动。

可靠性模型

当前的软件可靠性模型众多，但并没有一个最好的或者可以适用所有软件系统的软件可靠性模型，因此对于不同的软件系统，出于不同的可靠性分析目的，需要选择合适的软件可靠性模型。常见的 10 类软件可靠性模型有**种子法模型、失效率类模型、曲线拟合类模型、可靠性增长模型、程序结构分析模型、输入域分类模型、执行路径分析方法模型、非齐次泊松过程模型、马尔可夫过程模型和贝叶斯分析模型**。

可靠性模型选择考虑因素

(1)假设的适用性：模型假设是可靠性模型的基础，模型假设需要符合软件系统的现有状况，在软件系统中与假设冲突的因素达到几乎不存在的程度。往往一个模型的假设有很多，需要在选择模型时对每一条假设进行分析，评估现有软件系统中不符合假设的因素对可靠性评价有多大影响，以确定模型是否符合软件系统的可靠性评价工作。

(2)预测的能力与质量：预测的能力和质量是指模型根据现在和历史的可靠性数据，预测将来的可靠性和失效概率的能力，以及预测结果的准确程度。因此，应尽可能选择比较成熟的、应用较广的模型。

(3)输出值能否满足可靠性评价需求：根据可靠性测试目的来确定哪些模型的输出值满足可靠性评价需求。重要的可靠性定量指标包括：当前可靠度、平均无失效时间、故障密度、期望达到规定可靠性目标的日期、达到规定可靠性目标的成本要求等。

(4)使用的简便性：模型使用的数据在软件系统中易于收集；模型应该简单易懂；模型应该便于使用，最好有工具支持。

可靠性数据收集和处理问题

(1)可靠性数据规范不一致，对软件进行度量的定义混乱；

(2)数据收集过程存在于整个软件生命周期，但由于成本等因素，其连续性往往不能保证

(3)数据完整性不能保证，收集到的数据大多数是不完全的；

(4)数据质量和准确性不能保证；

(5)缺乏有效的技术和工具支持，难以进行自动分析；

(6)缺乏可靠性数据的交流与共享。

解决方法主要有：

(1) 尽早确定可靠性模型，明确要搜集的可靠性数据，确定涉及的术语、记录方法等；

(2) 制定可实施的可靠性数据搜集计划，并指定专人负责。保证数据的收集和验证与软件开发过程同步进行；

(3) 重视软件测试特别是可靠性测试产生的测试结果的整理和分析；

(4) 尽可能地利用工具进行收集工作，例如利用数据库进行存储和分析等。

我的范文

摘要

正文

从三个方面进行设计以提高系统的可靠性. 采用多种架构风格和设计模式, 以降低软件的复杂度; 采用 N 版本程序设计, 提高系统的容错能力; 采用业务数据校验机制, 提高系统的检错能力.

可靠性设计是软件设计的一部分, 我们在软件设计之前以及设计过程中都充分考虑了软件**架构风格**对可靠性的影响, 也在架构评估过程中将可靠性作为重要的质量属性评价指标. 架构风格是指软件系统组织方式的惯用模式, 其中组织方式 XXX.... 在交通信息平台的设计过程中, 我们采用了层次系统、数据库系统、基于事件系统以及管道过滤器等架构风格. (如层次系统: 依据层次系统风格的思想, 我们将系统分为 XXX 4 层, 每一层只与其上一层和下一层交互, 从粗粒度层面极大程度降低了系统耦合度, 各层可根据业务特征, 灵活采取相应的增强可行性的措施).

在各功能模块的设计过程中, 我们采用了抽象工厂、单例模式、适配器模式、责任链模式、策略模式等多种**设计模式**. 设计模式是一套被反复使用、多数人知道的、经过分类编目的、代码设计经验的总结。由于系统涉及多种数据库的连接和操作, 因此利用抽象工厂模式简化该操作. 分别为 DB2..XXX(具体怎么做). 该设计有效降低了系统的复杂度, 提高了系统可靠性.

为提高路况数据处理的可靠性, 我们采用了 N 版本程序设计. 路况根据车辆运行速度分为拥堵、缓行和畅通三个级别, 计算车速需要融合监控摄像头、雷达、GPS 等多种数据, 且

融合方式和计算方法多种多样，每种方法各有优缺点，在不同的交通状况下其可靠性也不同. 鉴于此，我们利用 N 版本程序设计的方法，让 3 个不同的开发团队利用不同的技术和算法分别判断路况等级，通过多数表决的方法输出最终展示给用户的路况等级. 该方法在单点路况判断失误甚至失效的情况下，通过多数表决的方法，赋予系统较强的容错能力，大大提高了路况判断的可靠性.

为提高 GPS 数据处理的检错能力以增强可靠性，我们采用了数据校核机制. 理想的车辆 GPS 数据应按照固定的时间周期返回车辆当前所在位置的经纬度坐标信息，然而由于车辆所处地理环境的复杂性以及硬件设备在采集和传输环节的误差，会出现 GPS 坐标点异常的情况，具体表现为位置异常(如车辆定位与湖泊)和车速异常(如车速为 200km/h). 针对这些情况，我们首先明确了检错对象为某车辆连续时间内的多个 GPS 数据，然后确定了合理的位置分布范围、位移跨度和速度区间范围作为检错实现方式. 在可接受的延迟时间内，对 GPS 数据逐个进行分析比对，当异常数据量较小时，可忽略异常数据或用插值法进行数据修复；当出现大量异常数据时，则停止该模块的运行，并报警通知相应人员进行处理. 我们制定的数据校核机制有效的实现了检错，提高了 GPS 数据处理的可靠性.

此外，我们还使用了**看门狗软件**检测系统的死循环，使用**心跳机制**监测服务是否正常运行等措施，进一步提高了系统可靠性.

发布以来，运行平稳，可靠性高.