极客时间训练营

系统架构设计师

第9章 系统可靠性设计

授课：王建平

目录

目录

软件可靠性基本概念

◆软件可靠性是软件产品在规定的条件下和规定的时间区间完成规定功能的能力。（★）
◆软件可靠性和硬件可靠性区别（★）
（1）复杂性：软件复杂性比硬件高，大部分失效来自于软件失效。
（2）物理退化：硬件失效主要是物理退化所致，软件不存在物理退化。
（3）唯一性：软件是唯一的，每个COPY版本都一样，而两个硬件不可能完全一样。
（4）版本更新周期：硬件较慢，软件较快。
◆软件可靠性的定量描述（★★★）
1.规定时间：自然时间、运行时间、执行时间（占用CPU）。
2.失效概率：从软件运行开始，到某一时刻t为止，出现失效的概率。可以看作是关于软件运行时间的一个随机函数。用F(t)表示。或指单位时间内失效的元件数与元件总数的比例。通常用λ表示。
3.可靠度：软件系统在规定的条件下、规定的时间内不发生失效的概率。等于1-失效概率。
4.失效强度：单位时间软件系统出现失效的概率。
5.平均失效前时间（MTTF）：平均无故障时间，发生故障前正常运行的时间。
6.平均恢复前时间（MTTR）：平均故障修复时间，发生故障后的修复时间。
7.平均故障间隔时间（MTBF）：失效或维护中所需的平均时间，包括故障时间以及检测和维护设备的时间。MTBF=MTTF+MTTR。

◆系统可用性=MTTF/(MTTF+MTTR)*100%。

典型真题

例：系统 ( )是指在规定的时间内和规定条件下能有效地实现规定功能的能力。它不仅取决于规定的使用条件等因素，还与设计技术有关。常用的度量指标主要有故障率(或失效率)、平均失效等待时间、平均失效间隔时间和可靠度等。其中，( )是系统在规定工作时间内无故障的概率。

A.可靠性.　　B.可用性　　C.可理解性　　D.可测试性

A.失效率　　B.平均失效等待时间　　C.平均失效间隔时间　　D.可靠度.

参考答案：A D

软件可靠性目标

可靠性目标（★）

◆使用失效强度来表示软件缺陷对软件运行的影响程度。然而在实际情况中，对软件运行的影响程度不仅取决于软件失效发生的概率，还和软件失效的严重程度有很大关系。这里引出另外一个概念--失效严重程度类。

◆失效严重程度类就是对用户具有相同程度影响的失效集合。

对失效严重程度的分级可以按照不同的标准进行，常见的是按对成本影响、对系统能力的影响等标准划分。

◆概率和影响两个方面影响。

表 9-1 给出了一个按照对成本的影响划分失效严重程度类的例子，这个例子涉及的软件系统是某电子商务运营系统。

表 9-1　按照对成本的影响划分失效严重程度类

失效严重程度类	定义（人民币万元）	失效严重程度类	定义（人民币万元）
1	成本＞ 100	4	0.1 ＜成本≤ 1
2	10 ＜成本≤ 100	5	成本＜ 0.1
3	1 ＜成本≤ 10		

表 9-2　按照对系统能力的影响划分失效严重程度类

失效严重程度类	定义
1	系统崩溃，重要数据不可恢复
2	系统出错停止响应，重要数据可恢复
3	用户重要操作无响应，可恢复
4	部分操作无响应，但可用其他操作方式替代

软件可靠性测试

可靠性测试

◆广义的软件可靠性测试是指为了最终评价软件系统的可靠性而运用建模、统计、试验、分析和评价等一系列手段对软件系统实施的一种测试。（★）

◆一个完整的软件可靠性测试包括：

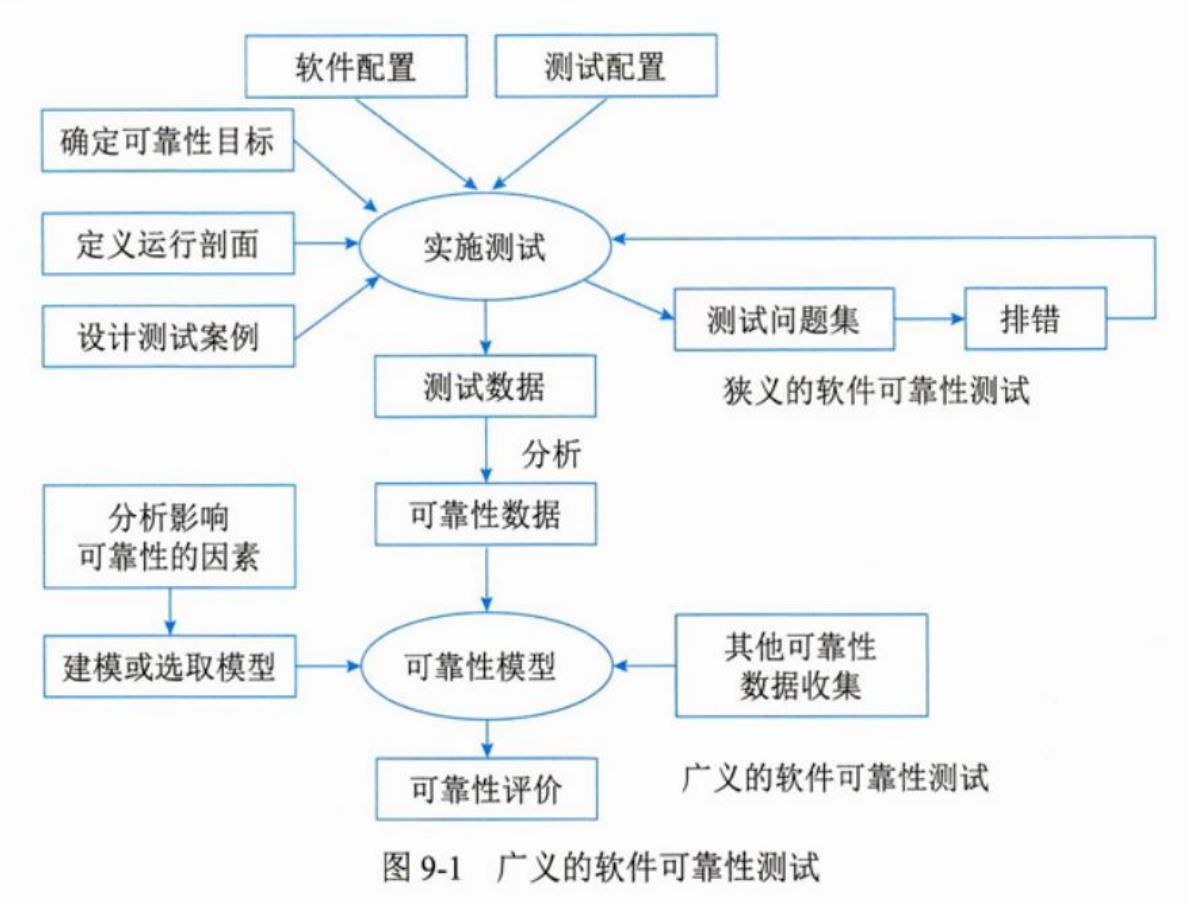


图 9-1　广义的软件可靠性测试

◆狭义的软件可靠性测试是指为了获取可靠性数据，按预先确定的测试用例，在软件的预期使用环境中，对软件实施的一种测试。狭义的软件可靠性测试也叫“软件可靠性试验”,它是面向缺陷的测试。（★）

◆可靠性测试的目的可归纳为以下3个方面：（★）

(1)发现软件系统在需求、设计、编码、测试和实施等方面的缺陷。

(2)为软件的使用和维护提供可靠性数据。

(3)确认软件是否达到可靠性的定量要求。

目录

影响软件可靠性的因素

◆影响软件可靠性的因素（★）
软件可靠性模型是指为预计或估算软件的可靠性所建立的可靠性框图和数学模型，建立可靠性模型是为了将复杂系统的可靠性逐级分解为简单系统的可靠性，以便于定量预计、分配、估算和评价复杂系统的可靠性。

◆从技术的角度来看，影响软件可靠性的因素如下：（★★★）
(1)运行剖面(环境)
(2)软件规模
(3)软件内部结构
(4)软件的开发方法和开发环境
(5)软件的可靠性投入

软件可靠性模型分类

◆软件可靠性模型分类（★★）

- 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
- 贝叶斯分析模型

软件可靠性模型分类

◆种子法模型：利用捕获一再捕获抽样技术估计程序中的错误数，在程序中预先有意“播种”一些设定的错误“种子”,然后根据测试出的原始错误数和发现的诱导错误的比例，来估计程序中残留的错误数。
其优点是简便易行，缺点是诱导错误的“种子”与实际的原始错误之间的类比性估量困难。

◆失效率类模型：用来研究程序的失效率。

◆曲线拟合类模型：用回归分析的方法研究软件复杂性、程序中的缺陷数、失效率、失效间隔时间，包括参数方法和非参数方法两种。

◆可靠性增长模型：这类模型预测软件在检错过程中的可靠性改进，用增长函数来描述软件的改进过程。

◆程序结构分析模型：是根据程序、子程序及其相互间的调用关系，形成一个可靠性分析网络。

◆输入域分类模型：是选取软件输入域中的某些样本“点”运行程序，根据这些样本点在“实际”使用环境中的使用概率的测试运行时的成功/失效率，推断软件的使用可靠性。这类模型的重点(亦是难点)是输入域的概率分布的确定及对软件运行剖面的正确描述。

软件可靠性模型分类

◆执行路径分析方法类模型：先计算程序各逻辑路径的执行概率和程序中错误路径的执行概率，再综合出该软件的使用可靠性。Shooman分解模型属于此类。

◆非齐次泊松过程模型(NHPP)：是以软件测试过程中单位时间的失效次数为独立泊松随机变量，来预测在今后软件的某使用时间点的累计失效数。

◆马尔可夫过程模型如下。
- 完全改错的线性死亡模型。
- 不完全改错的线性死亡模型。
- 完全改错的非静态线性死亡模型。

◆贝叶斯模型是利用失效率的试验前分布和当前的测试失效信息，来评估软件的可靠性。这是一类当软件可靠性工程师对软件的开发过程有充分地了解，软件的继承性比较好时具有良好效果的可靠性分析模型。

软件可靠性管理

◆软件可靠性管理是软件工程管理的一部分，它以全面提高和保证软件可靠性为目标，以软件可靠性活动为主要对象，是把现代管理理论用于软件生命周期中的可靠性保障活动的一种管理形式。

◆软件可靠性活动是贯穿于软件开发全过程的。（★）

1. 需求分析阶段

（1）确定软件的可靠性目标。
（2）分析可能影响可靠性的因素。
（3）确定可靠性的验收标准。
（4）制定可靠性管理框架。
（5）制定可靠性文档编写规范。
（6）制订可靠性活动初步计划。
（7）确定可靠性数据收集规范。

2. 概要设计阶段

（1）确定可靠性度量。
（2）制定详细的可靠性验收方案。
（3）可靠性设计。
（4）收集可靠性数据。
（5）调整可靠性活动计划。
（6）明确后续阶段的可靠性活动的详细计划。
（7）编制可靠性文档。

3. 详细设计阶段

（1）可靠性设计。
（2）可靠性预测（确定可靠性度量估计值）。
（3）调整可靠性活动计划。
（4）收集可靠性数据。
（5）明确后续阶段的可靠性活动的详细计划。
（6）编制可靠性文档。

4. 编码阶段

（1）可靠性测试（含于单元测试）。
（2）排错。
（3）调整可靠性活动计划。
（4）收集可靠性数据。
（5）明确后续阶段的可靠性活动的详细计划。
（6）编制可靠性文档。

5. 测试阶段

（1）可靠性测试（含于集成测试、系统测试）。
（2）排错。
（3）可靠性建模。
（4）可靠性评价。
（5）调整可靠性活动计划。
（6）收集可靠性数据。
（7）明确后续阶段的可靠性活动的详细计划。
（8）编制可靠性文档。

6. 实施阶段

（1）可靠性测试（含于验收测试）。
（2）排错。
（3）收集可靠性数据。
（4）调整可靠性模型。
（5）可靠性评价。
（6）编制可靠性文档。

典型真题

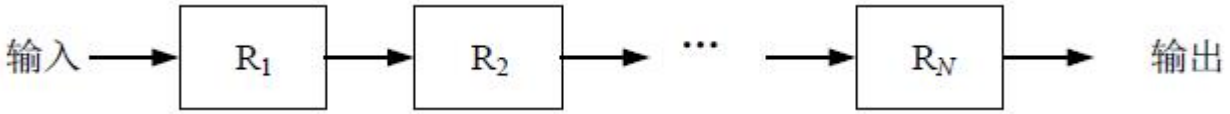
在可靠性模型中（）模型是用回归分析的方法研究软件复杂性、程序中的缺陷数、失效率、失效间隔时间。（）模型是用增长函数来描述软件的改进过程。

A.种子法模型　　B.失效率类模型　　C.曲线拟合类模型.　　D.可靠性增长模型
A.种子法模型　　B.失效率类模型　　C.曲线拟合类模型　　D.可靠性增长模型.
参考答案：C D

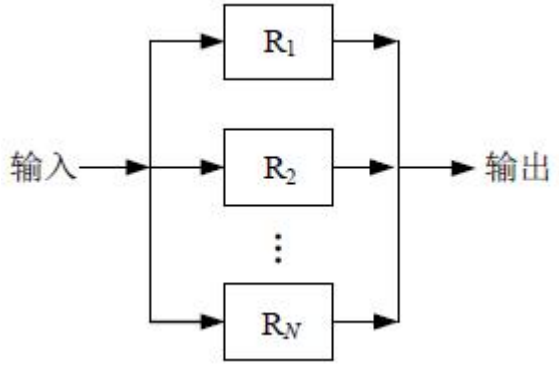
确定可靠性验收标准是（）阶段的任务。
A.需求分析　　B.概要设计　　C.详细设计　　D.实施阶段
参考答案：A

软件可靠性模型-补

◆串并联系统可靠性
◆无论什么系统，都是由多个设备组成的，协同工作，而这多个设备的组合方式可以是串联、并联，也可以是混合模式，假设每个设备的可靠性为R1，R2.....Rn，则不同的系统的可靠性公式如下：
◆串联系统，一个设备不可靠，整个系统崩溃，整个系统可靠性R=R1 * R2 * ... * Rn。

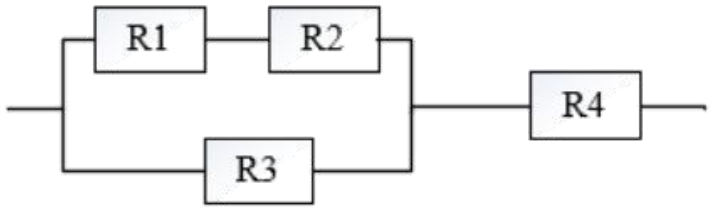


◆并联系统，所有设备都不可靠，整个系统才崩溃，整个系统可靠性R=1-(1-R1) * (1-R2) * ... * (1-Rn)。



典型真题

某计算机系统的可靠性结构如下所示，若所构成系统的每个部件的可靠度分别为R1、R2、R3和R4，则该系统的可靠度为（ 填空题 ）。



目录

软件可靠性设计原则及技术

◆软件可靠性设计原则：（★★）
（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，不能无限扩大化，并且排在功能度、用户需求和开发费用之**后**考虑。

◆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。（★★★）

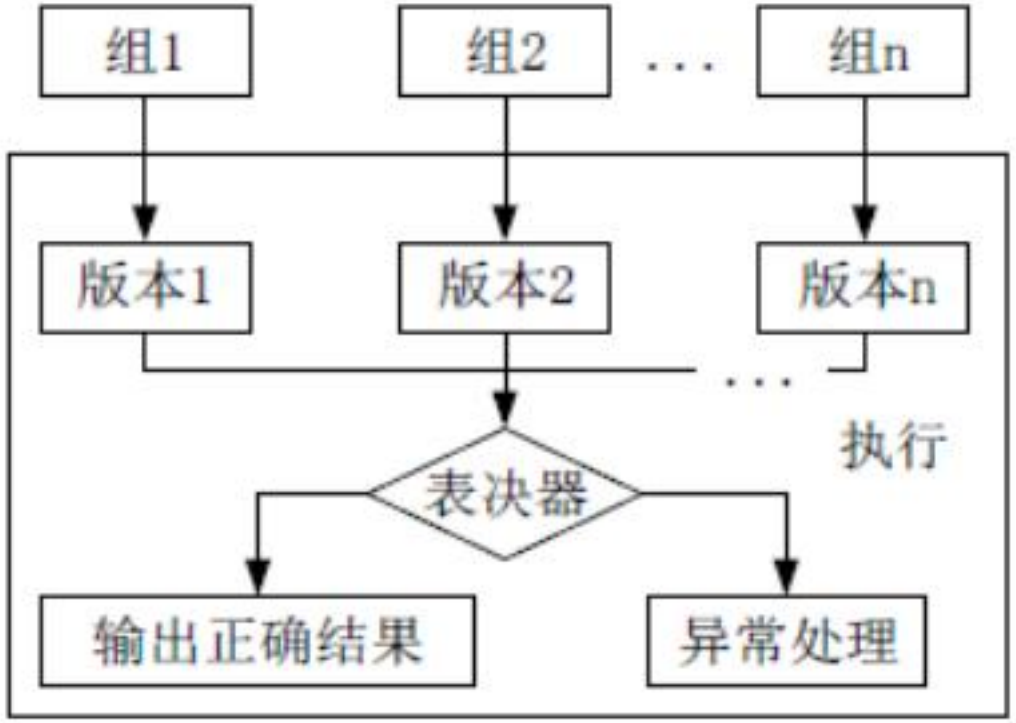
◆容错设计技术：常用的软件容错技术主要有：（★★★）
1.恢复块设计
2.N版本程序设计
3.冗余设计

软件容错技术

◆N版本程序设计：（★★★）

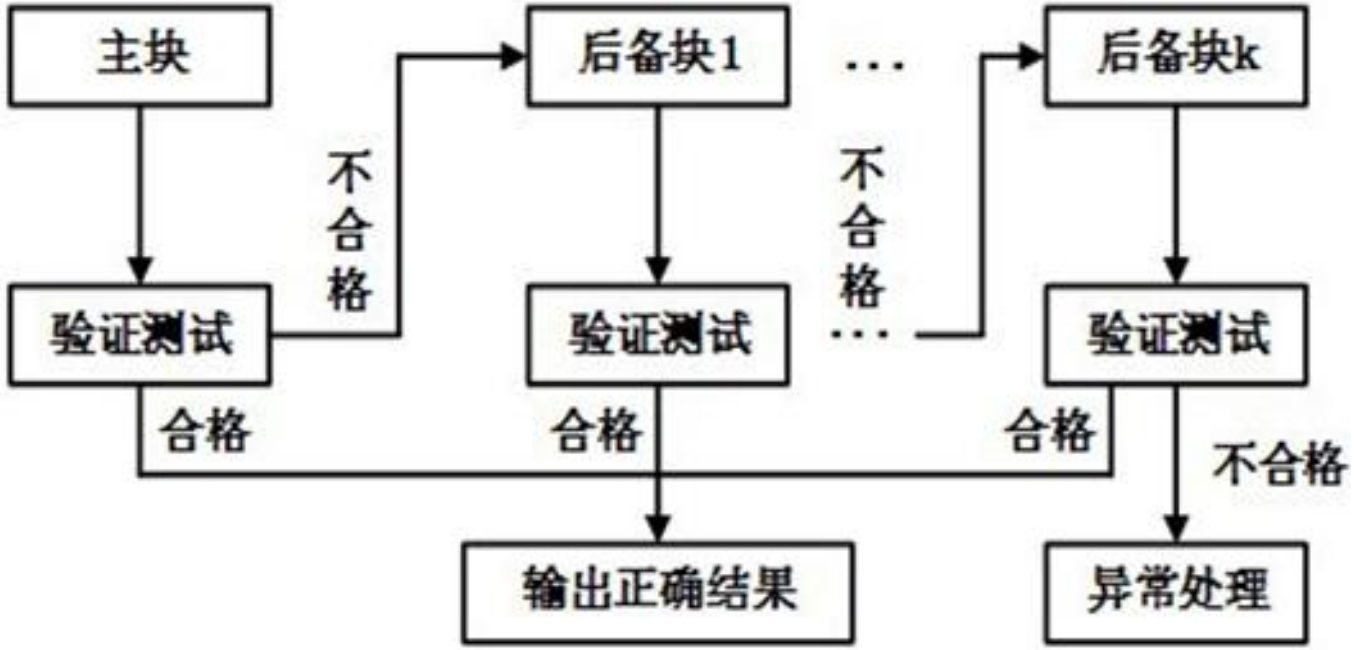
N版本程序设计是一种静态的故障屏蔽技术，采用前向恢复的策略，其设计思想是用N个具有相同功能的程序同时执行一项计算，结果通过多数表决来选择。

其中N个版本的程序必须由不同的人独立设计，使用不同的方法、设计语言、开发环境和工具来实现，目的是减少N个版本的程序在表决点上相关错误的概率。



软件容错技术

◆恢复块设计（动态冗余）：恢复块方法是一种动态的故障屏蔽技术，采用后向恢复策略。它提供具有相同功能的主块和几个后备块，一个块就是一个执行完整的程序段，主块首先投入运行，结束后进行验证测试，如果没有通过验证测试，系统经现场恢复后由一后备块运行。
被选择用来构建恢复块的程序块可以是模块、过程、子程序和程序段。（★★★）



软件容错技术

◆二者比较（★★★）

表 19-1 恢复块方法与N版本程序设计的比较

	恢复块方法	N 版本程序设计
硬件运行环境	单机	多机
错误检测方法	验证测试程序	表决
恢复策略	后向恢复	前向恢复
实时性	差	好

◆冗余设计（★★★）

软件的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。（双机热备、热可替换等）

软件可靠性设计技术

◆检错技术（★）

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。

检错技术实现的代价一般低于容错技术和冗余技术，但它有一个明显的缺点，就是不能自动解决故障，出现故障后如果不进行人工干预，将最终导致软件系统不能正常运行。

采用检错设计技术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。

◆降低复杂度（★）

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。

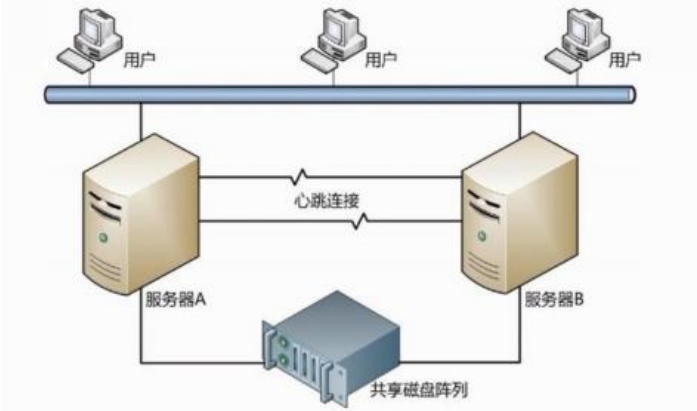
软件复杂性分为模块复杂性和结构复杂性。

- 模块复杂性主要包含模块内部数据流向和程序长度两个方面。
- 结构复杂性用不同模块之间的关联程度来表示。

系统配置技术

双机热备技术（★★★）
◆双机热备技术是一种软硬件结合的较高容错应用方案。该方案由两台服务器系统和一个外接共享磁盘阵列柜和相应的双机热备份软件组成。操作系统和应用程序安装在两台服务器的本地系统盘上，整个网络系统的数据是通过磁盘阵列集中管理和数据备份的。用户的数据存放在外接共享磁盘阵列中，当一台服务器出现故障时，备机主动替代主机工作，保证网络服务不间断。

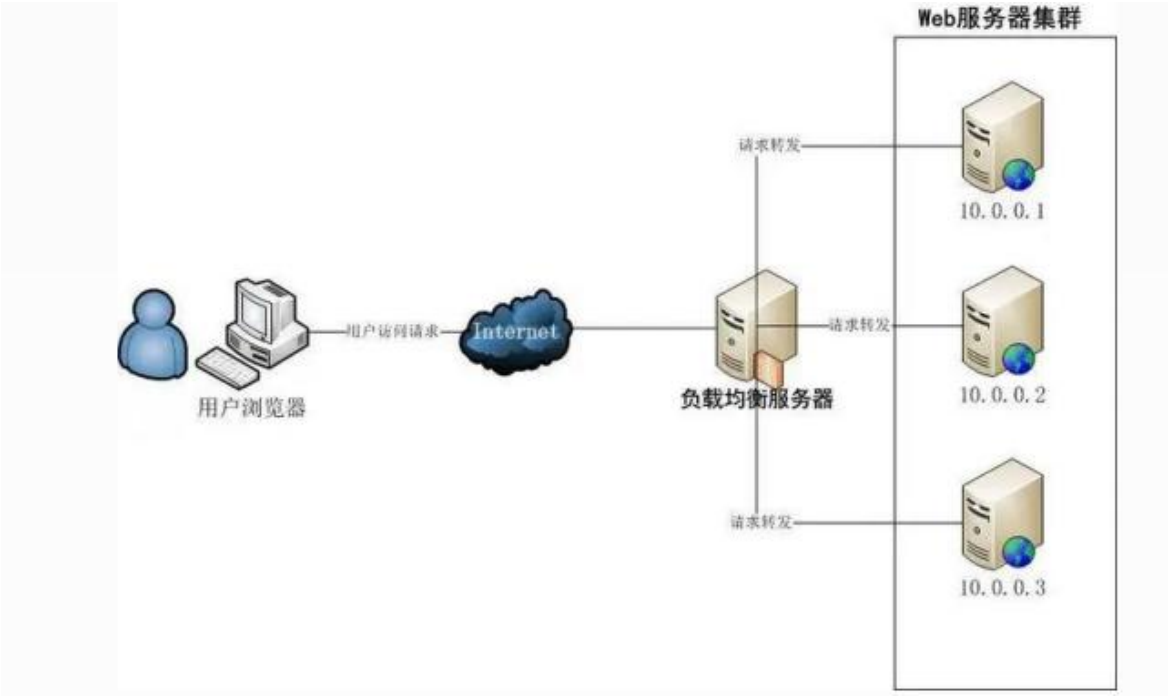
◆双机热备方案中，根据两台服务器的工作方式可以有3种不同的工作模式（★★★）
●双机热备模式(Active/Standby)--一个工作一个备份但有资源浪费
●双机互备模式---两个都在运行但是不同服务，当心不跳时候另一个接管两个服务。
●双机双工模式(实现负载均衡)---两个都在运行相同的服务



服务器集群技术

服务器集群技术（★★）

◆集群技术是指一组相互独立的服务器在网络中组合成为单一的系统工作，并以单一系统的模式加以管理。集群中所有的计算机拥有一个共同的名称，集群内任一系统上运行的服务可被所有的网络客户所使用。当某结点服务器发生故障时，这台服务器上所运行的应用程序将在另一结点服务器上被自动接管。



典型真题

软件复杂性分为模块复杂性和( ) 。( )是不同模块之间的关联程度的表示。

A.结构复杂性　　B.软件复杂性　　C.模块复杂度　　D.程序复杂性

A.结构复杂性　　B.软件复杂性　　C.模块复杂度　　D.程序复杂性

参考答案：AA

目录

软件可靠性测试概述

◆软件可靠性测试概述

传统的软件测试是面向错误的测试，测试所得的数据不能直接用于软件可靠性评价，必须经过一定的分析处理后方可使用可靠性模型进行可靠性评价。

软件可靠性测试由可靠性目标的确定、运行剖面的开发、测试用例的设计、测试实施、测试结果的分析等活动组成。（★★）

◆定义软件运行剖面（★）

定义运行剖面首先需要为软件的使用行为建模，然后是开发使用模型，明确需要测试的内容。

定义使用概率的最佳方法是使用实际的用户数据，如来自系统原型、前一版本的使用数据；

其次是由该软件应用领域的用户和专家提供的预期使用数据，在没有任何数据可用的情况下，只能是将每个状态现有的弧分配相同的概率，这是最差的一种方法。

由于软件可靠性行为是相对于软件实际的运行剖面而言的，同一软件在不同运行剖面下其可靠性表现可能大不相同，所以用于可靠性测试准备的运行剖面的开发与定义必须充分分析和考虑软件的实际运行情况。

软件可靠性测试概述

◆可靠性测试用例设计（★）
设计测试用例就是针对特定功能或组合功能设计测试方案，并编写成文档。测试用例的选择既要有一般情况，也应有极限情况以及最大和最小的边界值情况。因为测试的目的是暴露应用软件中隐藏的缺陷，所以在设计选取测试用例和数据时要考虑那些易于发现缺陷的测试用例和数据，结合复杂的运行环境，在所有可能的输入条件和输出条件中确定测试数据，来检查应用软件是否都能产生正确的输出。优先测试那些最重要或最频繁使用的功能，释放和缓解最高级别的风险，有助于尽早发现那些对可靠性有最大影响的故障，以保证软件的按期交付。

◆典型的测试用例包括以下内容：（★★）
测试用例标识、被测对象、测试环境及条件、测试输入、操作步骤、预期输出、判断输出结果是否符合标准、测试对象的特殊需求

软件可靠性测试概述

◆可靠性测试的实施（★★）
开发方交付的任何软件文档中与可靠性质量特性有关的部分、程序以及数据都应当按照需求说明和质量需求进行测试。在项目合同、需求说明书和用户文档中规定的所有配置情况下，程序和数据都必须进行测试。
软件可靠性数据是可靠性评价的基础。为了获得更多的可靠性数据，应该使用多台计算机同时运行软件，以增加累计运行时间。
应该按照相关标准的要求，制定和实施软件错误报告和可靠性数据收集、保存、分析和处理的规程，完整、准确地记录软件测试阶段的软件错误报告和收集可靠性数据。

◆测试活动结束后要编写《软件可靠性测试报告》,对测试用例及测试结果在测试报告中加以总结归纳。
测试报告应包括以下内容：（★★★）
- 软件产品标识
- 测试环境配置(硬件和软件)
- 测试依据
- 测试结果
- 测试问题
- 测试时间

系统可靠性评价

◆软件可靠性评价概述

软件可靠性评价工作是指选用或建立合适的可靠性数学模型，运用统计技术和其他手段，对软件可靠性测试和系统运行期间收集的软件失效数据进行处理，并评估和预测软件可靠性的过程。

软件可靠性评价过程包含如下三个方面

1.选择可靠性模型

2.收集可靠性数据

3.可靠性评估和预测

◆选择可靠性模型

对于不同的软件系统，不同的可靠性分析目的，模型的适用性是不一样的。可以从以下四方面进行模型的比较和选择。

1.模型假设的适用性

2.预测的能力与质量

3.模型输出值能否满足可靠性评价需求

4.模型使用的简便性

系统可靠性评价

◆收集可靠性数据（★）

面向缺陷的可靠性测试产生的测试数据经过分析后，可以得到非常有价值的可靠性数据，是可靠性评价所用数据的一个重要来源，这部分数据取决于定义的运行剖面和选取的测试用例集。

可靠性数据主要是指软件失效数据，是软件可靠性评价的基础，主要是在软件测试、实施阶段收集的。在软件工程的需求、设计和开发阶段的可靠性活动，也会产生影响较大的其他可靠性数据。因此，可靠性数据的收集工作是贯穿于整个软件生命周期的。

◆可靠性评估和预测

软件可靠性的评估和预测的主要目的，是为了评估软件系统的可靠性状况和预测将来一段时间的可靠性水平。

目前有不少支持软件可靠性估计的软件工具，只要将收集的失效数据分类、录入合适的可靠性模型，就可以获得软件可靠性的评价结果。

软件可靠性评估和预测以软件可靠性模型分析为主，但也要在模型之外运用一些统计技术和手段对可靠性数据进行分析，作为可靠性模型的补充、完善和修正。

本章重点回顾

1、软件可靠性量化指标
2、软件可靠性设计
3、软件可靠性测试过程

THANKS