SOFTWARE TESTING

Chapter 10

Software Reliability Testing

School of Data & Computer Science
Sun Yat-sen University

Approaches & Technologies

中山大學
SUN YAT-SEN UNIVERSITY

OUTLINE

- 10.1 概述
  - 软件危机
  - 软件可靠性
  - 软件可靠性工程
- 10.2 软件可靠性分析与设计
  - 软件可靠性的影响因素分析
  - 软件可靠性的失效机理分析
  - 软件可靠性设计技术
- 10.3 软件可靠性评估
  - 软件可靠性评估的概念
  - 软件可靠性评估方法
  - 软件测试与软件可靠性评估
- 10.4 软件可靠性测试

软件危机

- 软件危机的出现
  - 软件销售量和使用量呈几何级数增长，软件规模不断增大，复杂性急剧提高。
    - 航天飞机的飞行软件达50万行源代码，F-22 战斗机更达150多万行源代码，软件失效已成为系统瘫痪的主要原因。
    - 根据美国国防部和 NASA 的统计，当今武器系统和航天项目中的软件可靠性比硬件系统大约低一个数量级。
- 因软件故障而造成的重大事故不乏其例
  - F-18 战斗机在海湾战争中，飞行控制软件共发生了500多次故障。
  - 爱国者导弹因软件问题误伤了28名美国士兵。
  - 阿里安5型火箭的发动机控制系统软件的错误导致飞行试验失败。

软件可靠性

- Software Reliability (SR): (A) The probability that software will not cause the failure of a system for a specified time under specified conditions. (B) The ability of a program to perform a required function under stated conditions for a stated period of time. (IEEE Std 1633)
- Software reliability is the probability that software will work properly in a specified environment and for a given amount of time. Using the following formula, the probability of failure is calculated by testing a sample of all available input states.
  - Probability = Number of failing cases / Total number of cases under consideration (wikipedia)
- 软件可靠性是在规定条件下和在规定时间内软件的运行不会导致系统失效的概率。
  - 该概率是系统输入和系统使用的函数，也是软件中存在的故障的函数。
  - 系统输入将确定是否会激发软件中存在的故障。

- 软件可靠性
  - 影响软件可靠性的因素
    - 软件规模
    - 软件对实际需求的表述上的符合度
    - 软件的运行剖面
    - 软件复杂度
    - 软件的开发方法
    - 软件开发人员的能力和经验
    - 软件开发的支持环境
    - 软件可靠性设计技术
    - 软件的测试
    - 软件的投放方式
    - 。。。

软件可靠性工程

- 软件可靠性工程 (SRE) 是指为了保证经济、及时地实现软件可靠性目标而采取的系统化的技术和管理活动及方法。
  - 在软件生存周期各阶段，为了保证达到较高的可靠性，需要在一般软件工程活动的基础上再增加或加强一些活动。
  - 软件可靠性工程是在软件工程基础上发展起来的，其技术内涵包括可靠性分析、设计、度量、管理四个方面。
  - 准确评价/预计软件可靠性是软件工程的重大问题之一。
  - 软件可靠性工程是一门得到普遍承认，但仍处于不成熟的、正在发展确立阶段的新兴工程学科。
    - 20世纪60年代后期：国外开始加强对软件可靠性的研究工作。
    - 20世纪90年代：各种可靠性模型和预测方法得到建立，形成较为系统的软件可靠性工程体系。

相关标准

- IEEE 1633-2008/2016 IEEE Recommended Practice on Software Reliability
- GB/T 14394-2008 计算机软件可靠性和可维护性管理
- GB/T 28171-2011 嵌入式软件可靠性测试方法
- GB/T 29832.1-2013 系统与软件可靠性 第1部分：指标体系
- GB/T 29832.2-2013 系统与软件可靠性 第2部分：度量方法
- GB/T 29832.3-2013 系统与软件可靠性 第3部分：测试方法

软件可靠性的影响因素分析

- 软件规模
  - 随着软件规模的增加，软件可靠性问题愈显突出。
    - 例如，航天项目作为一类庞大的系统工程，其控制软件具有相当的规模和复杂度。
  - 软件可靠性增长模型主要应用于综合测试阶段。
- 软件的运行环境 (剖面)
  - 软件故障是软件错误在一定输入条件下被激活的结果。
    - 如果在运行环境中从不包含激活软件缺陷的输入条件，软件的可靠性就恒为1。
    - 如果在运行环境中总包含激活软件缺陷的输入条件，软件的可靠性就恒为0。

软件可靠性的影响因素分析

- 软件内部结构
  - 软件内部结构越复杂，软件复杂度越高，内含的缺陷数目就可能越大，因而软件可靠度也就可能越低。
    - 例如，航天软件内部结构一般比较复杂，而且存在一定的动态变化，对可靠性的影响因素分析带来困难。
- 软件可靠性设计技术
  - 软件可靠性设计技术是指软件设计阶段中采用的以保证和提高软件可靠性为主要目标的软件技术。
  - 目前对软件可靠性设计概念的外延尚未明确界定，可以借助硬件产品的可靠性设计技术，如失效模式与影响分析 (FMECA)、故障树分析 (FTA) 等，进行软件可靠性设计。
  - 在软件设计中是否采用软件可靠性设计技术，或者采用什么样的可靠性设计技术，对软件的可靠性具有很大影响。

软件可靠性的失效机理分析

- 软件失效机理的描述方式
  - 软件错误
    - 软件错误是在软件生存期内的各种人为错误，其结果导致软件缺陷的产生。
      - 在整个软件生存期中，都贯穿着人的干预；人作为一种客观实体，在软件设计中难免会犯错误。
      - 软件错误是一种人为过程，是相对于软件本身的外部错误。
    - 一个软件错误必定产生一个或多个软件缺陷。

软件可靠性的失效机理分析

软件失效机理的描述方式 (续)

软件缺陷

- 软件缺陷是存在于软件 (文档、数据、程序) 之中的各种偏差。当软件运行于某一特定条件时，软件缺陷被激发。
  - 软件缺陷是存在于软件内部的一种静态形式。
- 一个软件缺陷被激发时，便产生一个软件故障。
- 同一个软件缺陷在不同的条件下被激发，可能导致不同的软件故障。

软件可靠性的失效机理分析

软件失效机理的描述方式 (续)

软件故障

- 软件故障是软件运行过程中出现的一种不希望或不可接受的内部状态。
  - 软件故障在软件运行过程中出现，是一种动态行为。
- 软件故障出现时如果没有适当的处理措施，便会产生软件失效。
- 同一个软件故障在不同条件下可能导致不同的软件失效。

软件失效

- 软件失效是软件运行时产生的一种不希望或不可接受的外部行为结果。
  - 例如，软件运行中由于软件故障引发的死机现象是一种常见的严重软件失效结果。

软件可靠性的失效机理分析

- 软件错误分类
  - 软件设计错误
    - 软件设计错误由软件设计过程中对任务书和软件需求理解出现偏差和错误引起，与设计者的能力和经验有关。
    - 软件设计错误包括软件文档错误、模型和算法错误及其它人为错误。
  - 软件编程错误
    - 软件编程错误由编程者的编程错误或者对设计文档理解错误引起，与编程者的能力和经验有关。
    - 软件编程错误包括程序语法错误、语义错误、软件实现逻辑错误；此外还有软件源程序与设计文档不符引发的错误以及其它人为错误。

软件可靠性的失效机理分析

- 软件错误分类 (续)
  - 软件外部输入错误
    - 该错误出现在软件运行过程中。由于软件对外部输入条件都有相应的要求，如果输入数据不满足条件，加上对输入数据的检验和越界保护机制不完备，可能会引发数值范围越界、数组指针越界、除以零等错误。
  - 软件异常处理错误
    - 软件异常处理机制不完备导致软件出现异常时不能及时得到处理而引起软件失效。
      - 软件异常处理在整个软件完成的工作中占有很高比例(60%以上)。

软件可靠性设计技术

- 软件可靠性设计技术指那些适用于软件设计阶段，以保证和提高软件可靠性为主要目标的设计技术。
- 软件可靠性设计技术包括
  - (1) 软件在线自检和动态分析技术
  - (2) 软件容错技术
  - (3) 软件失效模式、影响及危害性分析技术 (FMECA)
  - (4) 软件故障树分析技术 (FTA)
  - (5) 软件复杂性控制技术
  - (6) 软件可靠性预计和估计技术
  - (7) 软件可靠性建模技术

- 软件在线自检和动态分析技术
  - 用于对正在执行的程序性能进行实时检查分析；
  - 软件动态分析可以检验程序的测试覆盖率和所有功能。
- 软件容错技术
  - 容错技术是软件设计中常用的方法之一。
  - 在软件设计中对于各种输入数据取值范围的越界检查和保护、对不满足函数定义域的数据处理以及对各种异常的处理设计都可以提高软件的可靠性。

软件失效模式、影响及危害性分析

- 软件失效模式、影响及危害性分析 (FMECA, Failure Mode Effects and Criticality Analysis) 是检查软件设计和代码在出现意外的异常输入或意外的异常环境条件时的特性的一种分析技术。
  - FMECA 用于识别和修正软件中由于设计缺陷造成的系统故障；
  - 合理利用软件 FMECA 方法可在任何等级提供软件错误信息，以保证在需求定义阶段避免软件故障、在测试阶段确定软件错误类型和改正措施。
  - 软件故障确定后，可以利用容错、自检验技术或制定软件保护措施，以减少程序执行时出现故障的概率。
- 软件 FMECA 可利用多种技术，其中危险分析法和可行性研究是软件需求定义阶段所采用的两种技术。
  - 前者用于确定故障模式及其后果；
  - 后者用于决定如何消除这些模式的故障。

- 软件故障树分析
  - 故障树分析 (FTA, Fault Tree Analysis) 是一种自上而下的分析技术。
    - 将不希望发生的事件作为顶级事件，然后由熟悉系统及关键事件的人员进行实施或者管理，以识别和消除引起这些不希望事件的状态。
  - 软件 FTA 的最主要用途是把顶级关键事件转变为在分系统级必须防止的事件，然后利用 FMECA 确定有害事件的潜在原因。
    - 通常需要建立新的配置或者进行程序重新设计和完善，以保证这些有害事件发生的概率低于一个可接受的门限。
- 软件复杂性控制技术
  - 在软件设计中，采用模块化、层次化设计等方法，均可以降低软件的复杂度，从而提高软件的可靠性。

软件可靠性预计和估计技术

- 软件可靠性预计和估计是软件开发人员和用户检查所开发的软件是否满足要求的常用工具。
  - 软件可靠性预计在开发阶段的早期 (在不存在可执行代码前) 利用从以前类似的软件中所获得的数据进行；
  - 软件可靠性估计在测试和使用阶段进行.
- 软件可靠性预计和估计一般可通过两种途径进行：
  - 数学模型法：利用以前已有的较为成熟的软件可靠性模型进行预计和估计。
  - 经验公式法：利用所开发的软件的一些基本特性、软件开发环境及应用类型来预计或估计软件可靠性。

软件可靠性建模技术

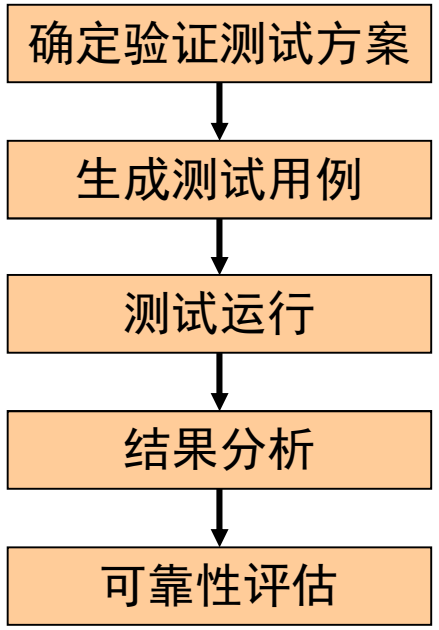
- 软件可靠性建模技术包括软件可靠性模型、软件交付时间模型、软件可用度模型等，其中软件可靠性模型最多，而且用途最为广泛。
- 软件可靠性模型通常分为两大类：分析模型、经验模型。

软件可靠性评估的概念

- 软件可靠性评估采集在软件测试和软件运行期间得到的失效数据，应用统计推理方法，确定模型参数并评估系统当前的软件可靠性指标。
  - 根据软件系统可靠性结构 (单元与系统间可靠性关系)、寿命类型和各单元的可靠性试验信息，选择合适的概率统计方法，对模型参数进行估计，在此基础上得到软件可靠性指标的各种统计推断。
  - 目前，软件可靠性评估主要是以一些常用的软件可靠性模型为基础进行可靠性指标评估。

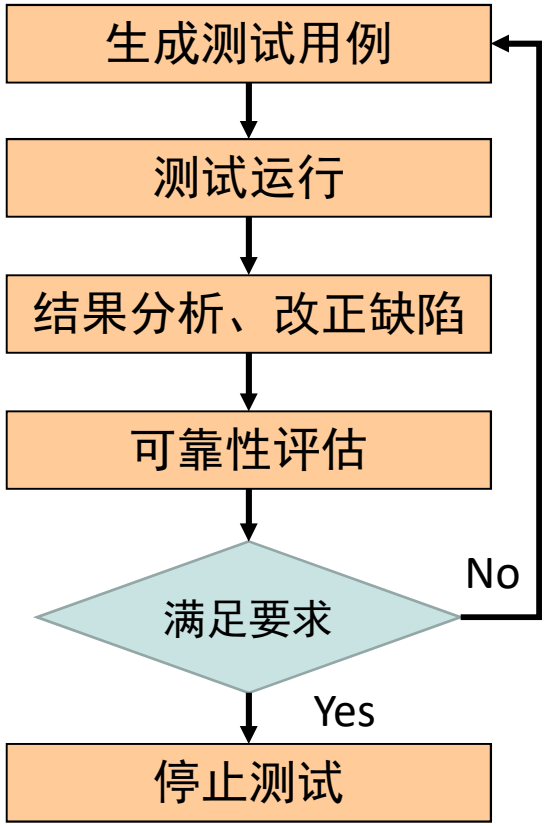
软件可靠性评估方法

- 软件可靠性评估方法包括
  - 基于软件可靠性测试的验证方法
  - 基于软件可靠性建模的方法
- 基于软件可靠性测试的验证方法
  - 在给定的统计置信度下，验证软件当前的可靠性水平是否满足用户的要求 (即软件规格说明书中规定的可靠性指标是否得到满足)。
    - 一般在软件验收阶段进行，在软件需求方参与的情况下实施。
  - 主要过程：根据现场测试的故障情况，利用双方都认可的某种可靠性验收模型或软件工具进行可靠性的定量评价，以判断该软件是否达到需求说明书中所约定的可靠程度。



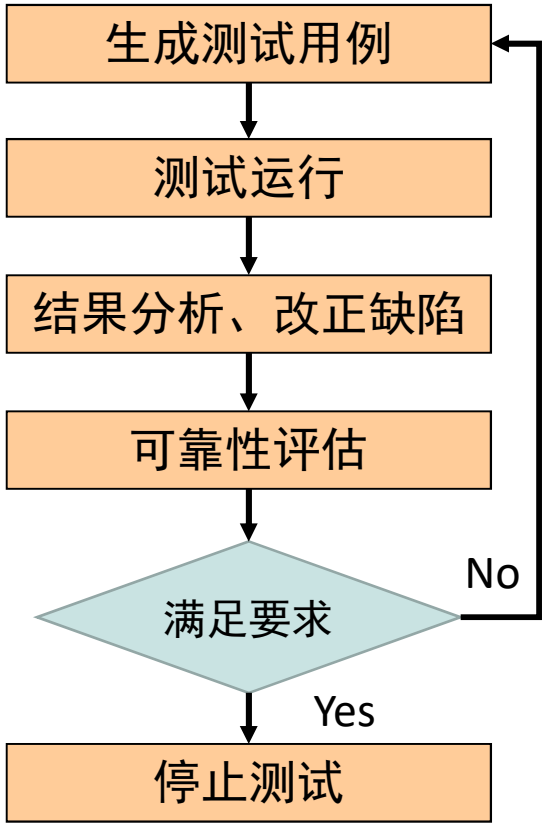
软件可靠性评估方法

- 基于软件可靠性建模的方法
  - 基于建模的方法在测试时，将所确定的失效交由开发者分析和修改，建立软件的一个新的版本，再进行下一次测试。在“测试—排错—建立新版本”的迭代过程中，被发现的软件错误不断被剔除，软件可靠性呈增长趋势，故又称为软件可靠性增长建模，它是当前软件可靠性建模的主要内容。
  - 基于建模的方法主要用于软件的开发阶段，测试与缺陷的排除联系在一起，一般在开发过程的系统测试阶段进行。



软件可靠性评估方法

- 基于软件可靠性建模的方法 (续)
  - 基于建模的方法与“验证法”比较
    - 共同点：故障情况依赖于测试；
    - 区别：在进行测试的同时修改故障，并通过对所收集的故障行为进行建模分析，估计软件可靠性的实际水平。

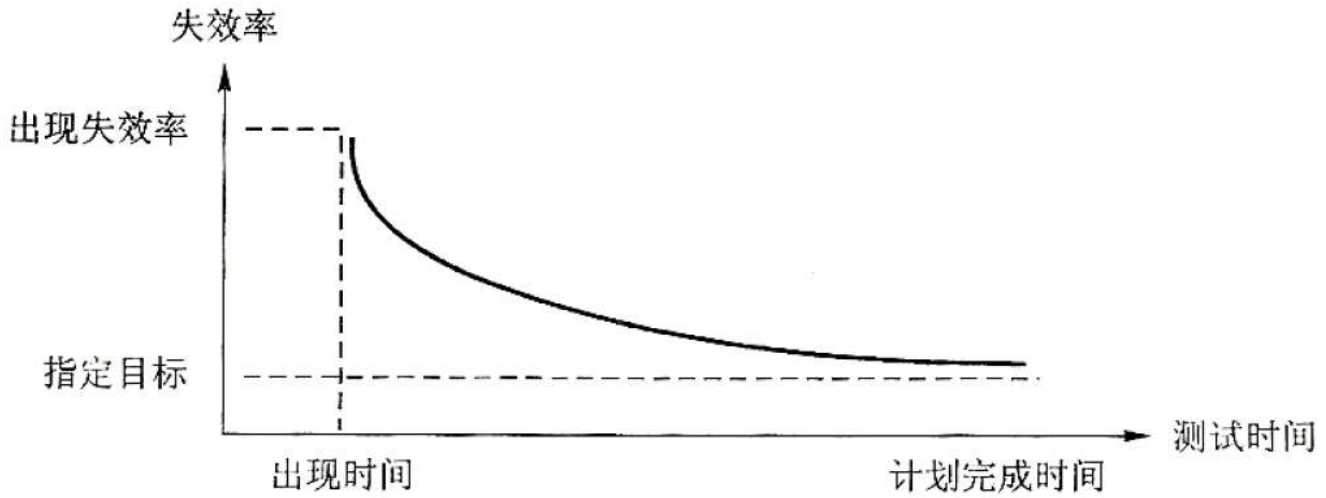


软件可靠性评估方法

- 软件可靠性建模
  - 软件可靠性模型是软件可靠性评估中的一个重要概念，是软件可靠性定量分析技术的基础。
    - 目的：根据与软件可靠性有关的数据，以统计的方法给出软件可靠性的估计值或预测值。
  - 软件可靠性工程中使用的2种模型：
    - 软件可靠性结构模型
      - 类似硬件可靠性模型
    - 软件可靠性预计模型

软件可靠性评估方法

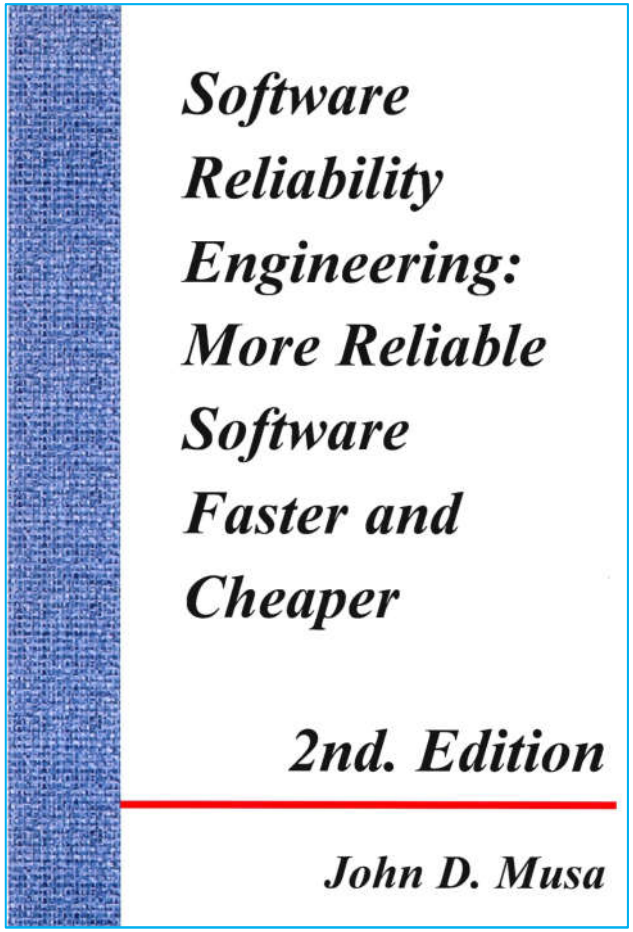
- 软件可靠性建模
  - 软件可靠性预计模型
    - 软件的失效率总体上随时间的增加而减小；
    - 在任意给定时间里，可以观察到软件失效率的历史；
    - 通过对结果的统计可以预测失效率曲线，从而估计达到规定目标还需要多少测试时间，以及测试结束时软件可靠性的期望值。



软件可靠性评估方法

- 典型的软件可靠性建模
  - 一个有效的软件可靠性模型应尽可能地将影响软件可靠性的因素在软件可靠性建模时加以考虑，尽可能简明地反映出来。
  - *Musa* 和 *Okumoto* 提出的模型分类方案：将软件按照以下特征进行分类：
    (1) Time domain：时钟时间与执行时间的比较；
    (2) Category：在确定时间内的失效总数，分为有限和无限2个子类；
    (3) Type：时间的失效数分布，包括泊松分布和二项式分布；
    (4) Class：用期望失效数目表示失效强度函数的函数形式，仅对有限失效类有效；
    (5) Family：用期望失效数目表示失效强度函数的函数形式，仅对无限失效类有效。

- 软件可靠性评估方法
  - 典型的软件可靠性建模
    - *John D. Musa*, Software Reliability Engineering, 2004



软件可靠性评估方法

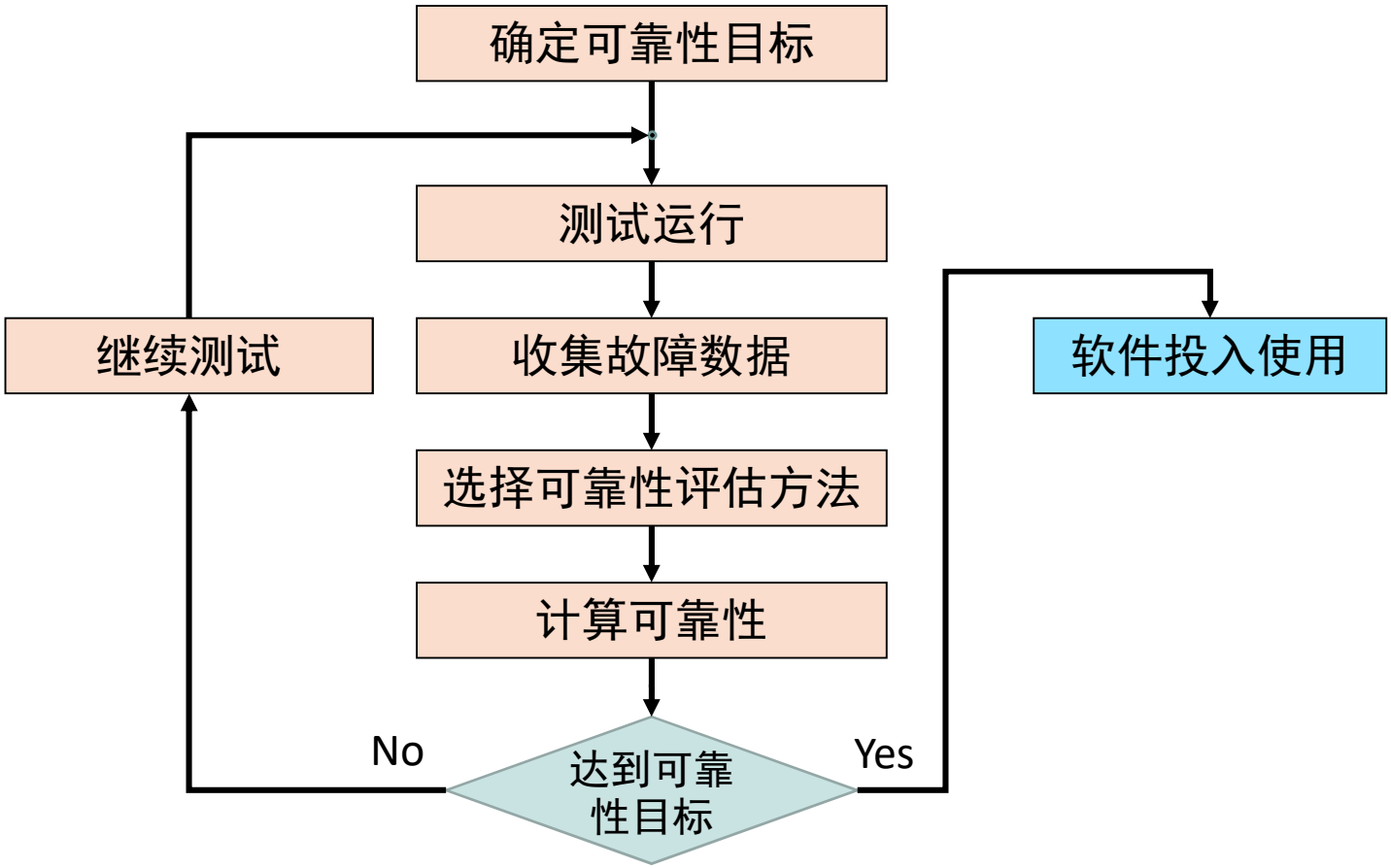
- 典型的软件可靠性建模 (续)
  - 根据以上特征分类，常见的几种可靠性模型有：
    - 指数类失效时间模型
    - *Weibull Gamma* 失效时间模型
    - 无限失效类模型
    - 贝叶斯模型

软件可靠性评估方法

- 软件可靠性数据
  - 软件可靠性数据是可靠性评估的基础
  - 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。
  - 时间定义的软件可靠性数据可以分为四类，并可互相转化：
    - 失效时间数据：发生一次失效所累积经历的时间
    - 失效间隔时间数据：本次失效与上一次失效的间隔时间
    - 分组时间数据：某个时间区间内发生了多少次失效
    - 分组时间内的累积失效数：某个时间区间内的累积失效数

- 软件测试与软件可靠性评估
  - 软件测试是软件可靠性评估的基础，也是提高软件可靠性的主要手段。
    - 软件可靠性模型利用软件测试所提供的故障数据，估算软件的可靠性，对软件将来的故障行为进行预测，以协助开发人员监督软件开发过程，辅助软件过程管理。
  - 软件可靠性评估反过来为软件测试服务。
    - 软件可靠性评估的结果是衡量软件测试工作是否充分的判断依据。

软件测试与软件可靠性评估



软件可靠性测试的概念

- 什么是软件可靠性测试
  - 软件可靠性测试是指为了达到或验证用户对软件可靠性的要求而对软件进行的测试，是一种有效的软件测试和软件可靠性评价技术。
    - 软件可靠性测试不能保证软件中残存的错误数达到最少，其目的是保证软件的可靠性达到较高的要求。
    - 软件可靠性度量的准确性是软件可靠性测试的重要问题。
      - 测试环境应该尽量反映软件真实的运行环境 (全面地或者局部性地)，才有可能通过可靠性测试对软件在实际运行中的可靠性进行准确评价。

软件可靠性测试的概念

- 软件可靠性测试的目的
  - 在具备使用代表性的环境中 (在统计意义下该环境能够反映出软件的使用环境特性) 执行目标软件，以证实该软件的可靠性需求是否得到正确实现。
    - 采集准确的数据以便于进行软件可靠性评价。
      - 软件可靠性评价一般可分为四个步骤进行：数据采集、模型选择、模型拟合以及软件可靠性评估。
      - 数据采集是整个软件可靠性评价工作的基础，数据的准确性关系到软件可靠性评价的准确程度。
    - 发现所有对软件可靠性影响大的错误，验证软件可靠性满足一定的要求。
    - 估计、预计软件可靠性水平。

- 软件可靠性测试的概念
  - 软件可靠性测试的特点
    - 软件可靠性测试从硬件可靠性测试发展而来，但二者失效的原因不同。
      - 硬件失效一般是由于元器件老化引起的，因此硬件可靠性测试强调随机选取多个相同的产品，统计它们的正常运行时间。
        - 正常运行的平均时间越长，则硬件就越可靠。
      - 软件失效一般是由设计缺陷造成的，软件的输入决定了是否会激发软件内部存在的故障。
        - 使用同样一组输入反复测试软件并记录其失效数据是没有意义的；
        - 在软件没有改动的情况下，这种数据只是首次记录的不断重复，不能用来估计软件可靠性。

软件可靠性测试的概念

软件可靠性测试的特点 (续)

- 软件可靠性测试按照软件的测试模型 (对软件实际使用情况的统计规律的描述) 对软件进行测试。
  - 软件可靠性测试是面向失效的测试，以用户将要使用的方式来测试软件，每一次测试代表用户将要完成的一组操作，使测试成为最终产品使用的预演。
  - 软件可靠性测试在具备使用代表性的环境或在软件的预期使用环境中，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、评价等一系列手段对软件系统实施特定的功能性测试。
    - 环境的使用代表性指的是在统计意义下该环境能反映出软件的使用环境特性。

软件可靠性测试的概念

软件可靠性测试的特点 (续)

- 软件可靠性测试不同于一般的软件功能测试。
  - 强调测试输入与典型使用环境输入统计特性的一致，强调对功能、输入、数据域及其相关概率的先期识别。
  - 强调按实际使用的概率分布随机地选择测试实例，才能得到比较准确的可靠性估计，也有利于找出对软件可靠性影响较大的故障。
  - 准确记录软件的运行时间，并强调测试需求的覆盖面 (其输入覆盖一般也要大于普通软件功能测试的要求)。
  - 对于一些特殊的软件，如容错软件、实时嵌入式软件等，进行软件可靠性测试时需要有多种测试环境。这是因为在使用环境下常常很难在软件中植入错误以进行针对性的测试 。

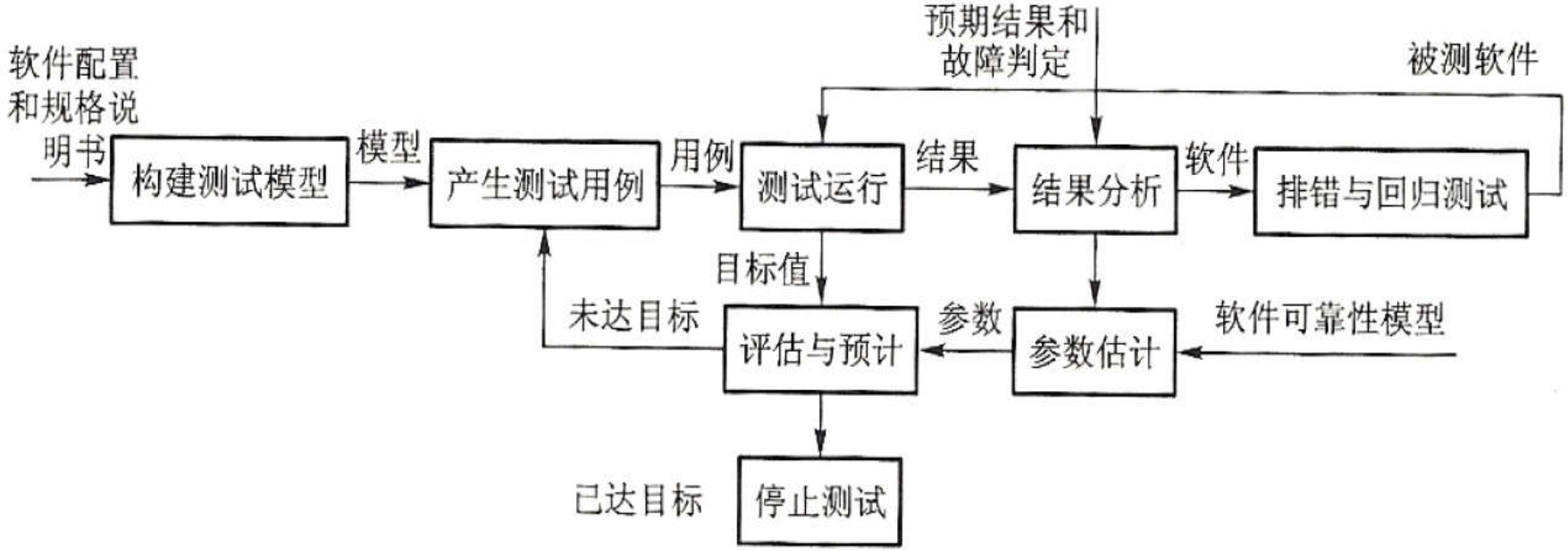
软件可靠性测试的概念

- 软件可靠性测试的意义
  - 软件可靠性测试是获取软件可靠性估算数据的重要手段。
  - 通过发现软件系统可靠性缺陷，为软件的使用和维护提供数据，确认软件是否达到可靠性的定量要求。
  - 目前软件可靠性测试用得最多的方法是类似于硬件操作剖面上的统计测试方法，即基于被测软件操作剖面的统计测试方法。
    - 这是最为传统、经典的软件可靠性测试方法；
    - 通过这种方法，能够实现软件可靠性的定量评估，从而有效地保障和提高软件的质量。

软件可靠性测试流程

- 软件可靠性测试流程由三个步骤构成
  - 构造测试模型
    - 按用户实际使用软件的方式，模拟软件的真实运行环境。
  - 产生测试用例并执行测试
    - 根据测试模型，随机产生测试用例，然后在真实的测试环境或仿真测试环境中，对系统进行测试。
  - 可靠性评价
    - 根据所收集的失效数据，选择适当的软件可靠性模型对测试结果进行分析，评价和预计软件的可靠性，以确定可靠性测试是否停止。

软件可靠性测试流程


软件配置
和规格说
明书
构建测试模型
模型
产生测试用例
用例
测试运行
预期结果和
故障判定
结果
结果分析
软件
排错与回归测试
被测软件
目标值
未达目标
评估与预计
参数
参数估计
软件可靠性模型
已达目标
停止测试

- 软件可靠性测试的基本方法
  - 基于运行剖面的可靠性测试方法
    - 1993年，由 *J. D. Musa* 提出。
    - 基本思想
      - 用户使用软件的方式对于软件可靠性测试和可靠性评价有十分重要的影响。利用软件运行剖面定量描述用户实际使用软件的方式，据此得到比较准确的软件可靠性指标。
    - 此方法多用于理论研究。

软件可靠性测试的基本方法

- 基于使用模型的统计测试方法
  - 由 *Harlan D. Mills* 和 *James A. Whittaker* 提出。
  - 基本思想
    - 在软件的实际使用过程中，软件的使用方式会表现出一定的统计特性，基于使用模型的统计测试就是建立软件的使用模型，根据使用模型产生测试用例，并对软件的可靠性进行度量。
  - 使用模型：软件使用中所有可能的情形及其发生的概率。
    - 使用模型将软件的使用方式以模型的方法表示出来，描述了软件的使用特性。
  - 基于马尔可夫链的使用模型在净室软件工程 (Cleanroom Software Engineering) 中得到了比较成功的应用，国外已经有一些用于构造使用模型的辅助工具。

软件可靠性测试的基本方法

- 基于操作剖面的统计测试方法
  - 操作剖面指系统测试数据输入域以及各种输入数据的组合使用概率；操作剖面描述了系统的实际使用情况。
    - 软件的操作剖面是“软件对使用条件的定义。即软件的输入值用其按时间的分布或按它们在可能输入范围内的出现概率的分布来定义”。
    - 操作剖面是否能代表、刻画软件的实际使用取决于可靠性工程人员对软件的系统模式、功能、任务需求及相应的输入激励的分析，取决于他们对用户使用这些系统模式、功能、任务的概率的了解；操作剖面构造的质量将对测试、分析的结果是否可信产生最直接的影响。
    - 操作剖面的建立可以与软件开发并行进行，缩短软件开发和发布所需的时间。

软件可靠性测试的基本方法

- 基于操作剖面的统计测试方法 (续)
  - 统计测试是通过对输入统计分布进行分析来构造测试用例的一种测试设计方法。
    - 统计测试标识出频繁执行的部分，并相应地调整测试策略，针对这些频繁执行的部分进行详尽的测试。
    - 统计测试通过提高关键模块的安全性和可靠性，来提高整个系统的安全性和可靠性，以提高测试的性价比。
    - 统计测试进行的前提条件是生成如实反映系统使用情况的使用模型。

软件可靠性测试的基本方法

- 基于操作剖面的统计测试方法 (续)
  - 基于操作剖面的统计测试方法
    - 基于操作剖面的统计测试方法在对软件的实际使用情况进行统计的基础上建立软件的操作剖面，并采用统计测试的方法进行测试。
      - 用这种方法所获得的测试数据与软件的实际运行数据比较接近，可直接用于软件可靠性估算。
    - 利用使用模型在软件所有可能操作中进行统计上正确的采样，以样本运行结果为基础，得出软件操作可靠性结论。
    - 实验结果说明，随机测试优于输入域划分测试和结构覆盖测试，随机测试具有更好的效费比，并使测试人员能够对软件可靠性作出可靠的估算。但随机测试可能无法完全测试软件处理异常情况的能力。

软件可靠性测试的效果

- 软件可靠性测试是软件可靠性保证过程的关键步骤。
  - 经过软件可靠性测试的软件并不能保证该软件中残存的错误数最小，但可以保证该软件的可靠性达到较高的要求。
  - 从工程的角度来看，一个高可靠性的软件不仅意味着该软件的失效率较低，而且意味着一旦该软件失效，由此所造成的危害也较小。
  - 一个大型的工程软件没有错误是不可能的，至少理论上还不能证明一个大型的工程软件能没有错误。
    - 保证软件可靠性的关键不是确保软件没有错误，而是要确保软件的关键部分没有错误。
      - 更确切地说，是要确保软件中没有对可靠性影响较大的错误。这正是软件可靠性测试的目的之一。

软件可靠性测试的效果

- 软件可靠性测试实例设计的出发点是寻找对可靠性影响较大的软件故障。
  - 要达到同样的可靠性要求，可靠性测试比一般的功能测试更有效，所花的时间也更少。
  - 软件可靠性测试的环境是具有使用代表性的环境。
    - 保证所获得的测试数据与软件的实际运行数据比较接近，可用于软件可靠性估计。
  - 总之，软件可靠性测试比一般的功能测试更加经济和有效，它可以代替一般的功能测试，而一般的软件功能测试却不能代替软件可靠性测试，而且一般功能测试所得到的测试数据也不宜用于软件可靠性估计。

软件可靠性测试的阶段划分

- 软件可靠性测试一般可分为四个阶段
  - 制定测试方案
    - 制定测试方案时需要特别注意被测功能的识别和失效等级的定义。
  - 制定测试计划
    - 制定测试计划时需要设计测试实例。
  - 进行测试并记录测试结果
    - 进行测试时要确定输入顺序，并确定程序输出的预期结果，这时也需注意测试覆盖问题。
  - 编写测试报告

软件可靠性测试的功能识别

- 软件可靠性测试过程首先是进行功能识别。
  - 功能识别的目的是确定使用剖面。
- 功能识别的两大目标：

(1) 识别所有被测功能以及执行这些功能所需的相关输入。

- 分析软件的功能集合、功能之间的约束条件、功能之间的独立性、相互关系和相互影响。
- 分析系统的不同运行模式、失效发生时系统的重构策略等对软件运行方式有较大影响的因素。

软件可靠性测试的功能识别

- 功能识别的两大目标：(续)

(2) 识别每一个使用需求及其相关输入的概率分布。

- 为了得到能够反映软件使用需求的有代表性的概率分布，测试人员必须和系统工程师、系统运行分析员和需求方共同合作。
- 由于可靠性的要求，输入数据的概率分布应包括合法数据的概率分布和非法数据的概率分布两部分。
- 有时为了更好地反映实际使用状况，还需给出那些影响程序运行方式的条件，如硬件配置、负荷等的概率分布。

软件可靠性测试的失效等级

- 失效等级的引入
  - 定义失效等级主要是为了解决下面两个问题：
    - 对发生概率小但失效后危害严重的功能需求的识别；
    - 对可不查找失效原因、并不做统计的功能需求的识别。
  - 在制定测试计划时，失效及其等级的定义应由测试人员、设计人员和需求用户共同协定。

软件可靠性测试的失效等级

- 失效等级的使用原则
  - 按照一般的等级定义，如果存在1级和2级失效可能性，那么就应该进行故障树分析，标识出所有可能造成严重失效的功能需求和其相关的输入域、外部条件和发生的可能性。
  - 对引起1级和2级失效的功能需求及其相关的输入域必须进行严格的强化测试。
  - 对引起3级失效的功能可按其发生概率选择测试实例。
  - 第4级失效可不查找原因，也可在以后的版本中处理。

软件可靠性测试的覆盖

- 软件可靠性测试必须保证输入覆盖和环境覆盖
  - 输入覆盖和环境覆盖是准确估计软件可靠性的基础。
- 输入覆盖包括：
  - 输入域覆盖，即所有被测输入值域的发生概率之和必须大于软件可靠度的要求；
  - 重要输入变量值的覆盖；
  - 相关输入变量可能组合的覆盖，以确保相关输入变量的相互影响不会导致软件失效；
  - 设计输入空间与实际输入空间之间区域的覆盖，即不合法输入域的覆盖；
  - 各种使用功能的覆盖。
- 环境覆盖
  - 测试时必须覆盖所有可能影响程序运行方式的环境条件。

软件可靠性测试的具体步骤

1. 制定测试方案

- 目标是识别软件功能需求，触发该功能的输入和对应的数据域，确定相关的概率分布及需强化测试的功能。

(1) 分析功能需求

- 分析各种功能需求，识别触发该功能的输入及相关的数据域 (包括合法与不合法的两部分)。
- 分析时要注意的问题
  - 列出所有的系统运行模式；
  - 列出所有影响程序运行方式的外部条件，评价它们的影响程度；
  - 分析各种功能需求之间的相关程度以及关联因素。

软件可靠性测试的具体步骤

1. 制定测试方案 (续)

(2) 定义失效等级

- 判断是否存在出现危害度较大的1级和2级失效的可能性。
- 如果这种可能性存在，则应进行故障树分析，标识出所有可能造成严重失效的功能需求和其相关的输入域。

(3) 确定概率分布

- 确定各种不同运行模式下失效的发生概率，判断是否需要对不同的运行模式进行分别测试。
- 如果需要应给出各种运行模式下各数据域的概率分布，否则给出各数据域的概率分布。
- 判断是否需要强化测试某些功能。

(4) 整理概率分布的信息

- 将这些信息编码送入数据库。

软件可靠性测试的具体步骤

2. 制定测试计划

(1) 根据前一阶段整理的概率分布信息生成相对应的测试实例集，并计算出每一测试实例预期的软件输出结果。

- 在按概率分布随机选择生成测试实例的同时，要保证测试的覆盖面。

(2) 编写测试计划，确定测试顺序，分配测试资源。

- 本阶段前一部分的工作需要处理大量的信息和数据，因此需要软件工具的支持，建立数据库，并产生测试实例。
- 另外，有时预测软件输出结果也需要大量的计算，有些复杂的软件甚至需要仿真器模拟输出结果。

软件可靠性测试的具体步骤

3. 测试与测试报告编写

- 被测软件的测试环境 (包括硬件配置和软件支撑环境) 应和预期的实际使用环境尽可能一致，对某些环境要求比较严格的软件 (如嵌入式软件) 则应完全一致。
- 测试时按测试计划和顺序对每一个测试实例进行测试，判断软件输出是否符合预期结果。
- 测试时应记录测试结果、运行时间和判断结果。如果软件失效，那么还应记录失效现象和时间，以备以后核对。
- 按软件可靠性估计的要求整理测试记录，并将结果写成报告。

Thank you!