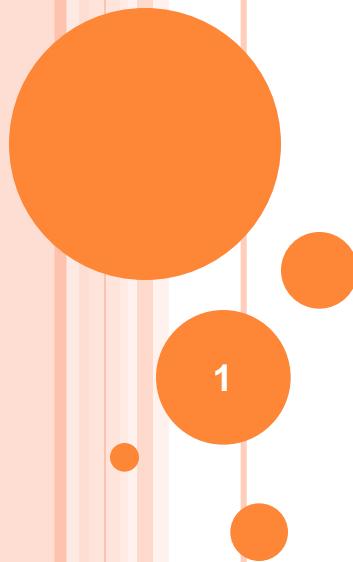




软件可靠性和安全性设计分析

曾福萍



1



课程安排

周次	内容
1	基础知识
2	软件FMEA
3	软件FTA
4	其他软件可靠性安全性分析技术
5	软件可靠性安全性设计方法及设计方准则
6	软件故障检查、故障处理及信息时间容错
7	软件结构容错及设计准则的实施
8	讨论

软件可靠性安全性设计分析

2



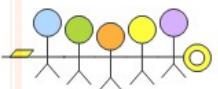
授课目的

- 从5W2H的角度充分理解软件可靠性安全性设计分析的内涵

Why, What, When, Who, Where, How, Howmuch

- 提高分析问题解决问题的能力

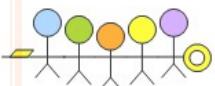
所涉及的一些解决问题的方法能推广到其它方面，从而提高分析问题解决问题的能力





参考资料

1. 《软件可靠性、安全性与质量保证》 黄锡滋
编著 电子工业出版社 2002年10月
2. 《软件可靠性工程手册》 Michael R.LYU主编，
电子工业出版社 1997年3月
3. 软件安全性相关标准
4. 软件可靠性安全性设计分析方面的论文及期刊等
5. 软件工程方面的书籍，如《软件工程》 张海藩
编著 人民邮电出版社 2003年7月
6. 软件容错方面的书籍及期刊、论文等





考核要求

- 总成绩
= 平时 (40%) + 期末 (60%)
 - 平时：课堂讨论、提问及考勤
 - 期末：讨论和论文



最终提交论文要求

○ 共同内容

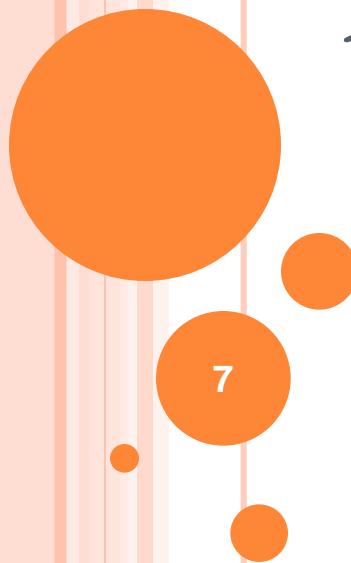
- 从5W2H的角度阐述软件可靠性安全性设计分析的内涵。

○ 特定内容（从下列内容中选一）

- 1、模型驱动方法在软件可靠性设计中的应用，含模型驱动方法的概念、开发思想、范例等内容。
- 2、形式化方法在软件可靠性设计中的应用，含形式化方法的概念、开发思想、范例等内容。
- 3、汇编语言的软件可靠性安全性设计准则，含准则内容、说明及示例等内容。
- 4、SFMEA的应用案例，含案例系统概述、应用流程及应用结果等内容。
- 5、SFTA的应用案例，含案例系统概述、应用流程及应用结果等内容。



第一讲：基础知识





基础知识-主要内容

软件可靠性安全性
设计分析

- 一、软件**可靠性**安全性概念及关系
- 二、软件可靠性安全性**分析**概念及关系
- 三、软件可靠性安全性**设计**概念及关系

软件可靠性安全性设计分析



一、软件可靠性安全性概念及关系





基础知识1-软件可靠性概念

GB/T 11457 软件可靠性两种定义：

- ① 在规定条件下在规定的时间内软件不引起系统失效的概率，该概率是系统输入和系统使用的函数，也是软件中存在的缺陷的函数。系统输入将确定是否会遇到已存在的缺陷（如果有缺陷存在的话）。

- ② 在规定的时间周期内所述条件下程序执行所要求的功能的能力

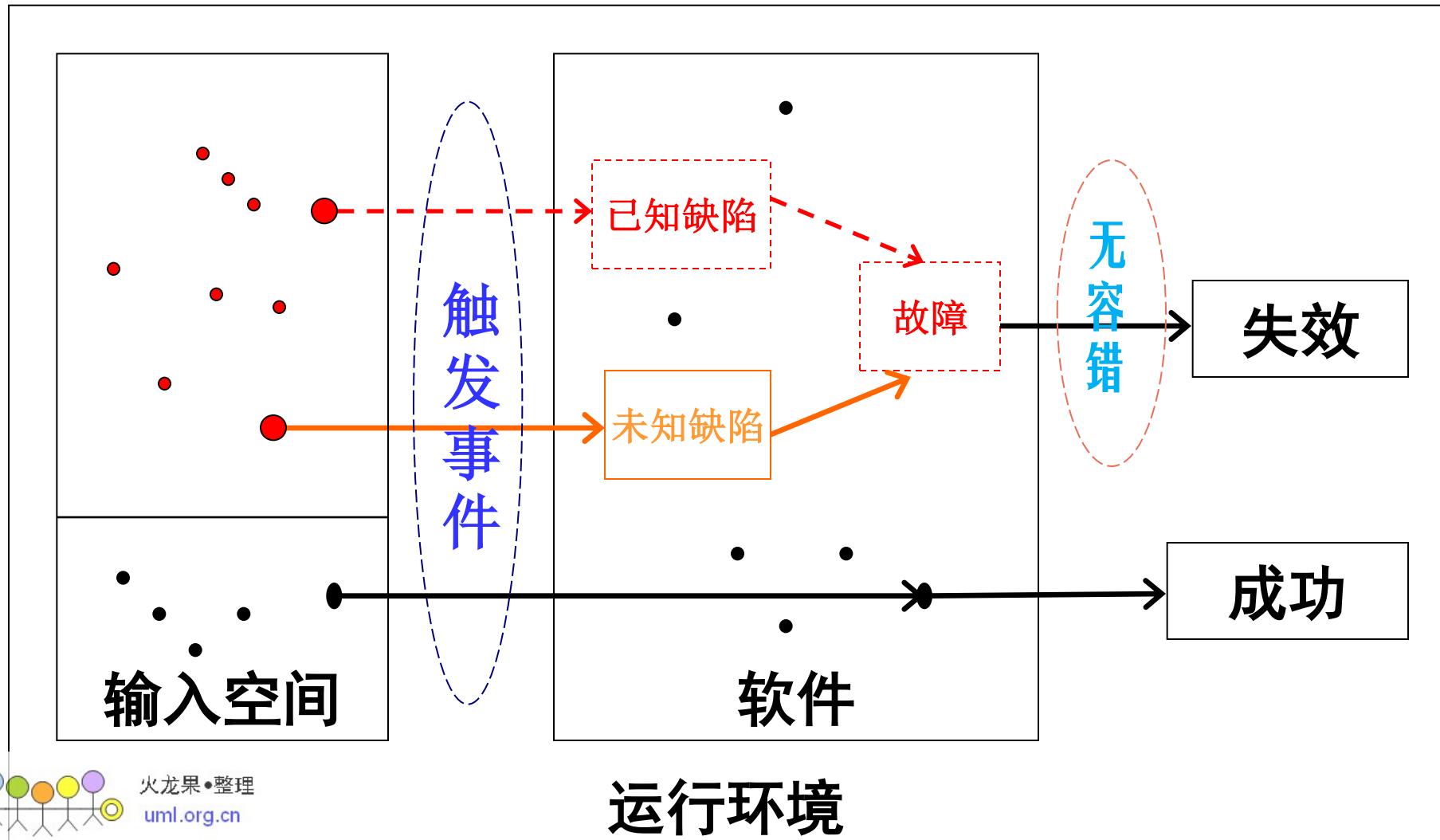
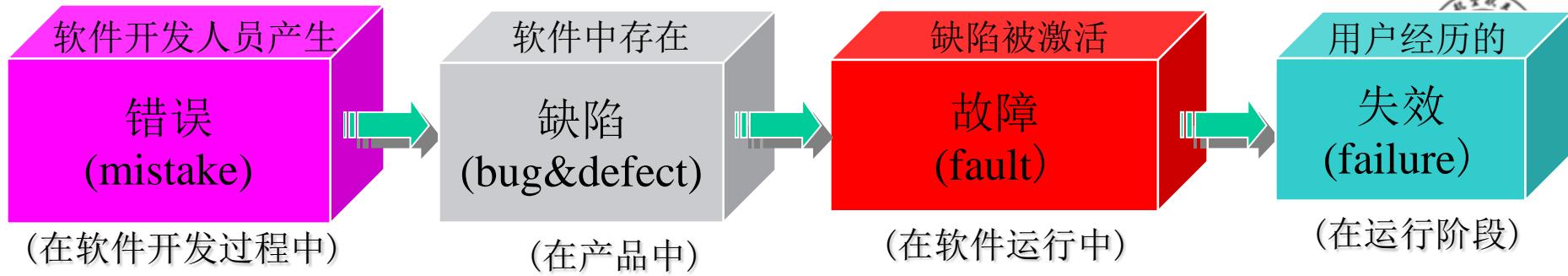


基础知识1-软件可靠性涉及的几个概念

- 错误、缺陷、故障、失效

- ◆ IEEE软件工程术语
- ◆ GB/T11457-95
- ◆ SW-CMM
- ◆ IEEE的软件可信性度量的标准

- 错误(Wrong): 在软件开发过程中出现的不符合期望或不可接受的人为差错。
- 缺陷(Defect): 存在于软件中的、不期望的或不可接受的偏差。
- 故障(fault/failure): 产品不能执行规定功能的状态。
- 失效(failure): 产品丧失完成规定功能的能力的事件。





软件可靠性相关概念

- 理想条件:

- 基本可靠性 (**basic reliability**)

产品在规定的条件下、规定的时间内无故障工作的能力。

- 固有可靠性 (**inherent reliability**)

通过设计和制造赋予产品的，并在理想的使用和保障条件下所呈现的可靠性。

- 实际使用条件:

- 使用可靠性 (**operational reliability**)

产品在实际使用条件下所表现出来的可靠性。它反映了产品设计、制造、安装、使用、维护、环境等因素的影响。一般用可靠性使用参数及其量值描述。





基础知识1-构成软件可靠性的要素

○ 1、规定的功能

- 软件执行的功能和性能要求
- 软件失效的定义

○ 2、规定的条件

- 软件运行的物理环境
- 软件输入及其分布，运行剖面

同一个软件，不同的用户、不同的使用方式与输入，不一样的软件可靠性





基础知识1-构成软件可靠性的要素

○ 3、规定的时间

日历时间——编年时间

时钟时间——从程序执行开始到程序执行结束完毕所经过的时钟时间

执行时间——处理机实际用于执行程序指令的时间

例：一个软件，5周内运行70小时，其中55小时为该软件的执行时间，则：

日历时间为 5周；

时钟时间为70小时；

执行时间为55小时。





构成软件使用可靠性的要素

- 软件执行的功能和性能要求
- 软件的运行环境
- 软件失效的定义
- 软件输入的分布——运行剖面



基础知识2-软件安全性概念

序号	来源	定义描述
1	学者Nancy G.Leveson	软件安全性涉及确保软件在系统环境中运行而 不产生不可接受的风险 。
2	2004年美国航天航空局的软件安全性标准	软件工程和软件保证的方面，提供了一个系统的方法来标识、分析和跟踪危险和危险功能（例如，数据和指令）的软件缓解和控制，以确保软件在 系统中更加安全地运行 ”。
3	美国国防部的三军联合提出的软件系统安全手册	将系统安全性工程（包括软件系统安全性）定义为“在系统寿命周期各阶段运用工程和管理原理、准则和技术，以便在使用效能、时间和费用的约束范围内使 安全性最优并且风险降低 ”。
4	GJB/Z 102-97 《软件可靠性和安全性设计准则》	软件运行 不引起系统事故 的能力。
5	GJB/Z 142-2004 《军用软件安全性分析指南》	“软件具有的 不导致事故发生的能力 ”。确切的说，软件安全性是软件的功能安全性。
6	GJB 5236	软件产品在指定使用周境下，达到对人类、业务、软件、财产或环境造成损害的 可接受的风险 级别的能力。



基础知识2-软件安全性概念-启示

- 强调要在系统环境中讨论软件安全性
- 软件安全性是软件的一个质量属性或一种能力
- 软件安全是一个相对的概念，软件安全性的目的并非追求绝对安全，而是采取各种技术或方法使软件引起的安全风险在可接受范围内

安全性工作的本质：

- ①标识风险；
- ②实现安全性风险的消除或降低。



软件安全性-几个概念

○ 事故

造成人员伤亡、职业病、设备损坏或财产损失的一个或一系列意外事件。--GJB/Z 142-2004

○ 危险

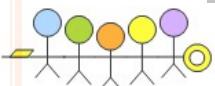
可能导致事故的状态。--GJB 900

可能导致或有助于事故或灾难（人员伤亡、或系统毁坏、或财产损失或环境破坏等）发生实际条件或潜在条件（1维）

○ 风险

不期望的事件或状态发生的严重度与可能性（2维）

$$\boxed{\begin{array}{l} \text{风险} \\ \text{损失发生概率} \end{array}} = \boxed{\begin{array}{l} \text{概率} \\ \text{事件发生概率} \end{array}} \times \boxed{\begin{array}{l} \text{严重度} \\ \text{事件损失度} \end{array}}$$





软件安全性相关标准

序号	标准名称及概况
1	MIL-STD-882系列，《系统安全性大纲要求》
2	Joint Software System Safety Committee, 美海陆空三军联合提出的《软件系统安全性手册》
3	NASA-STD-8719.13B 《软件安全性标准》 NASA-GB-8719.13 《软件安全性指南》
4	IEC61508 《电气/电子/可编程电子安全相关系统的功能安全》
5	EN-50128 《铁路应用：铁路控制和防护系统的软件》
6	DEF Stan 00-55 《防御设备安全相关软件要求》
7	ARP4761 《民航机载系统和设备的安全性评估过程指南和方法》 ARP4754 《高度整合或复杂航空器系统合格审定考虑》 RTCA DO-178B 《机载系统和设备认证中的软件考虑》
8	GJB 900-1990 《系统安全性通用大纲》
9	GJB/Z 99-1997 《系统安全工作手册》
10	GJB/Z102-1997 《软件可靠性和安全性设计准则》
11	GJB/Z 142-2004 《军用软件安全性分析指南》





理解：软件怎么能是危险的？

- 每个危险至少有一个原因，反过来，危险原因可能会导致一些后果（损害、疾病和经济损失）。
- **危险原因**可以是一个硬件缺陷或软件缺陷、一个人员操作错误、或者一个不期望的输入或事件，并且它会导致一个危险。
- 对于每一个危险原因，必须至少有一个控制方法，通常是一个设计特征（硬件/软件）或一个过程性步骤。
- **危险控制**是一种用于预防危险、降低危险发生可能性或者降低危险影响的方法。
- 危险控制使用硬件、软件、操作规程或者这些方法的组合，以避免危险。



原因	控制	控制措施示例
硬件	硬件	带有减压阀门的压力容器
硬件	软件	故障检测和安全保证功能；或者激活/点火检查，以便激活或防止危险的条件
硬件	操作员	操作员打开开关，以便切断失效单元的电源
软件	硬件	硬连接的计时器或者离散的硬件逻辑，以屏蔽非法的命令或者数据。传感器直接启动一个安全性开关，以禁用一个软件控制系统。机器人手臂的硬停止
软件	软件	两个独立的处理器，一个处理器检查另一个处理器，并在检测到故障时进行干预。模仿期望的性能，并检测偏离
软件	操作员	操作员在显示屏上观察控制参数的违反，并终止处理
操作员	硬件	在点火电路中串联三个电器开关，以容许两个操作员错误
操作员	软件	软件确认由操作员启动的危险开关。软件防止在不安全的模式下运行
操作员	操作员	两个机组人员，一个机组成员发出命令，另一个机组成员进行监督





危险的软件

它是一个危险原因

它是一个危险控制

它为安全性关键的决策提供信息

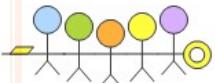
它被用作一种失效/故障检测的手段

它能影响危险软件的所有软件



软件控制危险的示例

- 监视危险的硬件（通过工具）并在偏离超出已确立的限度时执行一个纠正措施
 - 例如，在电源处于过压状态时，关掉电源（或者降低功率）
- 监视潜在的危险条件（例如温度）并警告操作员
 - 例如，在压力超过预定的阀值时，发出声音报警
- 禁止某些可能导致危险事件的活动处于运行状态
 - 例如，在存在有毒气体的情况下，在试验序列期间防止毒气室舱门被打开





基础知识3-软件可靠性安全性关系

?



软件可靠性**包含**软件安全性



软件可靠性**被包含于**软件安全性



软件可靠性与软件安全性**没有关系**



软件可靠性与软件安全性**侧重点不一样**



硬件安全性与可靠性的关系

硬件安全性

硬件可靠性

安全性是一种状态	可靠性是一种能力
安全性研究对象：危险	可靠性研究对象：故障
安全性重点关注对于危险源的控制，如产品自身固有的危险特性（如能量或毒性）、产品（硬件或软件）的故障、有害的环境等	可靠性重点关注产品在工作中出现的故障的控制



软件安全性与可靠性的关系

软件程序
状态的改
变过程

- ①有些状态可能是不安全状态，会导致安全事故的发生。
- ②有些状态可能引起软件失效的状态，导致不能实现功能。
- ③有些状态上述二者都涉及。

在规定的时间内，如果软件运行的真实环境与运行前规定的环境相关，则软件是可靠的就可判断软件是安全的





二、软件可靠性安全性分析概念及关系





分析1-软件可靠性分析内涵

- 软件在使用中发生失效会导致任务的失败（不可靠）。因此，应在软件设计过程中，对**可能发生的失效进行分析**（对影响可靠性大的失效），采取**必要的措施避免**引起失效的缺陷引入软件。
- 在系统测试、投入使用后对软件进行失效分析可以为**失效纠正措施**的制定提供依据，同时为避免类似问题的发生提供借鉴。这些工作将会大大提高使用中软件的可靠性，减少由于软件失效带来的各种损失。

第一步：识别出软件失效



第二步：制定出失效纠正措施





软件可靠性分析的常用方法

- 软件失效模式及影响分析 (SFMEA)
- 软件故障树分析 (SFTA)
- 其他软件可靠性分析方法
 - SFMEA与SFTA综合分析
 - 软件潜藏分析 (SCA)
 - Petri网分析
 -



分析2-软件安全性分析内涵

- 对和软件安全性相关的**特定信息**进行的系统而有序的**获取和评价**过程。--GJB/Z 142
- 通过对软件及其运行环境的分析，发现软件中与系统**危险条件**相关的设计缺陷及**危险产生条件**，并分析危险的发生概率，确认软件的危险风险指数。
- 软件安全性分析目的有两点：

**第一：识别出软件可能存在的
危险原因**

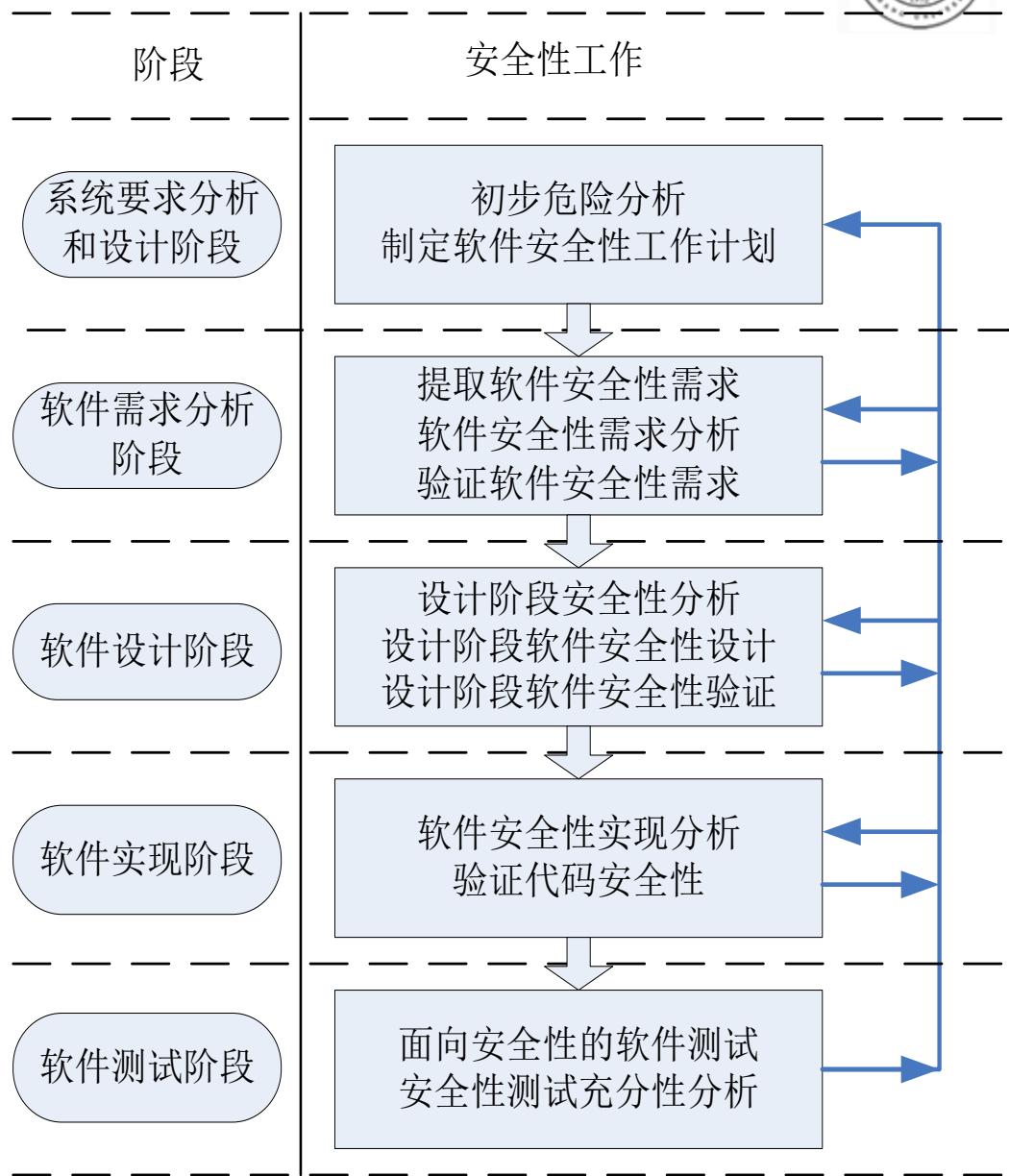


**第二：给出设计解决方法和验
证手段**



软件安全性工作

- 下面主要以NASA 8719.13的要求和方法为主线，结合软件工程现状，论述以危险控制为核心的软件安全性工作。
- 在软件开发过程的各阶段，主要的软件安全性工作如图所示：





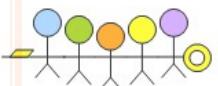
软件安全性分析任务-GJB/Z 142

- 软件需求安全性分析
- 软件结构设计安全性分析
- 软件支持工具和编程语言安全性分析
- 软件详细设计安全性分析
- 软件编码安全性分析
- 软件测试安全性分析
- 软件变更安全性分析



软件安全性分析的常用方法

- 特定的软件安全性分析方法
 - 软件失效模式及影响分析 (SFMEA)
 - 软件故障树分析 (SFTA)
 - 事件树分析 (ETA)
 - 危险与可操作性研究 (HAZOP)
 - 软件偏差分析 (SDA)
 - 基于系统故障理论的分析 (STAMP)
 -
- 通用的软件安全性分析方法
 - 初步危险分析 (PHA)
 - 功能危险分析 (FHA)
 - 软件子系统危险分析 (SSHA)
 -





分析3-软件可靠性安全性分析关系

软件可靠性分析

软件安全性分析

分析技术存在交叉，如FMEA、FTA、Petri网

侧重于分析软件失效严重性等级以及对软件可靠性影响较大的软件失效模式（发生概率较大的失效）

侧重于分析软件中与系统危险条件相关的设计缺陷及危险产生条件（后果严重的失效）

软件可靠性安全性分析

35



软件安全性分析的特点

- 软件安全性分析不能只从软件本身出发，**必须从系统角度进行分析**，考虑软件使用过程中软件、硬件和操作人员的相互作用，分析软件可能的工作时序、适用条件、逻辑缺陷及其可能造成的不利影响
- 软件安全性分析是系统安全性分析的一部分，必须在系统安全性分析的基础上进行
- **软件与硬件安全性分析的重点不同。**硬件安全性分析的重点为硬件故障以及共因失效所引起的系统危险状态。软件属于逻辑产品，无磨损。很多情况下并不是软件失效，而是在软件正常工作时，在某种特殊条件下软硬件相互作用导致的不安全情况。分析重点为软件设计缺陷，以及软件使用过程中软件、硬件和操作人员的相互作用。



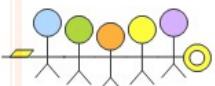
软件安全性分析的特点

- 与软件安全性测试的侧重点不同。软件安全性测试只能对软件输入组合所涉及的执行路径进行验证，而难以测试那些与某些特殊的输入组合和时序相关的软件设计缺陷。这种与软件的设计细节和软硬件交互时序等密切相关的软件设计缺陷必须采用软件安全性分析的方法。
- 全面系统化分析：由于软件的逻辑、数据、时序等设计缺陷，与软件相关的硬件故障和状态等都有可能引起软件失效，导致系统进入危险状态。因此分析时必须对软件进行全方位的分析，从系统顶层至软件的源代码，从外部的运行环境到软件内部的设计细节，包括软件的静态、动态、逻辑和物理模型。



软件安全性分析的特点

- **人员要求：**软件安全性分析必须从系统角度分析软件可能的运行时序、运行状态、适用条件、逻辑缺陷及其可能造成的不利影响，这就不可避免地会涉及到各种不同领域的专业知识与经验积累。因此，分析时以人为主进行分析，任何软件分析工具只能起辅助作用。这就对软件安全性分析人员提出了较高的要求。分析时要求有专门知识的软件安全性分析人员、熟悉系统结构的系统总体设计人员、软件设计人员、领域专家参加，共同工作





三、软件可靠性安全性设计概念及关系





设计1-软件可靠性设计内涵

• 软件可靠性设计的实质是在常规的软件设计中，应用各种必须的方法和技术，使程序设计在兼顾用户的各种需求时，全面满足软件的可靠性要求。

• 三点说明：

—**过程**：软件的可靠性设计应和软件的常规设计紧密地结合，贯穿于常规设计过程的始终。

—**范畴**：这里所指的设计是广义的设计，它包括了从需求分析开始，直至实现的全过程。

—**目的**：设计可靠的软件





软件可靠性要求

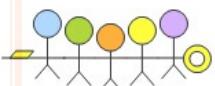
软件可靠性定性要求

定性要求：采用非量化的形式来设计、评价和保证软件的可靠性。

定性要求：包括定性设计要求和定性分析要求。

软件可靠性定量要求

定量要求：规定软件的可靠性参数、指标和评估、验证方法、用定量方法组织实施软件的可靠性设计、分析、测试、验证和管理。

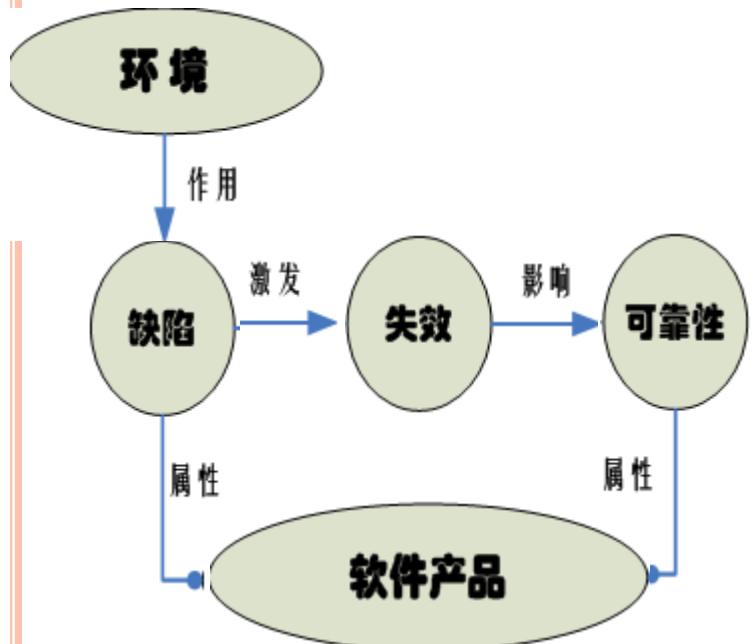




软件可靠性定性要求-示例

序号	定性设计方法	目的
1	可靠性设计准则	将可靠性要求及使用约束条件转化为软件开发设计的边界条件，规定专门的技术要求和设计准则，规范和约束软件的可靠性设计过程及行为
2	简化设计	在综合权衡的基础上，降低软件模块复杂性与结构复杂性以降低软件的总体复杂性，提高基本可靠性
3	重用设计	通过软件重用，在确保软件各组成单元可靠性的前提下，提高软件系统可靠性
4	健壮性设计	提高软件防止错误输入的能力以及在发生故障时能有效地控制故障的蔓延和扩散，确保软件的固有可靠性
5	容错设计	通过N版本程序设计和恢复块技术等实现软件容错，提高软件任务可靠性

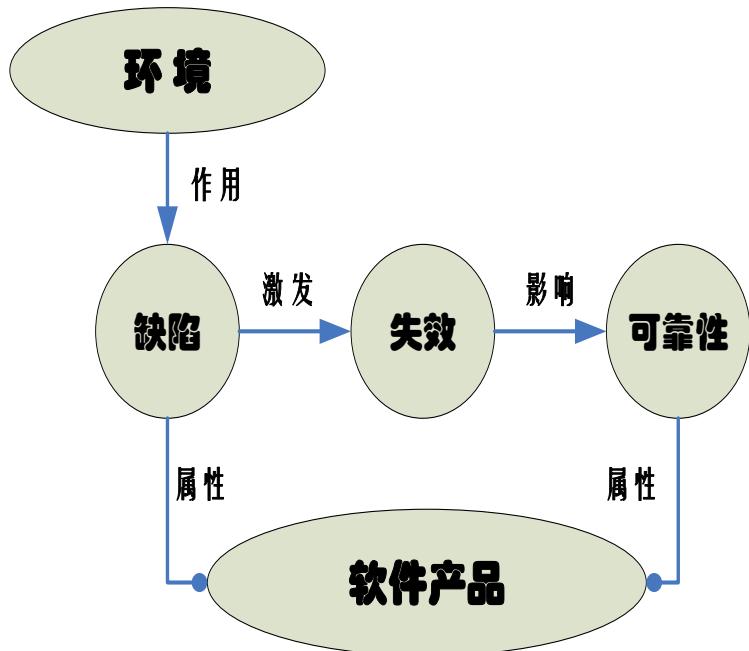
软件可靠性设计的实质



- 软件缺陷是软件的根本属性之一
- 软件**可靠性**是软件的生命，是软件的重要属性之一
- 软件**失效**是影响软件可靠性的关键
- 软件**缺陷**是导致软件失效的根本原因



软件可靠性设计的实质



- 软件可靠性设计的**目的**: 将可靠性设计到软件产品当中;
- 软件可靠性设计的**实质**: 在软件设计的全过程中与软件缺陷作斗争的过程。
 - 一方面: 尽量减少缺陷
 - 二方面: 避免缺陷暴露



软件可靠性设计的手段

手段



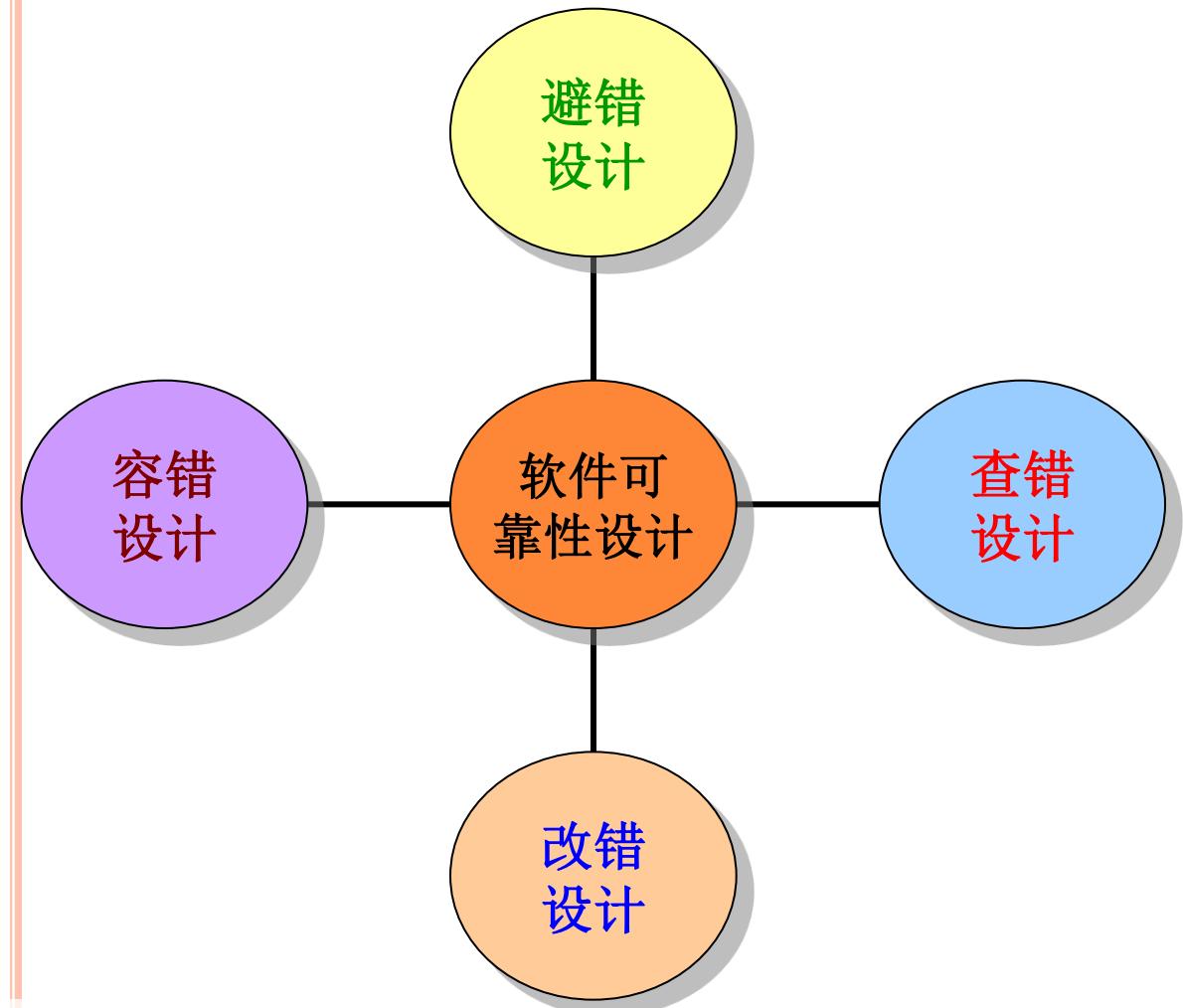
软件可靠性设计方法

软件可靠性设计准则

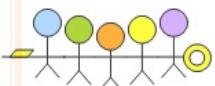




软件可靠性设计的方法

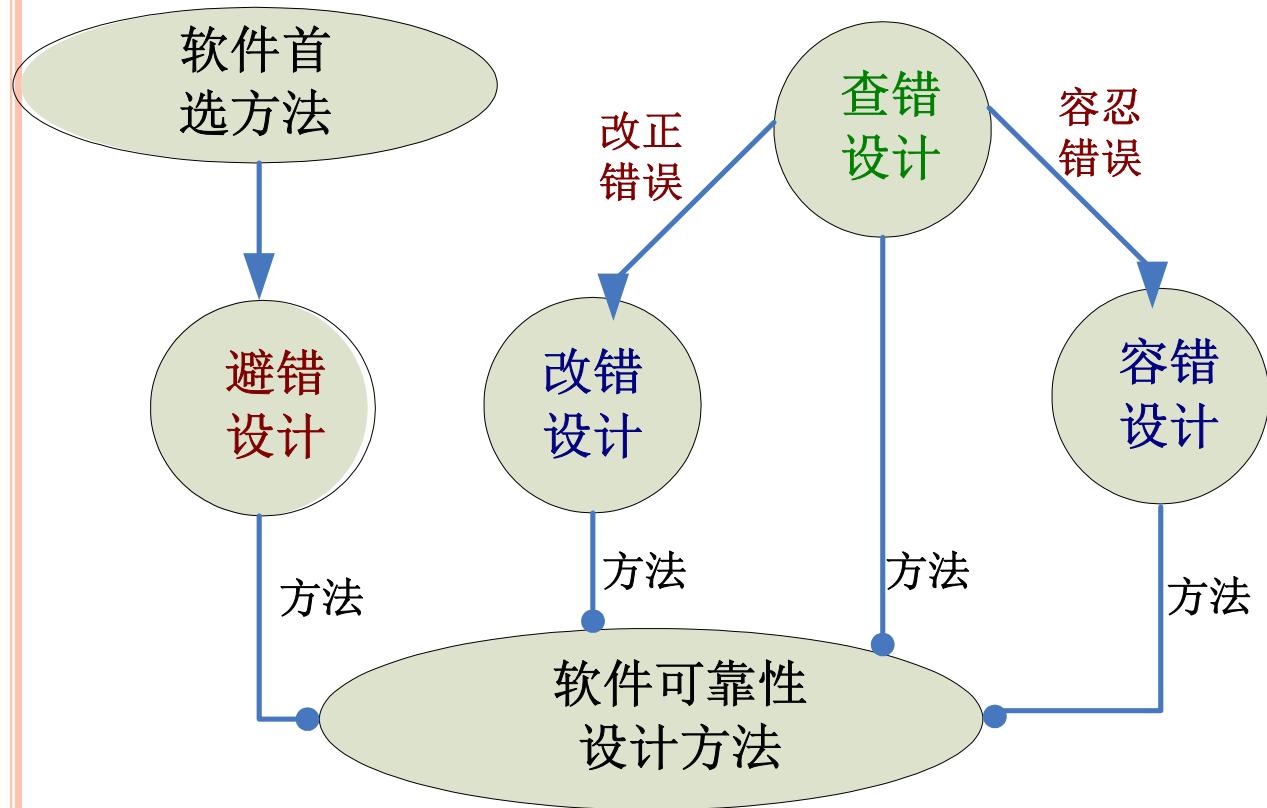


避错设计是使软件产品在设计过程中，少或多地产生错误，在程序中，或使程序在自指错误发生时，能自动地检测到错误、修正错误、减少错误、提高程序的健壮程度。避错设计是一种特殊的故障功能，使程序在错误已被触发的情况下，系统仍然具有正常运行能力的一种设计方法。





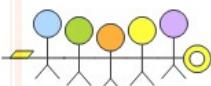
软件可靠性设计的手段



软件避错设计是
软件可靠性设计
的首选方法

软件查错设计是
软件改错设计和
容错设计的前提
条件

软件可靠性设计
方法从广义的分
类即可分为：软
件避错设计和容
错设计





软件可靠性设计的方法

软件避错设计 :尽量减少缺陷

避错设计是使软件产品在设计过程中，不发生错误或少发生错误的一种设计方法。总的设计原则是控制和减少程序的复杂性。软件避错设计体现了以预防为主的思想。

软件容错设计 :避免缺陷暴露

容错设计是指在设计中赋予程序某种特殊的功能，使程序存在缺陷的情况下，系统仍然具有正常运行能力的一种设计方法

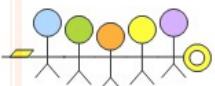




软件可靠性设计准则

○什么是软件可靠性设计准则

- 是一种**设计规范**，从软件可靠性角度出发，设计人员必须遵守的设计要求，是已有的、相似软件的工程经验的**总结**，并系统化、科学化、规范化而成。

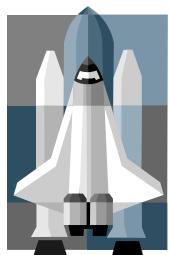




设计2-软件安全性设计内涵

软件的安全性设计主要是确保软件**不引起**或导致系统进入一个**危险**的状态。

软件应能检测出系统是否进入一个危险的状态，并在进入危险状态时采取纠正措施；以及在出现故障时能减轻损失。



危险消除与控制





危险消除与控制方针-NASA

- 消除危险
- 最低风险设计
- 纳入安全性设备
- 提供警告或报警
- 制定管理规程并进行培训



设计3-软件可靠性安全性设计关系

- **设计过程相同：**应和软件的常规设计紧密地结合，贯穿于常规设计过程的始终。
- **设计目的不同：**
 - 可靠性：设计可靠的软件
 - 安全性：设计安全的软件
- **设计对象不同：**
 - 安全性：硬件的失效、操作人员的错误等也可能影响软件的正常运行，从而导致系统进入危险的状态，因此软件安全性设计时必须对这种危险情况进行分析，并在设计时加以消除和控制
 - 可靠性：仅针对系统要求和约束进行设计，考虑常规的容错需求，并不需要进行专门的危险分析





小结

- 一、软件可靠性安全性概念及关系
- 二、软件可靠性安全性分析概念及关系
- 三、软件可靠性安全性设计概念及关系

Thank You !

