第二章
软件可靠性分析

陈仪香

华东师范大学软件学院可信智能团队TrIG

2024年9月18日

TrIG 华东师范大学软件学院可信智能团队
Trustworthy Intelligence Group

1.1 软件

- 软件=程序+文档

著名计算机科学家徐家福

1.2 软件质量

质： 性质，一种事务区别于其他事务的根本属性

质量：产品、工作等的优劣程度。

陆书平，万森，张秋霞编，现代汉语字典，商务印书馆，2014.7

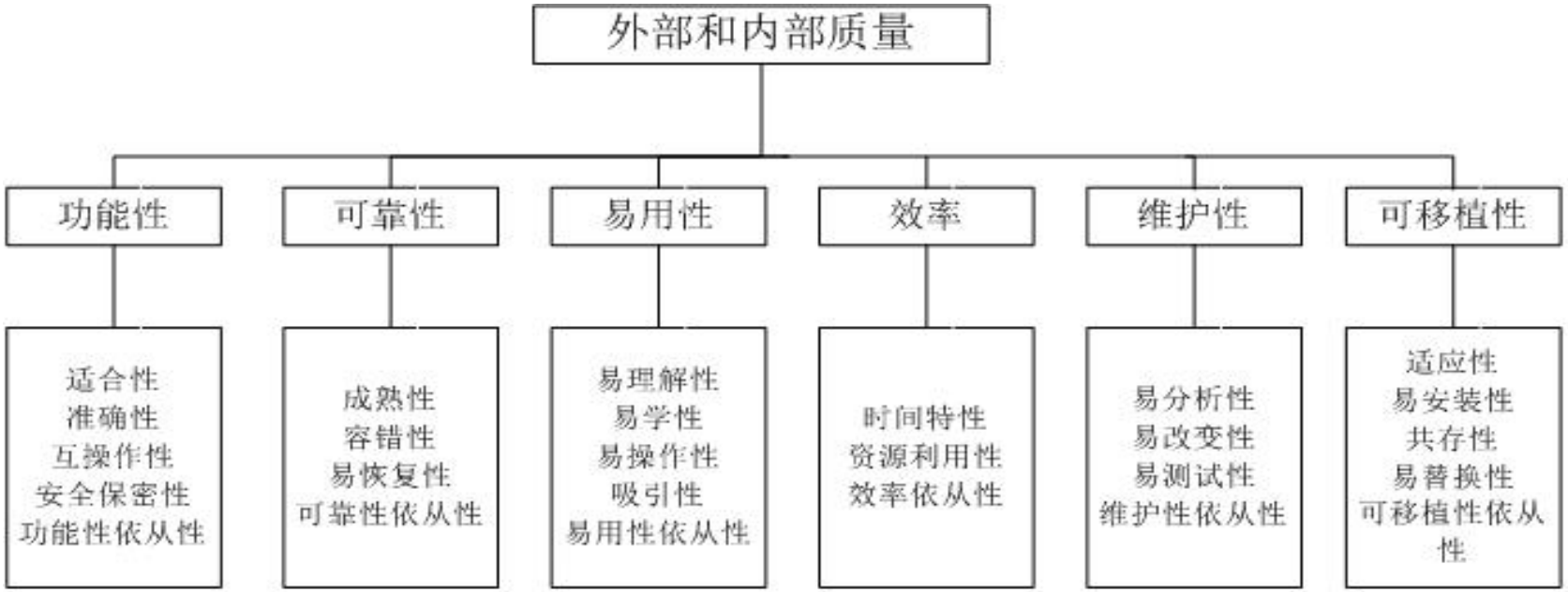
软件质量： 软件的优劣程度

软件=程序+文档（徐家福）

程序=算法+(程序设计语言)+数据结构（Wirth）

1.3 软件质量模型ISO 9126

1991年，ISO/IEC推出ISO/IEC 9126：1991《信息技术 软件产品评价 质量特性及其使用指南》，在此标准中，定义了六种质量特性，并且描述了软件产品评估过程的模型；



ISO 9126: Analysis of Quality Models and Measures，
Chapter · September 2010
DOI: 10.1002/9780470606834.ch10

1.3 软件质量模型

Gilles 1992

Six sessions were carried out with a range of companies from different market sectors: a management consultancy, a software house, a retail organization, a bank, a utility company and a manufacturing company。

The exercise defines relationships as:
(1) If characteristic A is enhanced, then Characteristic B is likely to be degraded, (-) or
(2) If characteristic A is enhanced, then Characteristic B is unlikely to be affected, (0) or
(3) If characteristic A is enhanced, then Characteristic B is likely to be enhanced (+).

学习什么内容？

课程内容安排为7章，与配套的教材相关，计划13周（上课26课时），17周考试。
（第2周中秋放假，第4周国庆节放假，第7周校运动会放假

学习什么内容

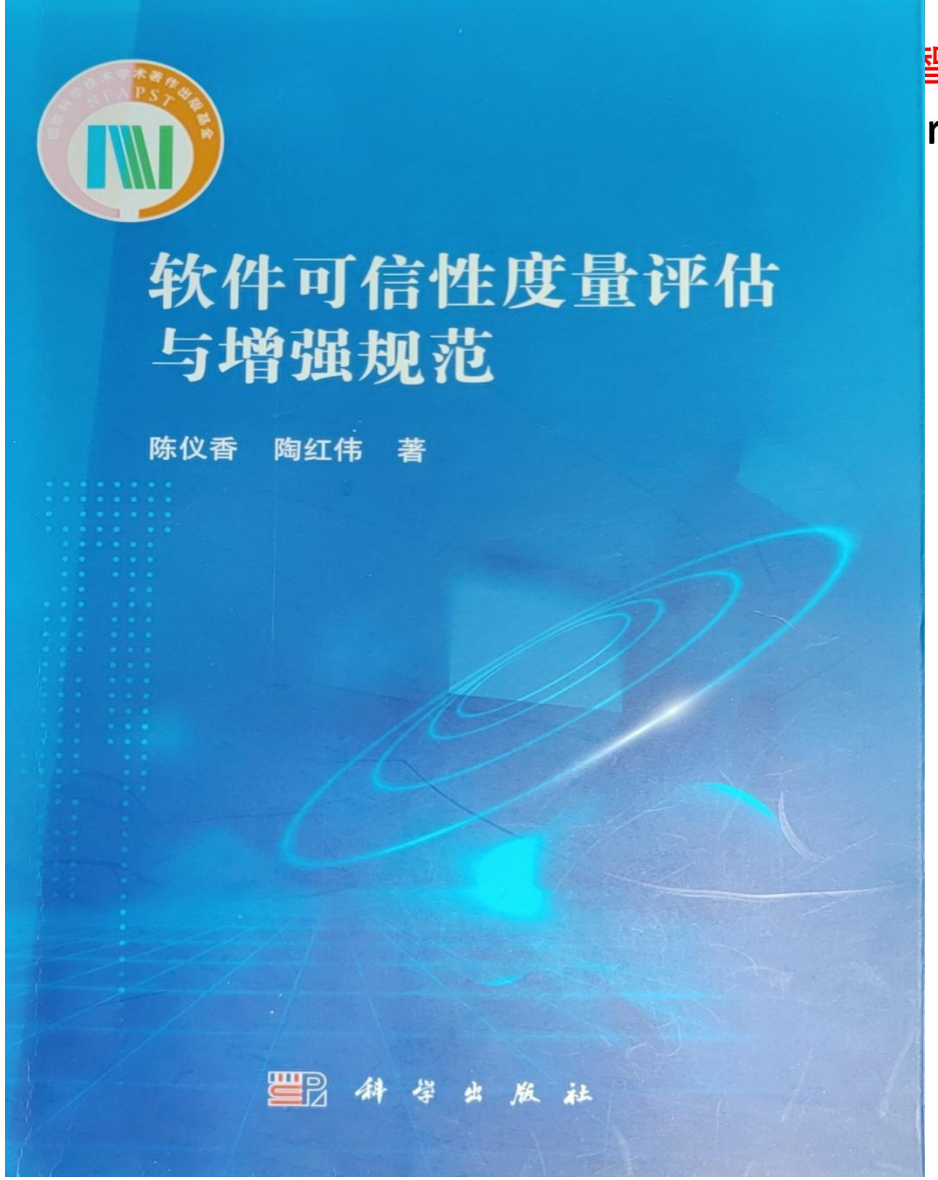
第一章 软件质量模型（支撑课程目标1、2）
本章重点解释软件质量的基本概念和模型，以及软件质量发展历史和标准。
讲授软件质量模型的概念和简史。

第二章软件可靠性模型（支撑课程目标1、2）
本章讲授软件可靠性概念和度量模型、以及软件可靠性分析。

第三章 软件可信性模型（支撑课程目标1、2）
本章讲授软件可信性的概念和发展过程，软件可信属性模型、软件属性分层模型和软件质量可信度量元模型。

第四章 基于属性的软件可信度量模型（支撑课程目标1、2）
本章讲授基于属性的软件可信度量性质和度量模型、基于属性分解的软件可信度量性质和度量模型、以及分级模型。

推荐教材



2.《软件可靠性工程》，主编：陆民 燕，国防工业出版社，2011.4年

1.《 软件可信性度量评估与增强规范 》（第一版），陈仪香，陶红伟著，科学出版社，2019年。

第二章 软件可靠性分析

2.1 软件可靠性

2.2 软件可靠性分析

2.3 软件可靠性评估

TrIG
华东师范大学软件学院可信智能团队
Trustworthy Intelligence Group

本章参考书

2.《软件可靠性工程》，主编：陆民 燕，国防工业出版社，2011.4年

2.1 软件可靠性

●按照GB/T11457-95-软件工程术语，软件可靠性的定义为：

（1） 在规定的条件下， 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运到已存在的缺陷（如果有缺陷存在的活）。

（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。

其中(1)是一个定量的定义，用此定义称之为可靠度更为确切，

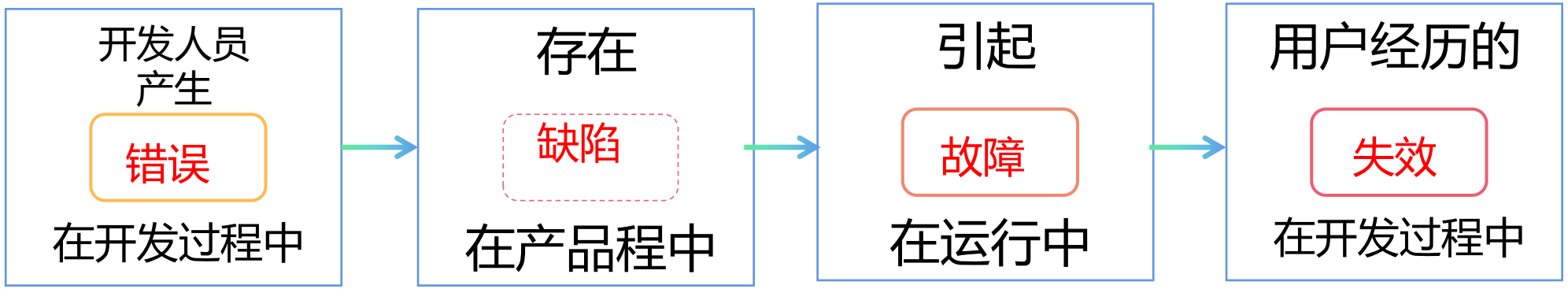
而（2）则是一个定性的定义。

2.1 软件可靠性

软件失效的随机性：从软件可靠性的定义可以看出，软件可靠性是使用概率度量的， 因此软件失效的发生是随机的。

但是经验又告诉我们，在使用一个程序是，在其他条件（如初始值、配置）不变的前提下，相同的输入数据会得到相同的输出结果，无论结果是否正确都是确定的，不会存在随机性问题。

那么如何理解软件失效的随机性问题呢?



2.1 软件可靠性--度量参数

我们将面向软件用户的可靠性度量参数称为软件可靠性参数。

可靠度：$R(t_0)=P(\xi>t_0)$

失效率$Z(t)=(1/R(t))\ f(t)$

失效强度：$\lambda(t)=\frac{\mathrm{d}u(t)}{\mathrm{d}t}$ 其中U(t)=E(N(t)), N(t)=t时刻发生失效的个数

平均失效前时间MTTF：MTTF是指当前时间到下一次失效时间的均值

平均失效间隔时间MTBF：两次相邻失效时间间隔的均值

缺陷密度DD：千行源代码的严重等级缺陷数的比例 $DD=\frac{D}{KSLOC}$

故障密度FD：千行源代码的严重等级失效的故障数（唯一性）的比例 $FD=\frac{F}{KSLOC}$

需求依从性：

2.1 软件可靠性--度量参数

需求依从性（Requirements Compliance）：

数据元素：

I_R 表示由于不一致的需求而引起的错误比例；

N_R 表示由于不完整的需求而引起的错误比例；

M_R表示由于曲解的需求而引起的错误比例。

N_1表示在一个版本或者模块中不一致的需求数。

N_2表示在一个版本或者模块中不完整的需求数。

N_3表示在一个版本或者模块中曲解的需求数。

计算公式：

$$I_R = (\frac{N_1}{N_1 + N_2 + N_3}) \times 100\%$$

$$N_R = (\frac{N_2}{N_1 + N_2 + N_3}) \times 100\%$$

$$M_R = (\frac{N_3}{N_1 + N_2 + N_3}) \times 100\%$$

$N_1+N_2+N_3$是一个版本或者模块中不一致、不完整、易曲解的需求数之和。

2.2 软件可靠性分析

- 软件可靠性分析是在软件设计过程中，对可能的失效进行分析，采取必要的措施避免引起失效的缺陷引入软件。
- 软件可靠性分析也可以在系统测试、设计定型或投入使用后， 为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，提高软件质量。

2.2 软件可靠性分析

本节介绍以下四种软件可靠性分析方法：

1、软件失效模式和影响分析（Software Failure Tree Modes and Effects Analysis, 软件FMEA）、（只介绍这种）

2、软件故障数分析（Software Fault Tree Analysis, 软件FTA）、

3、将二者结合的软件FEMA与FTA综合分析，以及

4、事件树分析

2.2 软件可靠性分析--软件FMEA

软件失效模式和影响分析（Software Failure Modes and Effects Analysis, 软件FMEA):

通过识别软件失效模式，分析造成的后果，研究分析各种失效模式产生的原因，寻找消除和减少其有害后果的方法，以尽早发现潜在的问题，并采取相应的措施，从而提高软件的可靠性和安全性。

FMEA是一种传统的可靠性、安全性分析方法，在硬件的可靠性工作中已获得了广泛应用，对提高硬件的可靠性、安全性发挥了重要作用。。

2.2 软件可靠性分析--软件FMEA

软件失效模式和影响分析（Software Failure Modes and Effects Analysis, 软件FMEA):

软件FMEA概念的提出始于1979年（Reifer),近年来，软件FMEA的应用有逐步增多的趋势，主要集中在嵌入式软件，并成功应用于安全关键领域，如医疗仪器、军用产品、汽车业等。

Reifer J. Software Failure Modes and Effects Analysis[c],IEEE Transactions on Rel

2.2 软件可靠性分析--软件FMEA

软件失效模式（Software Failure Modes）:指软件失效发生的不同方式。例如：输出结果错误、精度不满足要求。

软件失效就是泛指程序在运行中丧失了全部或部分功能、出现偏离预期的正常状态的事件或行为结果。比如：死机、计算结果错误等。

软件失效是有软件故障引起的，归根结底是由软件中潜藏的软件缺陷引起的，如逻辑遗漏、数据错误等。

软件失效模式分析是软件FMEA的基础，只有将被分析对象的所有可能的失效模式尽可能地全面分析出来，才能采取相应措施改进设计，防止失效现象的发生。

2.2 软件可靠性分析--软件FMEA

软件FMEA分两种分析方法：系统级软件FMEA和详细级软件FMEA。

系统级软件FMEA：在软件开发阶段的早期，即需求分析和设计阶段的早期。 用于发现软件需求或软件体系结构等存在的缺陷。其主要分析对象是软件需求分析阶段的软件功能或设计阶段的软件部件。

详细级软件FMEA：分析对象是已经编码实现的软件单元或由伪代码描述的单元， 因此至少在软件设计完成以后进行。通过分析单元的输入变量和算法失效模式，推导出对输出变量产生的影响。

这两种分析方法不同，失效模式也不同。

2.2 软件可靠性分析--软件FMEA

系统级软件失效模式：适用于系统级软件FMEA的分析对象。

在IEEE “Standard to Classification for Software Anomalies 93”标准中给出了软件异常的分类。如下所示，可作为失效模式的参考。IEEE的软件异常分类方法具有较高的公信度，但该标准不是专门针对软件FMEA制定的， 因此在应用中需根据具体工程项目适当变通。

2.2 软件可靠性分析--软件FMEA

9种系统级软件失效模式：IEEE 软件失效模式参考。

（1） 操作系统挂起

（2） 程序挂起

（3）程序失败：程序不能启动；程序运行不能终止；程序不能退出。

（4）输入问题：错误输入被接受；正确输入被拒绝；描述不正确或遗漏；参数不正确或遗漏。

（5）输出问题： 错误的格式；不正确的结果或数据；不完全或遗漏；拼写问题、语法问题。

2.2 软件可靠性分析--软件FMEA

9种系统级软件失效模式：IEEE 软件失效模式参考。

（6）未达到要求的性能：错误的格式；不正确的结果或数据；不完全或遗漏；拼写问题、语法问题。

（7）发现整个产品失败。

（8）系统错误信息。

（9）其他：程序运行改变了系统配置参数；程序运行改变了其他程序的数据；其他。

2.2 软件可靠性分析--软件FMEA

系统级软件失效模式：国军标软件失效模式指南。

在国军标《GJB/Z 1391-2006故障模式、影响及危害分析指南》中，给出了嵌入式软件FMEA的分析方法，其中软件的失效模式（系统级）见表2.2.1。在该表中，软件失效模式分为两大类：通用失效模式和详细失效模式。详细失效模式是对通用失效模式的细化，又分为五子类：输入失效、输出失效、程序失效、未满足功能及性能要求失效、其他类型。

在进行系统级软件FMEA时， 应根据被分析软件的不同特点选择合适的失效模式。同时，也可以在分析事假中，不断积累、丰富和完善软件实现模式。

2.2 软件可靠性分析--软件FMEA

序号	类别	软件失效模式示例			
1	软件通用失效模式	1） 运行时不符合要求			
		2）输入不符合要求			
		3）输出不符合要求			
2	软件详细失效模式	输入失效	1）未收到输入	输出失效	1）输出结果错误（如输出项缺损或多余等）
			2）收到错误输入		2）输出数据精度轻微超差
			3）收到数据轻微超差		3）输出数据精度中度超差
			4）收到数据中度超差		4）输出数据精度严重超差
			5）收到数据严重超差		5）输出参数不完全或遗漏
			6）收到参数不完全或遗漏		6）输出格式错误
			7）其他		7）输出打印字符不符合要求
					8）输出拼写错误/语法错误
					9）其他

2.2 软件可靠性分析--软件FMEA

序号	类别	软件失效模式示例			
2	软件详细失效模式	程序失效	1）程序无法启动	未能满足功能及性能要求失效	1）未达到功能/性能要求
			2）程序运行中非正常中断		2）不能满足用户对运行时间的要求
			3）程序运行不能终止		3）不能满足用户对数据处理量的要求
			4）程序不能退出		4）多用户系统不能满足用户数的要求
			5）程序运行陷入死循环		5）其他
			6）程序运行对其他单位或环境产生有害影响		
			7）程序运行轻微超时		
			8）程序运行明显超时		
			9）程序运行严重超时		
			10）其他		

2.2 软件可靠性分析--软件FMEA

序号	类别	软件失效模式示例		
2	软件详细失效模式	其他	1）程序运行改变了系统配置要求程序无法启动	6）人为操作错误
			2）程序运行改变了其他程序的数据	7）接口失效
			3）操作系统错误	8）I/O定时不准确导致数据丢失
			4）硬件错误	9）维护不合理/错误
			5）整个系统错误	10）其他

2.2 软件可靠性分析--软件FMEA

详细级软件失效模式: 主要针对变量的失效模式。

当变量被赋予非预期的值就会发生失效，这是最常见的失效模式，即“不正确的值”。

因为变量失效模式取决于变量类型， 而变量类型与特定的编程语言有关，因此变量的失效模式是与编程语言相关的。

针对一般通用的高级编程语言(C/C++/Java)，典型的变量失效模式如下表。

2.2 软件可靠性分析--软件FMEA--详细级软件失效模式

基本变量类型	扩展变量类型	失效模式	基本变量类型	扩展变量类型	失效模式
Char	unsigned chair	错误值 超出下限 超出上限	Float	short float	错误值 小于下限 大于上限 小于正确值 大于正确值
	signed chair			long float	
Bool	-------	应该为真但实际为假		unsigned	
		应该为假但实际为真		signed float	
Int	short int	错误值 小于下限 大于上限 小于正确值 大于正确值 等于某值 大于等于某值 小于等于某值	Double	short double	错误值 小于下限 大于上限 小于正确值 大于正确值
	long int			long double	
	unsigned int			unsigned double	
	signed int			signed double	
Bit	-------	应该为1但实际为0 应该为0但实际为1	Void*	Int*，Chair*，float*	指向空 指向错误地址

2.2 软件可靠性分析--软件FMEA

软件失效影响(software failure effect) 是指软件失效模式对软件系统的运行、功能或状态等造成的后果。例如软件失效会影响任务的完成和功能实现或造成设备损坏。

软件失效影响严酷度(Severity) 是指软件失效模式所产生的后果的严重程度。最严重的后果是导致人员伤亡、对环境造成灾难性破坏，而轻微的后果仅降低使用的舒适性、方便性等。

软件失效影响及其严酷度的确定可以为失效模式改进措施的制定提供依据。对于产生后果越严重的失效模式，越应采取有效措施加以概念，以避免危险事件的发生。

2.2 软件可靠性分析--软件FMEA

在国军标《GJB/Z 1391-2006故障模式、影响及危害性分析指南》中，给出了武器装备常用的严酷度类别及其定义。 见下表。在进行软件FMEA时，应根据软件的特点加以实例化，制定适合的失效影响严酷度类别。

严酷度类别	严重程度定义
I类（灾难的）	引起人员死亡或产品（如飞机、坦克、导弹及船舶等）毁坏、重大环境损害
II类（致命的）	引起人员的严重伤害或重大经济损失或导致任务失败、产品严重损坏及严重损害
III类（中等的）	引起人员的中等程度伤害或中等程度的经济损失或导致任务延误或降级、产品中等程度的损坏及环境损害
IV类（轻度的）	不足以导致人员伤害或轻度的经济损失或产品轻度的损坏及环境损害，但它会导致非计划性维护或修理

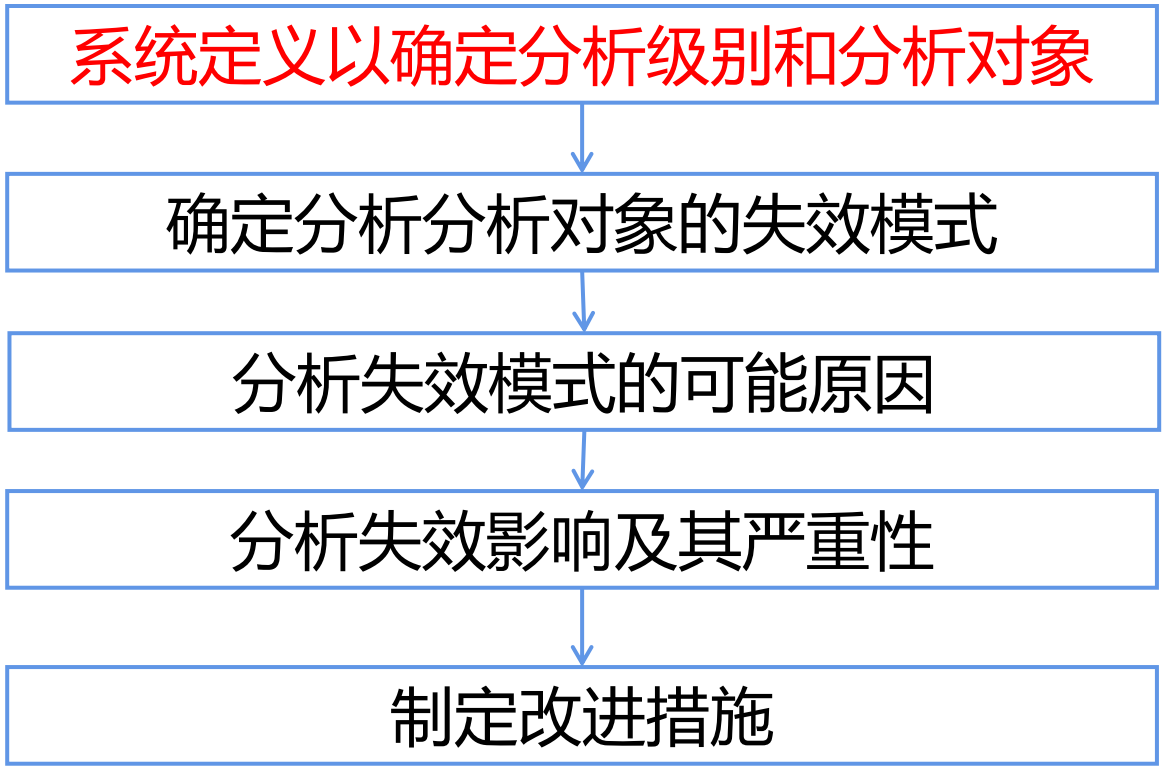
2.2 软件可靠性分析--软件FMEA

系统级软件FMEA分析步骤：一般包括系统定义、失效模式分析、失效原因分析、失效影响分析以及制定改进措施等五个步骤。

1） 系统定义

（1） 系统定义的主要目的是确定软件FMEA的分析级别（严酷度）和分析对象，以确定分析的重点。

（2）首先应说明系统的主要功能和次要功能、用途、系统的约束条件和失效判据等。 还应包含系统工作的各种模式的说明、系统环境条件以及软、硬件配置。

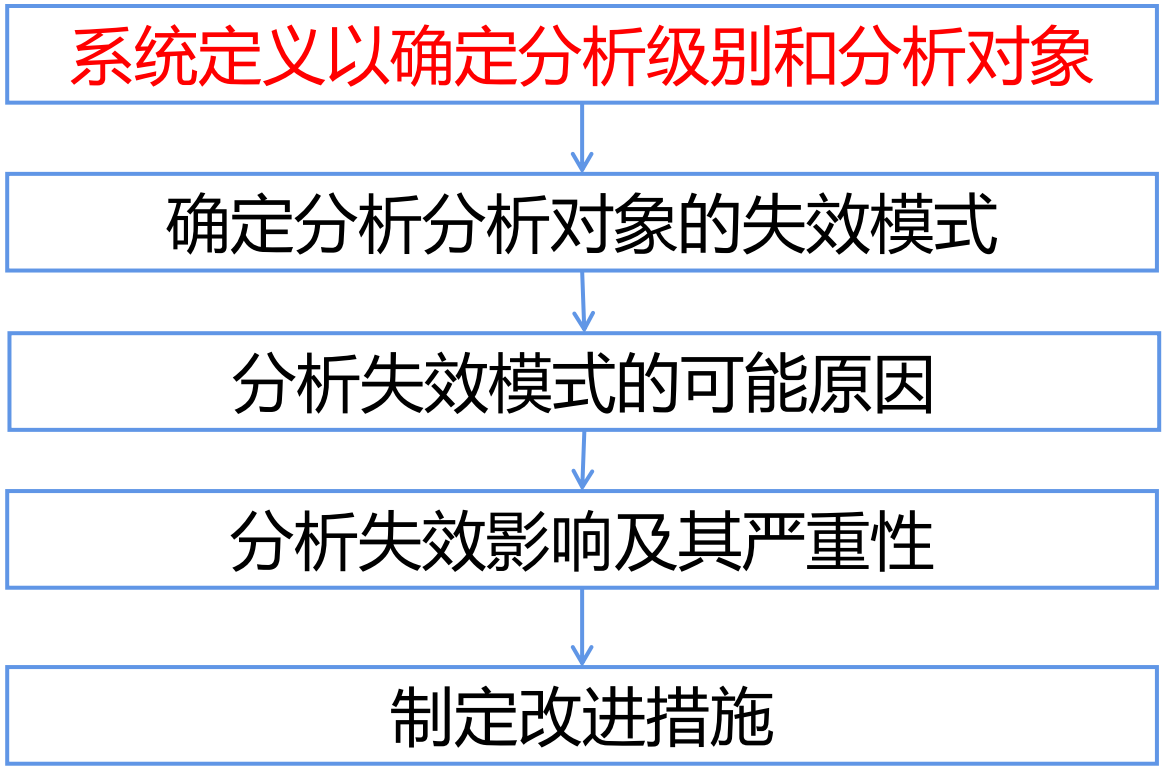


2.2 软件可靠性分析--软件FMEA

系统级软件FMEA分析步骤：一般包括系统定义、失效模式分析、失效原因分析、失效影响分析以及制定改进措施等五个步骤。

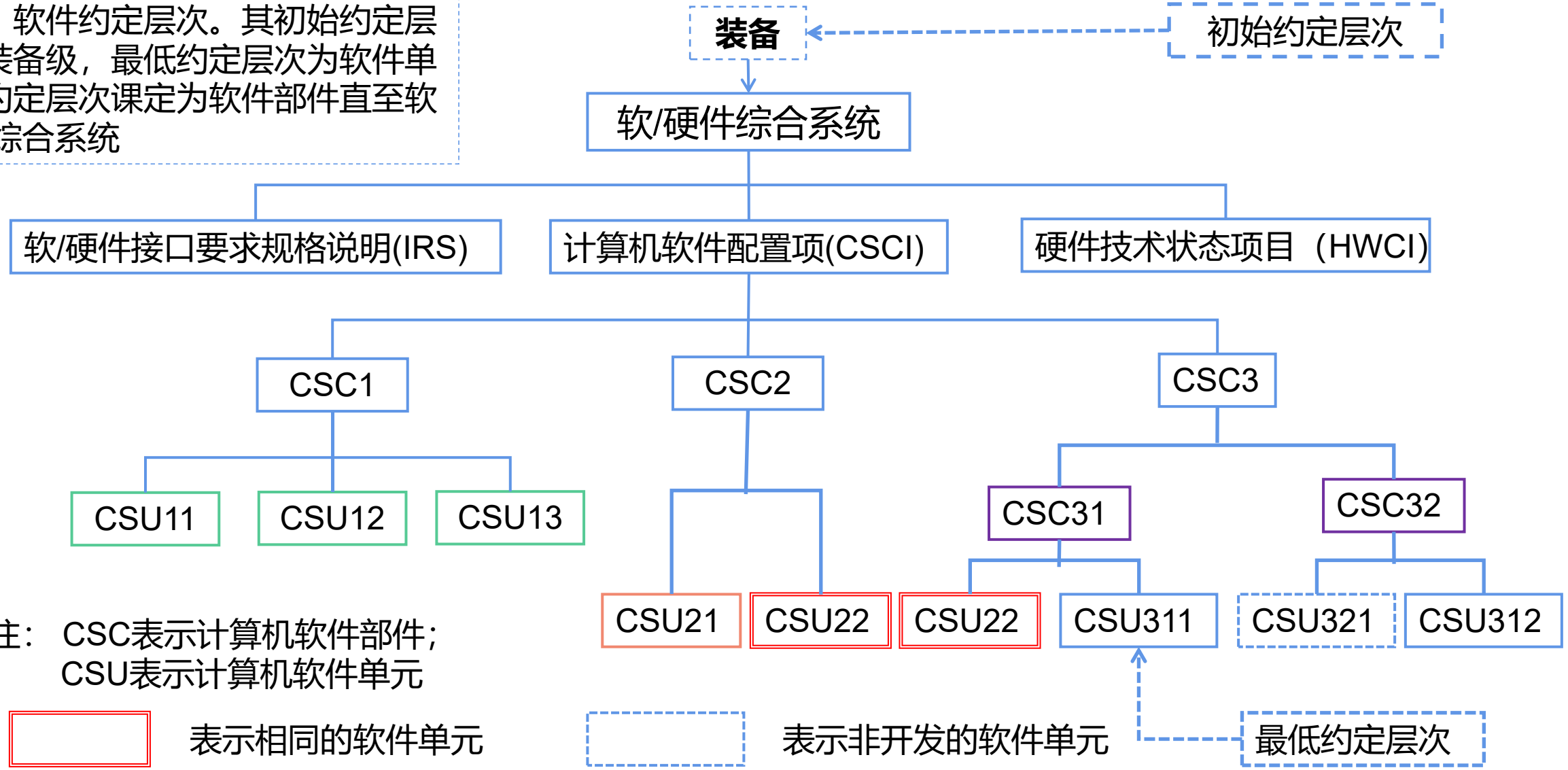
1） 系统定义

（3） 根据软件系统的功能、结构特征等层次结构确定系统的分析级别，即约定层次，以及分析对象，如功能模块、软件部件或单元等。 在层次结构的高层较易进行全面的分析，而在低层因可供参考的信息更丰富，因而分析更深入、但工作量会相应地增大。环境条件以及软、硬件配置。



2.2 软件可靠性分析--软件FMEA

示例：软件约定层次。其初始约定层次为装备级，最低约定层次为软件单元，约定层次课定为软件部件直至软/硬件综合系统



注： CSC表示计算机软件部件；
CSU表示计算机软件单元

表示相同的软件单元

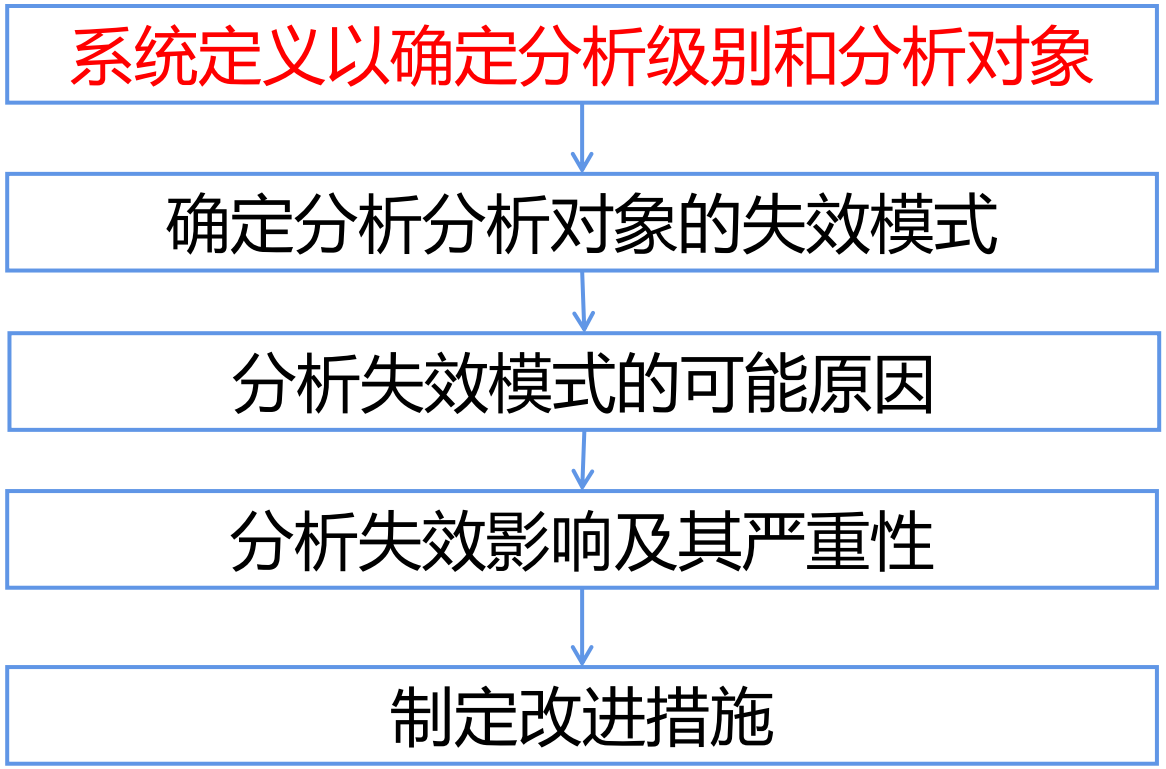
表示非开发的软件单元

2.2 软件可靠性分析--软件FMEA

系统级软件FMEA分析步骤：一般包括系统定义、失效模式分析、失效原因分析、失效影响分析以及制定改进措施等五个步骤。

1） 系统定义

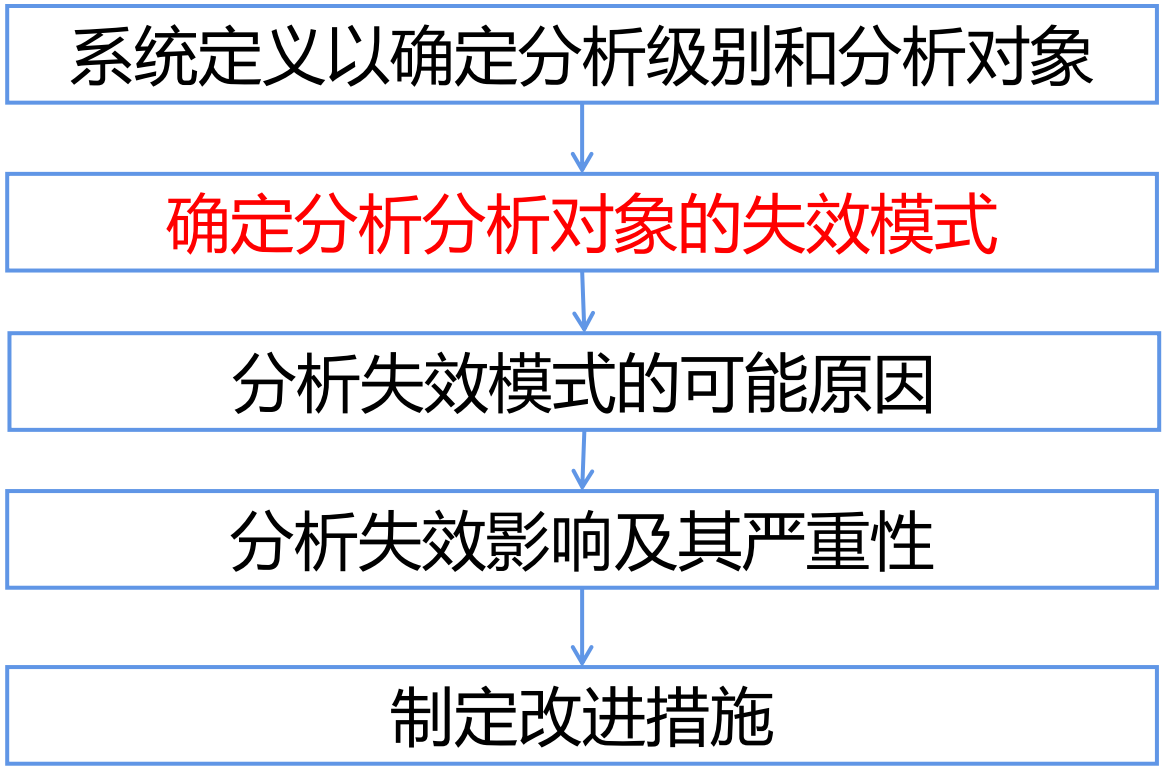
（4） 若受时间或经费等因素的影响无法对整个软件系统进行全面的分析时，可在分析前确定分析的重点。通过识别对系统功能和安全性（某个非功能属性：正确性、易用性等）影响较大的危险事件，确定对上述危险事件的出现有直接或间接关系的功能模块、软件部件等，作为软件FMEA分析的重点。



2.2 软件可靠性分析--软件FMEA

系统级软件FMEA分析步骤：一般包括系统定义、失效模式分析、失效原因分析、失效影响分析以及制定改进措施等五个步骤。

2） 失效模式分析
针对每个分析对象，参考失效模式，例如： 响应时间超时、输出错误值、不能满足用户对运行时间的要求等。

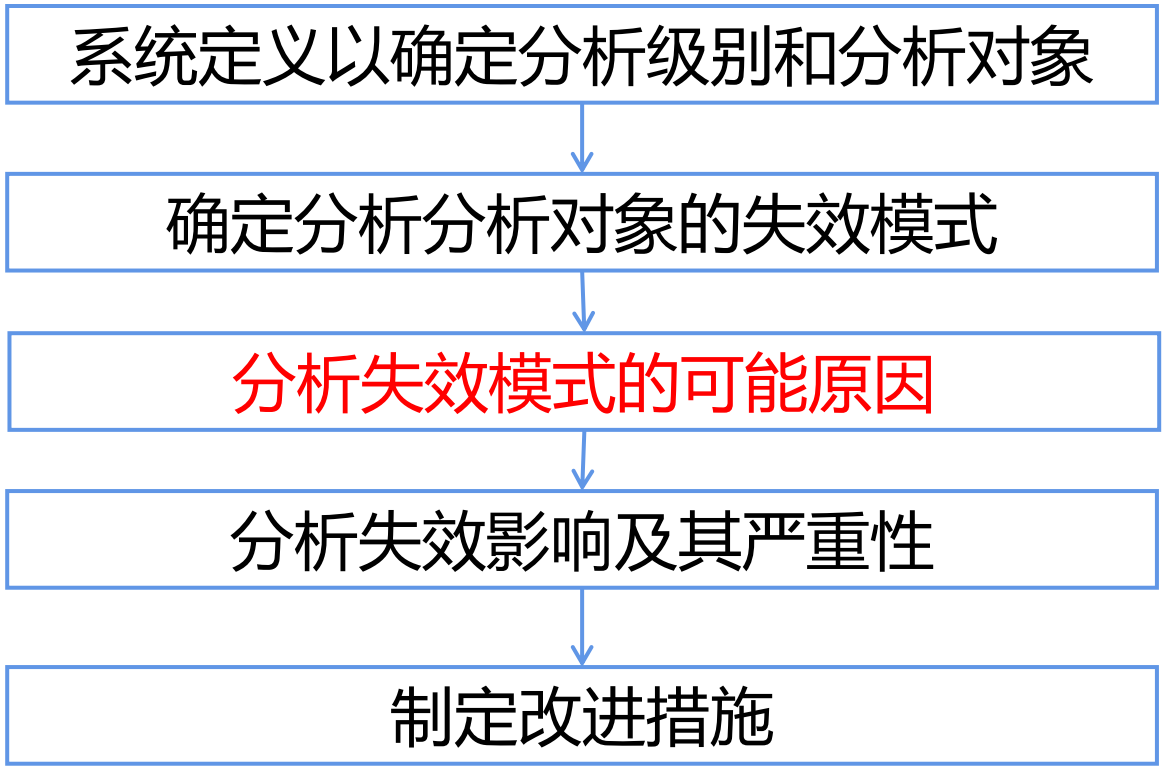


2.2 软件可靠性分析--软件FMEA

系统级软件FMEA分析步骤：一般包括系统定义、失效模式分析、失效原因分析、失效影响分析以及制定改进措施等五个步骤。

3） 失效模式原因

分析每个失效模式的所有可能原因（参照表）。软件失效的原因是软件中潜藏的缺陷，一个软件失效的产生可能是由一个软件缺陷引起的，也可能是由多个软件缺陷共同作用引起的。在进行失效原因分析时应尽可能全面地分析所有可能得软件缺陷，为制定改进措施提供依据。



2.2 软件可靠性分析--软件FMEA

潜在的软件失效原因表

一般失效原因	具体失效原因	一般失效原因	具体失效原因
逻辑遗漏或执行错误	● 遗忘细节或步骤 ● 逻辑重复 ● 忽略极限条件 ● 不必要的函数 ● 需求的错误表述 ● 未进行条件测试 ● 检查错误变量 ● 循环错误	软硬件接口失效	● 中断句柄错误 ● I/O时序错误 ● 时序错误导致数据丢失 ● 子函数调用不当 ● 子函数调用位置错误 ● 调用不存在的子函数 ● 子函数不一致
算法的编码错误	● 等式不完整或不正确 ● 丢失运算结果 ● 操作数错误 ● 括号使用错误 ● 精度损失 ● 舍入和舍去错误 ● 混合类型 ● 标记习惯不正确	数据错误或丢失	● 传感器数据错误或丢失 ● 操作数据错误或丢失 ● 嵌入到表中的数据错误或丢失 ● 外部数据错误或丢失 ● 输出数据错误或丢失 ● 输入数据错误或丢失

2.2 软件可靠性分析--软件FMEA

潜在的软件失效原因表

一般失效原因	具体失效原因
数据操作错误	● 数据初始化错误 ● 数据存取错误 ● 数据打包解包错误 ● 标志或索引设置不当 ● 变量参考错误数据 ● 数据越界 ● 变量缩放比率或单位不正确 ● 变量维度不正确 ● 变量类型错误 ● 变量下标错误 ● 数据范围不对

2.2 软件可靠性分析--软件FMEA

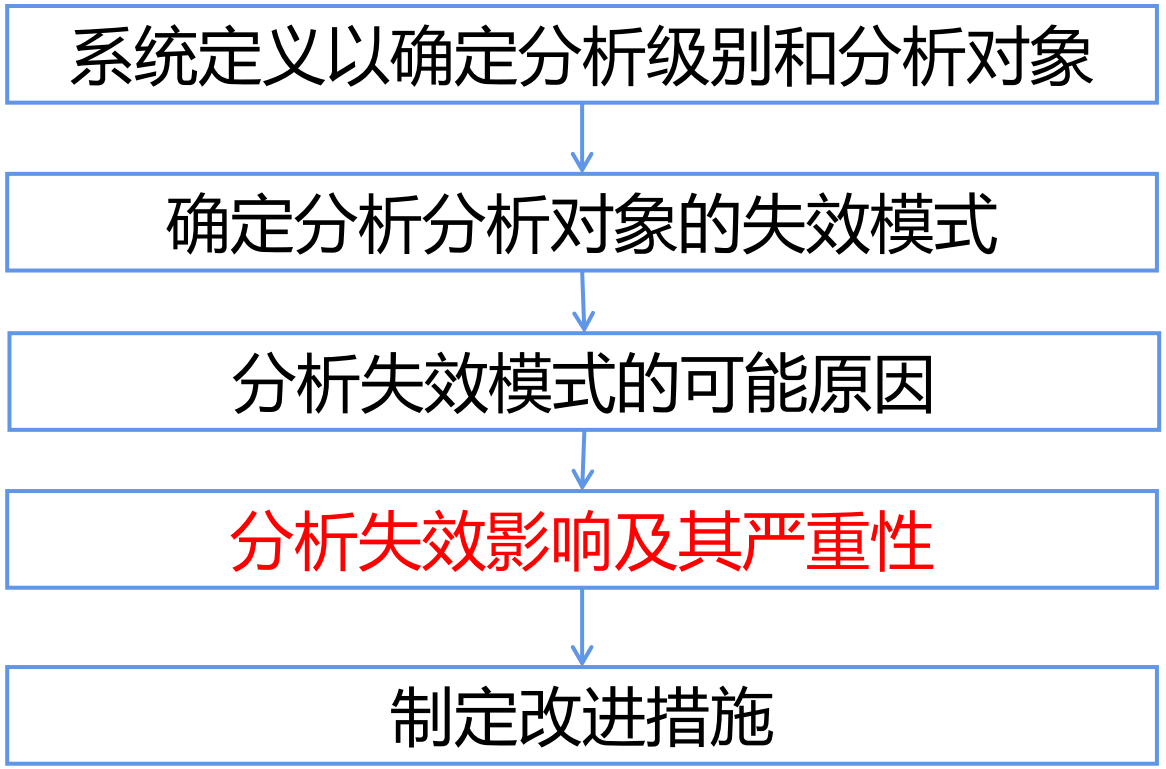
系统级软件FMEA分析步骤：一般包括系统定义、失效模式分析、失效原因分析、失效影响分析以及制定改进措施等五个步骤。

4） 失效影响分析

分析每个失效模式对局部、高一层，直至整个系统的影响，以及失效影响的严重性。

分析失效影响及其严重性的目的是识别软件失效所造成后果的严重程度，以便按照优先级为不同严重等级的失效制定改进措施。

可参照失效影响的严酷度类别分类确定软件失效影响的严酷度。



2.2 软件可靠性分析--软件FMEA

系统级软件FMEA分析步骤：一般包括系统定义、失效模式分析、失效原因分析、失效影响分析以及制定改进措施等五个步骤。

5） 制定改进措施

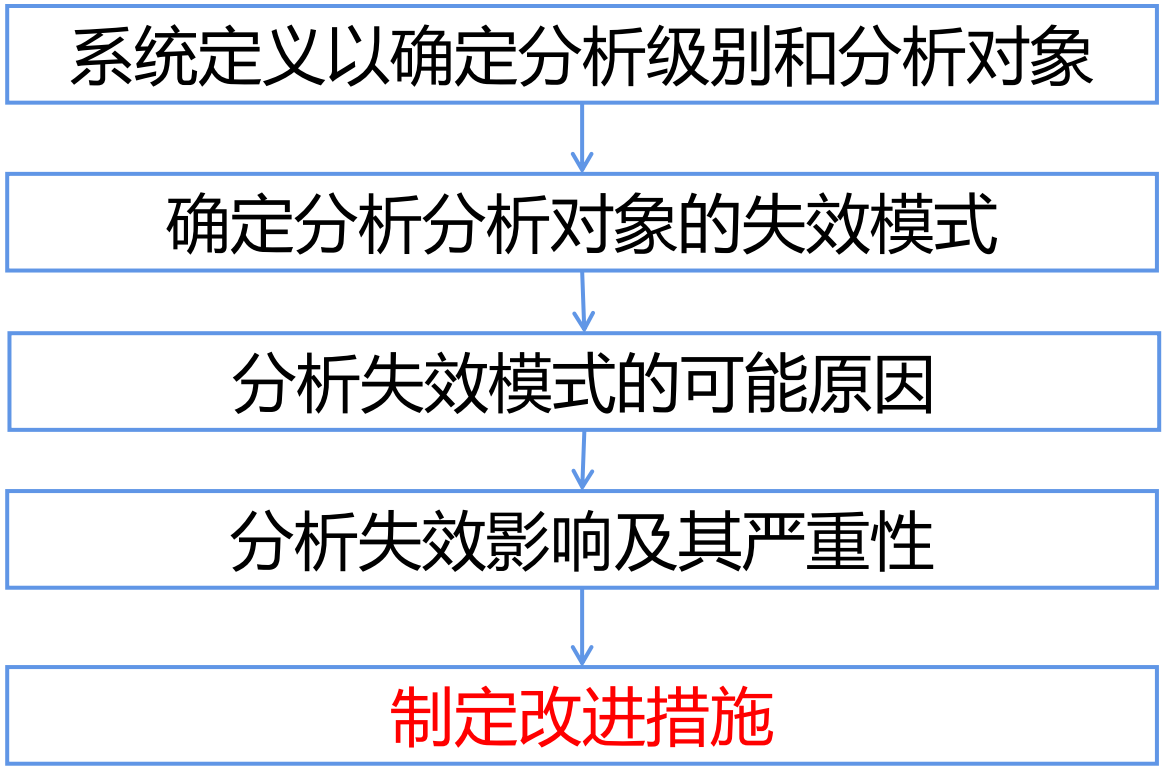
根据上述分析得到的失效产生的原因及影响的严重性等，确定出需要采取的改进措施。

改进措施主要有两种途径：

一是修改软件需求、设计或编码中的缺陷，增加软件的防护措施；

二是增加硬件防护措施。

进行软件FMEA时，应填写FMEA表。该表应能完整地体现分析的目的和取得成果。



2.2 软件可靠性分析--软件FMEA

软件FMEA 表示例

编号	单元	功能	失效模式	可能的失效原因	失效影响			严酷度	改进措施
					局部影响	高一层次影响	最终影响		
1.1	输出	输出数据提交用户显示	数值高于正常范围	逻辑问题 计算问题 数据操作问题	N/A	无	任务降级	III(中等)	.....
1.2			数值低于正常范围	逻辑问题 计算问题 数据操作问题	N/A	无	任务降级	III（中等)	.....
1.3									
1.4			输出数据没有显示	逻辑问题 计算问题 数据操作问题	N/A	无	任务中止	II或I致命或灾难	.....
1.X			.....	.....	.....	.....	.....	.....	.....

2.2 软件可靠性分析--软件FTA

软件危害性分析（Software Criticality Analysis, SCA）

- 危害性(criticality)是指对某一失效模式的后果及其发生概率的综合度量。
- 软件危害性分析是对软件的每一失效模式的严重程度及其发生的概率所产生的综合影响进行划等分类，以全面评价软件各种可能出现的失效模式的影响。

2.2 软件可靠性分析--软件FTA

软件危害性分析（Software Criticality Analysis, SCA）

软件危害性分析必须在软件FMEA工作基础上进行。GJB/Z1391-2006推荐采用功能及硬件FMEA中的风险优先数(RPN,Risk Priority Number)方法进行软件的危害性分析。

软件风险优先数SRPN的计算方法如下：

$$SRPN=SESR\times SOPR\times SDDR$$

其中：SESR为软件失效模式影响的严酷度等级；SOPR为软件失效模式的发生概率等级；SDDR为软件失效模式被检测难度等级。

2.2 软件可靠性分析--软件FTA

软件失效模式影响的严酷度等级(SESR)的评分准则

软件失效模式影响的严重性	软件失效模式影响的程度	评分等级	软件失效模式影响的严重性	软件失效模式影响的程度	评分等级
极高且无警告提示	影响系统运行的安全性，或不符合国家安全规定，且不能发出警告	10	低	系统仍能运行，但影响使用的方便与舒适性	5
极高且有警告提示	影响系统运行的安全性，或不符合国家安全规定，但能发出警告	9	较低	影响轻度	4
非常高	影响系统主要功能，且不能运行	8	非常次要的	影响轻微	3
高	系统仍能运行，但运行水平降级，用户不满意	7	极次要的	影响极小	2
中等	系统仍能运行，但丧失使用的方便与舒适性	6	无	无影响	1

2.2 软件可靠性分析--软件FTA

软件失效模式发生概率等级（SOPR）的评分准则

软件失效模式发生的可能性	软件失效模式发生概率Pm参考范围（每单元）	评分等级	软件失效模式发生的可能性	软件失效模式发生概率Pm参考范围（每单元）	评分等级
非常高（几乎不可避免发生失效）	$P_m \geq 5\times10^{-1}$	10	低（相对几乎无失效）	$1\times10^{-4} \leq P_m < 2\times10^{-4}$	4
	$1\times10^{-1} \leq P_m < 5\times10^{-1}$	9		$2\times10^{-5} \leq P_m < 1\times10^{-4}$	3
高（重复失效）	$1\times10^{-2} \leq P_m < 1\times10^{-1}$	8	非常低（几乎不可能失效）	$1\times10^{-5} \leq P_m < 2\times10^{-5}$	2
	$1\times10^{-3} \leq P_m < 1\times10^{-2}$	7		$2\times10^{-6} \leq P_m < 1\times10^{-5}$	1
中等（偶然失效）	$1\times10^{-3} \leq P_m < 2\times10^{-3}$	6			
	$2\times10^{-4} \leq P_m < 1\times10^{-3}$	5			

2.2 软件可靠性分析--软件FTA

软件失效被检测难度等级(SDDR)的评分准则

软件失效被检测的可能性	软件失效模式发生概率Pm参考范围（每单元）	评分等级	软件失效模式发生的可能性	软件失效模式发生概率Pm参考范围（每单元）	评分等级
完全不能确定	$P_D<2\times10^{-6}$可能发现失效原因/机理和失效模式，或根本无此类检测装置	10	中等	$1\times10^{-2}<P_D\leq2\times10^{-2}$可能发现失效原因/机理和失效模式	5
非常微小	$P_D\approx2\times10^{-6}$可能发现失效原因/机理和失效模式，	9	中等偏高	$2\times10^{-2}<P_D\leq5\times10^{-2}$可能发现失效原因/机理和失效模式	4
微小	$2\times10^{-6}<P_D\leq2\times10^{-4}$可能发现失效原因/机理和失效模式	8	高	$5\times10^{-2}<P_D\leq3.3\times10^{-1}$可能发现失效原因/机理和失效模式	3
非常低	$2\times10^{-4}<P_D\leq2\times10^{-3}$可能发现失效原因/机理和失效模式	7	非常高	$3.3\times10^{-1}<P_D$可能发现失效原因/机理和失效模式	2
低	$2\times10^{-3}<P_D\leq1\times10^{-2}$可能发现失效原因/机理和失效模式	6	完全确定	$P_D\approx0$完全能发现失效原因/机理和失效模式	1

2.2 软件可靠性分析--软件FTA

软件FMEA实施注意事项

1. 软件FMEA的应用重点在软件开发过程的早期，如需求分析与概要设计阶段，找出可能存在的与功能和性能相关的缺陷，以尽早完善需求分析与概要设计。

2. 根据分析阶段和级别不同，通常有两种分析方法、即系统级软件FMEA和详细级软件FMEA。详细级软件FMEA一般使用于可靠性、安全性关键单元，不适宜在整个软件范围内的应用。

2.2 软件可靠性分析--软件FTA

软件FMEA实施注意事项

3. 明确和掌握软件FMEA基本步骤中的主要内容：

(1) 软件失效模式是软件FMEA的基础。 失效模式的分析是否全面合理决定了软件FMEA的分析效果，是整个分析过程中最为关键的一步。分析人员可参考相关标准中或根据经验积累的软件失效模式，并结合软件的具体特点选择适用的失效模式。应积累开展软件FMEA的有关信息，并建立相应信息库，为有效开展软件FMEA提供支持。

2.2 软件可靠性分析--软件FTA

软件FMEA实施注意事项

3. 明确和掌握软件FMEA基本步骤中的主要内容：

(2) 软件失效原因是由于软件缺陷在运行时被触发而产生的。对软件FMEA失效原因分析就是要找出潜在的软件缺陷。

（3）失效影响既可对每一个软件失效模式造成的“局部、高一层次和最终”三级影响进行分析，又可对“具备、高一层次”的影响进行分析，或直接对“局部、最终”影响进行分析。

（4）分析对象既可以是软件功能模块、软件部件，又可以是底层的软件单元。

2.2 软件可靠性分析--软件FTA

本例已某型发动机控制软件为例说明系统级FMEA的分析方法

该型发动机控制软件（简称发控软件）嵌入在数字电子控制盒内，完成观测发动机状态、控制输出燃油、以及与飞行控制软件进行通讯等功能。

1） 系统定义

该系统功能示意图如图5-7。该发控软件主要通过采集频率量、模拟量和开关量等信号，进行状态判断，根据不同状态对发动机燃油进行相应的控制。

2.2 软件可靠性分析--软件FTA

1） 系统定义

该系统功能示意图如图5-7。

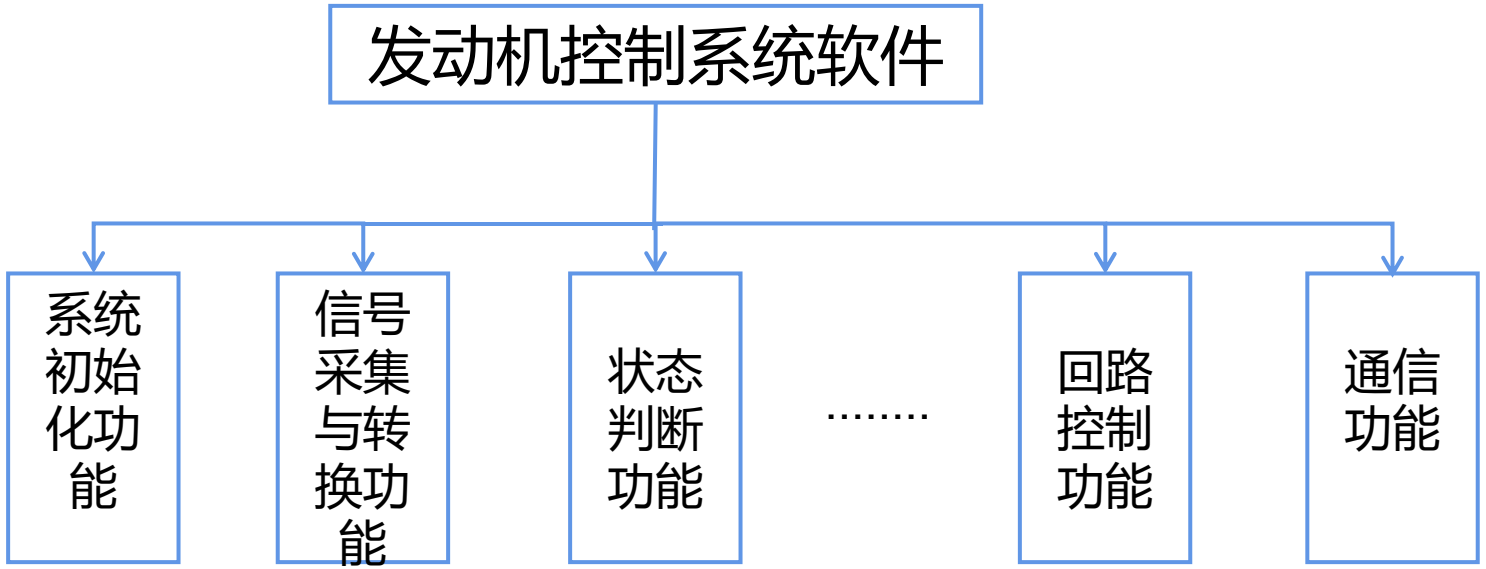


图5-7 发控软件功能图

2.2 软件可靠性分析--软件FTA

1） 系统定义

该系统功能示意图如图5-7。

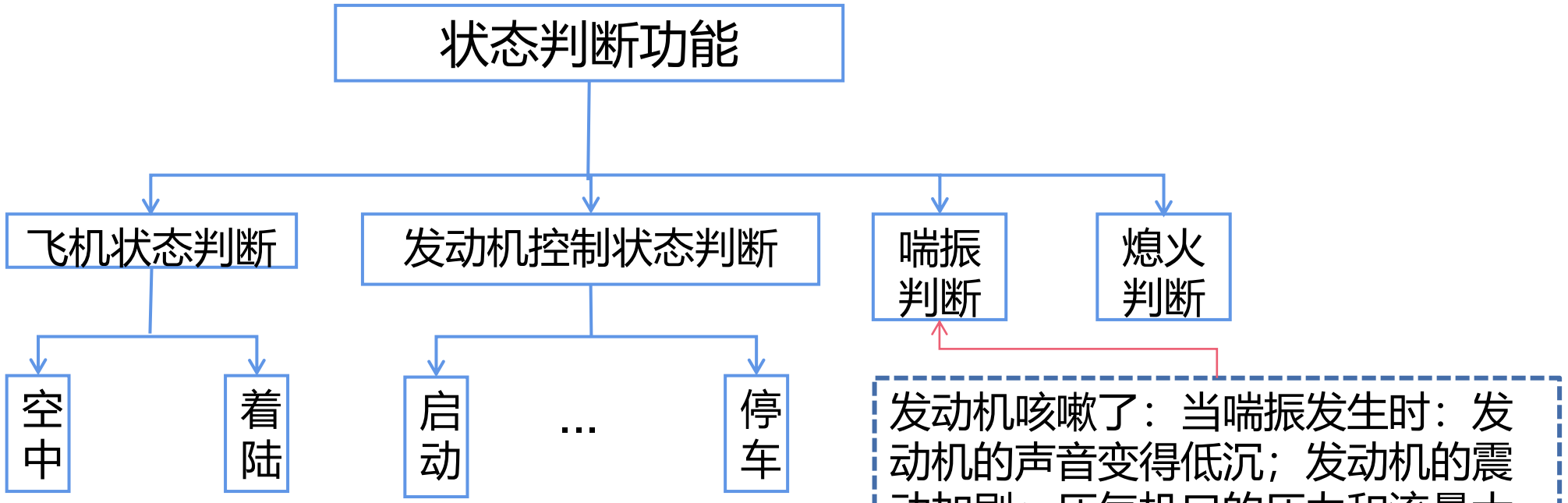


图5-8 状态判断功能图

2.2 软件可靠性分析--软件FTA

2）失效模式确定

状态判断是控制输出的基础，其中涉及的状态众多，判断条件复杂，本示例对状态判断中的熄火判断进行分析。

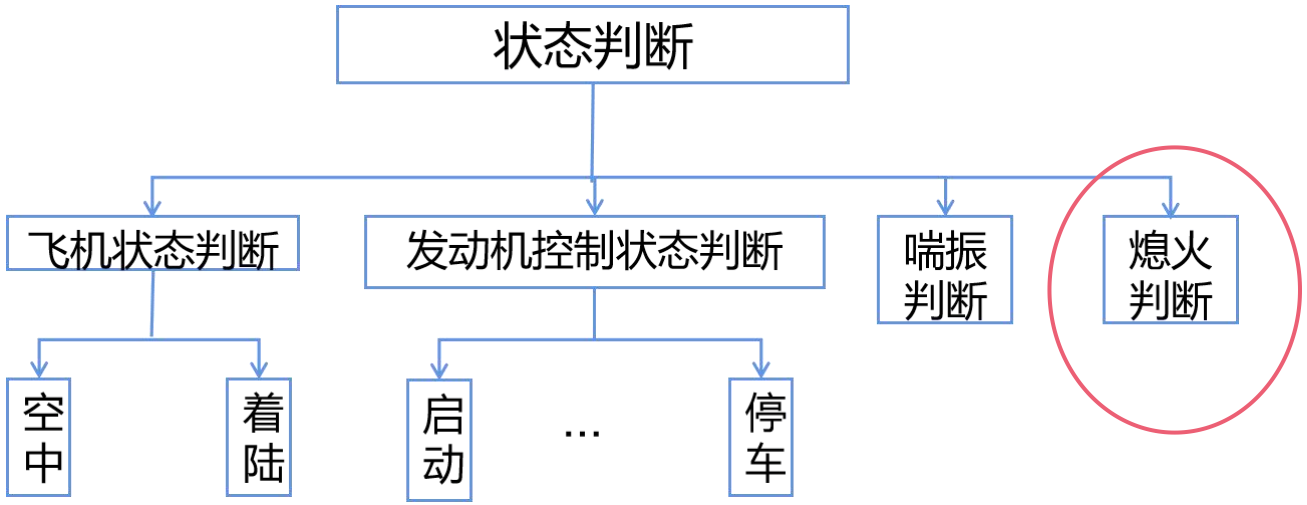


图5-8 状态判断功能图

熄火判断功能对采集而来的转速、温度等信号进行条件判断，若满足条件则将状态位置设置为熄火标注；否则不进行设置。

2.2 软件可靠性分析--软件FTA

2）失效模式确定

状态判断是控制输出的基础，其中涉及的状态众多，判断条件复杂，本示例对状态判断中的熄火判断进行分析。

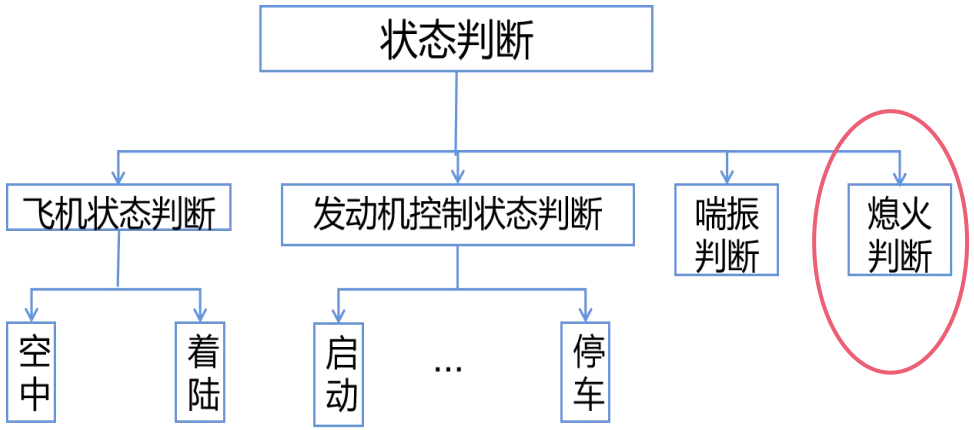


图5-8 状态判断功能图

经过人员分析与开发方讨论，确定以下失效模式：①错误地设置了熄火标识：当发动机实际并不处于熄火状态，但是软件采集得到信号判断后，依据判断条件却设置了发动机熄火标识。②没有执行设置熄火标识：当发动机实际处于熄火状态，但是软件采集得到信号判断，依据判断条件并没有设置发动机熄火标识。

2.2 软件可靠性分析--软件FTA

3）失效原因分析

针对每一个失效模式，分析其可能的原因。分析如下：

①“错误地设置了熄火标识”的原因分析：熄火判断中要考察转速和温度下降的速率，一个周期内有可能受到干扰，导致转速、温度下降率超过了规定值，如果只根据一个周期进行判断那么容易导致误判。

失效原因是未对转速、温度下降率进行过周期确认。

周期：同一事件两次发生间隔时间的最小值。

2.2 软件可靠性分析--软件FTA

3）失效原因分析

针对每一个失效模式，分析其可能的原因。分析如下：

②“没有执行设置熄火标识”的原因分析：当发动机实际处于熄火状态，但是发动机却判断为不处于熄火状态，得到了不符合实际的结果，分析其原因是熄火判断条件不够充分。

2.2 软件可靠性分析--软件FTA

4）失效影响分析

针对每一个失效模式，分别分析其对局部、高一层次，直至最终系统造成的影响及其严重程度。可以不用分析对高一层次的影响，而直接给出系统最终影响。严酷度类别定义如表5-16

严酷度类别	软件失效影响的程度
I	造成发动机停车或爆燃等，无法完成派遣任务
II	造成发动机在超温、超转、超压等边界条件下工作；或造成控制系统余度等级降低；或干扰飞行员的判断；或引起发动机喘振
III	造成控制软件计算精度损失，影响发动机性能
IV	仅对系统的易用性、维护性造成影响

2.2 软件可靠性分析--软件FTA

4）失效影响分析

下面分别介绍失效影响的分析过程：

①“错误地设置了熄火标识”的影响分析： 对于本层次来说给停车状态判断提供了错误信息；高一层次影响是熄火状态报告虚警等；而对于系统来说，造成飞行员对发动机状态误判。

②“没有执行设置熄火标识”的影响分析：局部影响是没有通知发动机停车状态判断；高一层次影响是发动机熄火状态下未及时执行补救措施；对于系统的最终影响是容易引起发动机爆燃。

2.2 软件可靠性分析--软件FTA

5）改进措施制定

根据以上失效模式、原因、影响以及严重程度的分析结果，和开发人员讨论后对每个失效模式制定相应的解决措施。

①“错误地设置了熄火标识”的改进措施制定：针对失效原因采取相应的措施，对每个周期进行转速、温度下降速率判断，并进行不少于1秒的时间确认。

2.2 软件可靠性分析--软件FTA

5）改进措施制定

根据以上失效模式、原因、影响以及严重程度的分析结果，和开发人员讨论后对每个失效模式制定相应的解决措施。

②“没有执行设置熄火标识”的改进措施制定：针对失效的原因采取相应的措施，可以通过试验和研究改进发动机熄火判断条件的充分性。

分析完失效原因、影响与改进措施后，填写软件FMEA表格，如表5-17所列。

2.2 软件可靠性分析--软件FTA

表5-17 某发动机控制软件系统级FMEA表

单元	失效模式	失效原因	失效影响			严酷度	改进措施
			局部影响	高一层次影响	最终影响		
熄火状态	错误地设置了熄火标识	未对转速、温度下降速率进行多周期确认	给停车状态判断提供错误信息，误认为已经熄火	熄火状态报告虚警	造成飞行员对发动机状态误判	II	每个周期进行转速、温度下降速率判断，并进行不少于1秒的时间确认
	没有执行设置熄火标识	熄火判断条件不充分	给停车状态判断错误信息，误认为没有熄火	发动机熄火状态下未及时执行补救措施	容易引起发动机爆燃	I	改进发动机熄火判断条件

第2.2节作业

1）总结系统级软件FMEA的定义以及分析步骤，并阐述每个步骤功能。

第二章 软件可靠性分析

2.1 软件可靠性

2.2 软件可靠性分析

2.3 软件可靠性评估

TrIG
华东师范大学软件学院可信智能团队
Trustworthy Intelligence Group

2.3 软件可靠性评估

国家标准GB11457中， 软件可靠性评估（Software Reliability Assessment）或软件可靠性评价（Software Reliability Evaluation）是指“确定现有系统或系统部件可靠性所达到的水平的过程”。国际标准IEEE Std 1633中， 软件可靠性评估被定义为“统计学技术在系统测试和运行期间收集的可观察失效数据上的应用，用于评价软件的可靠性”。

不难看出， 二者的定义均认为，软件可靠性评估是在获得了软件失效数据之后对软件可靠性水平的定量估计和评价。

2.3 软件可靠性评估

软件失效数据可以在下述两种情况下获得：

一是在测试阶段后期，通过软件可靠性测试（即按照软件的实际使用方式测试软件的一种方法），收集测试过程中的失效数据，对软件的可靠性水平进行估计，并能够对未来可能达到的可靠性水平进行预计；

二是在软件投入使用后，通过收集实际使用过程中软件的失效数据，对软件可靠性进行评估，并对未来软件可能达到的可靠性水平进行预计。

2.3 软件可靠性评估

软件可靠性评估结果，不仅可以给出实际的可靠性水平，也可以为下一代软件或同类型软件的可靠性定量要求的确定提供参考。

软件可靠性评估是软件可靠性工程中重要活动内容之一，具有及其重要的作用。

2.3 软件可靠性评估

软件可靠性评估，不能不得不提及软件可靠性评估模型，它是软件可靠性定量评估技术的基础。

软件失效具有随机性。软件在是t时刻发生的失效数是N(t)， 显然N(t)是一个随机数，且随时间t的变化而不同， 即{N(t), t>0}是一个随机过程。

问： 这个随机过程应该服从什么分布？

2.3 软件可靠性评估

问： 这个随机过程应该服从什么分布?

参照国际标准IEEE Std.1633，可将软件可靠性评估模型分为3中通用类别： 指数分布的非齐次泊松(Poisson)过程模型(Non-homogeneous Poisson process，NHPP)、非指数分布的NHPP模型和贝叶斯（Bayesian）模型。

2.3 软件可靠性评估

呈指数分布的NHPP模型

呈指数分布的NHHP模型使用随机过程和失效率函数方法。

失效率函数Z(t)是运行时间t的函数。

具有代表性的模型是Jelinski-Moranda模型， 1972年开发的可靠性模型，是最早建立的软件可靠性模型之一，曾用于麦克唐奈道拉斯海军工程中。现在大多数软件可靠性模型要么可认为是其变形或扩展，要么与其密切相关。该模型是软件可靠性研究领域分第一个里程碑。

2.3 软件可靠性评估

Jelinski-Moranda模型

（1） 假设与数据要求：

模型的基本假设如下：

① 程序中的固有错误数N_0是一个未知的常数；

② 程序中的各个错误是相互独立的，每个错误导致系统发生失效的可能性大致相同，各次失效间隔也是相互独立的；

③ 测试过程中检测到的错误都被排除，每次排错只排除一个错误，排错时间可以忽略不计，在排错过程中不引入新的错误；

2.3 软件可靠性评估

Jelinski-Moranda模型

（1） 假设与数据要求：

模型的基本假设如下：

④ 程序失效率在每个失效间隔时间内是常数，其数值正比于程序中残留的错误数，在第i个测试区间，其失效率函数为

$$Z(x_i)=k(N_0-i+1)$$

式中：k为比例常数， x_i为第i次失效间隔中以i-1次失效为起点的时间变量。

2.3 软件可靠性评估

Jelinski-Moranda模型

在第i个测试区间，其失效率函数为

$$Z(x_i)=k(N_0-i+1)$$

式中：k为比例常数， x_i为第i次失效间隔中以i-1次失效为起点的时间变量。

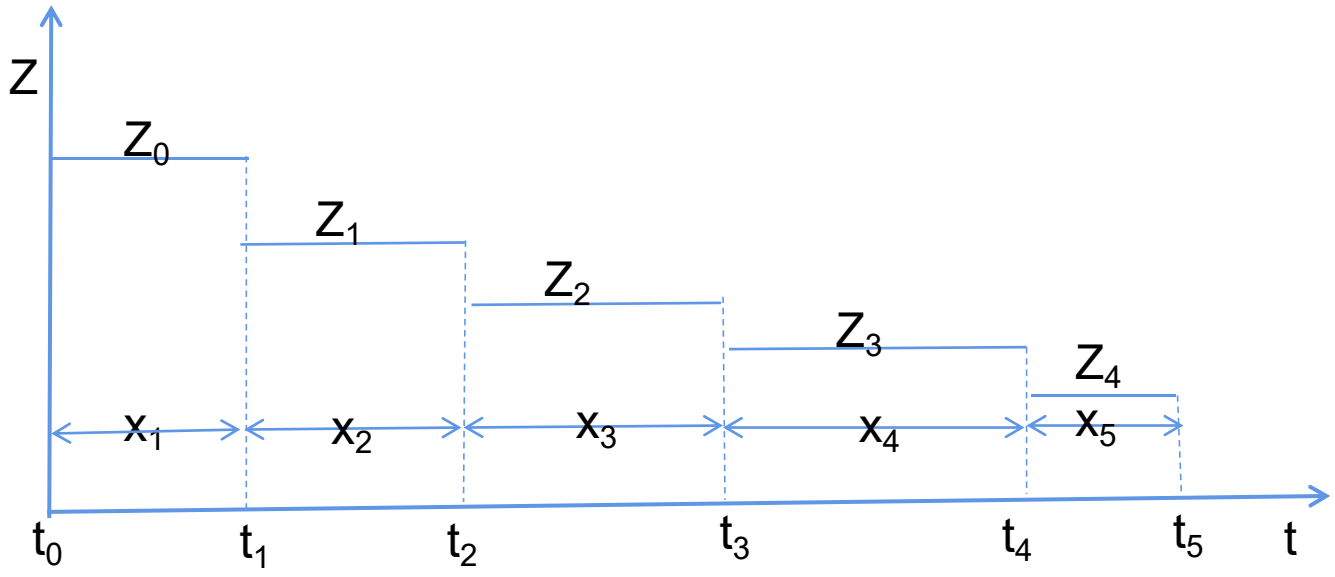


图7-1是J-M模型失效率随时间变化的曲线。图中ti是以时间0点为起点的第i次失效的累计发生时间， 即$x_i=t_i-t_{i-1}$，且i≥1， $t_0=0$。

2.3 软件可靠性评估

Jelinski-Moranda模型

（1） 假设与数据要求：

模型的基本假设如下：

⑤ 错误以相等的可能发生，且相互独立，错误检测率正比于当前程序中的错误数。

⑥ 软件的运行方式与预期的运用方式相似。

2.3 软件可靠性评估

Jelinski-Moranda模型

（2） 模型构造与参数估计：

在假设的基础上，运用可靠性工程学的基本理论，以第i-1次失效为起点的第i次失效发生的时间是一个随机变量，它服从以K（N_0-i+1）为参数的指数分布，其（失效）密度函数为

$$f(x_i)=\mathrm{k}(N_0-i+1)e^{-\mathrm{k}(N_0-i+1)x_i}$$

2.3 软件可靠性评估

Jelinski-Moranda模型

（2） 模型构造与参数估计：

它服从以k(N_0-i+1)为参数的指数分布，其（失效）密度函数为

$$f(x_i)=\mathrm{k}(N_0-i+1)e^{-\mathrm{k}(N_0-i+1)x_i}$$

其(失效)分布函数

$$F(x_i)=\int_0^{x_i} f(x_i)\mathrm{d}x_i=1-e^{-\mathrm{k}(N_0-i+1)x_i}$$

2.3 软件可靠性评估

Jelinski-Moranda模型

（2） 模型构造与参数估计：

其（失效）分布函数

$$F(x_i)=\int_0^{x_i} f(x_i)\mathrm{d}x_i = 1-e^{-\mathrm{k}(N_0-i+1)x_i}$$

其可靠性函数为

$$R(x_i)=1-F(x_i)=e^{-\mathrm{k}(N_0-i+1)x_i}$$

系数k和程序故有错误数N_0确定后，可靠性函数就确定了。

2.3 软件可靠性评估

Jelinski-Moranda模型　系数k和程序故有错误数N_0确定后，可靠性函数就确定了。

（2） 模型构造与参数估计：

假设总共发生了n个失效，则似然函数为

$$L(x_1,\cdots,x_n)=\prod_{i=1}^{n}f(x_i)=\prod_{i=1}^{n}\mathrm{k}(N_0-i+1)e^{-\mathrm{k}(N_0-i+1)x_i}$$

对上式两边取对数，得

$$\ln L(x_1,\cdots,x_n)=\sum_{i=1}^{n}\ln f(x_i)=\sum_{i=1}^{n}(\ln \mathrm{k}(N_0-i+1)-\mathrm{k}(N_0-i+1)x_i)$$

2.3 软件可靠性评估

Jelinski-Moranda模型 系数k和程序故有错误数N_0确定后，可靠性函数就确定了。

（2） 模型构造与参数估计：

假设总共发生了n个失效，则似然函数为 $L(x_1,\cdots,x_n)=\prod_{i=1}^{n} f(x_i)=\prod_{i=1}^{n} \mathrm{k}(N_0-i+1)e^{-\mathrm{k}(N_0-i+1)x_i}$

则模型参数的极大似然法估计值是以下方程组的解：

$$\begin{cases} \hat{\mathrm{k}}=\dfrac{\mathrm{n}}{\hat{N}\left(\sum_{i=1}^{n} x_i\right)-\sum_{i=1}^{n}(i-1)x_i} \\ \sum_{i=1}^{n}\dfrac{1}{\hat{N}-(i-1)}=\dfrac{n}{\hat{N}-(1/\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n}(i-1)x_i)} \end{cases}$$

其中 $\hat{\mathrm{k}}$ 和 $\hat{N}$是模型参数k和N_0的点估计值。
这是一个超越方程，可用数值计算法求解方程组，可以得到模型参数k和N_0的点估计值。

2.3 软件可靠性评估

Jelinski-Moranda模型 系数k和程序固有错误数N_0确定后，可靠性函数就确定了。

（3） 可靠性预计：

运用可靠性工程学的基本理论，利用上述推导出的估计值 $\hat{k}$ 和 $\hat{N}$，可相应地求得以下可靠性参数的估计值：

① 可靠度：

$$R_{n+1}(x) = R(x \mid t_n) = e^{-(\hat{N}-n)\hat{k}x}$$

② 不可靠度：

$$F_{n+1}(x) = 1 - e^{-\hat{k}(\hat{N}-n)x}$$

2.3 软件可靠性评估

Jelinski-Moranda模型 系数k和程序固有错误数N_0确定后，可靠性函数就确定了。

（3） 可靠性预计：

运用可靠性工程学的基本理论，利用上述推导出的估计值 $\hat{k}$ 和 $\hat{N}$，可相应地求得以下可靠性参数的估计值：

③失效密度：

$$f_{n+1}(x) = -\frac{dR_{n+1}(x)}{dx} = (\hat{N}-n)\hat{k}e^{-(\hat{N}-n)\hat{k}x}$$

2.3 软件可靠性评估

Jelinski-Moranda模型

系数k和程序固有错误数N_0确定后，可靠性函数就确定了。

（3） 可靠性预计：

运用可靠性工程学的基本理论，利用上述推导出的估计值 $\hat{k}$ 和 $\hat{N}$，可相应地求得以下可靠性参数的估计值：

④ 平均失效前时间MTTF：给定第n个软件失效发生时刻t_n，由失效独立性假设知，t_n时刻之后软件失效的MTTF为：

$$T_{TF_{n+1}} = E\{X_{n+1} \mid x_1, \cdots, x_n\} = \int_0^{\infty} R_{n+1}(x)\mathrm{d}x = \frac{1}{k(N_0 - n)}$$

则 $$\hat{T}_{TF_{n+1}} = \frac{1}{\hat{k}(\hat{N} - n)}$$

由此我们可以估计出软件可靠性的参数。他们都是依赖于前n次失效时刻，依据这n次失效时间，估计出了J-M模型的参数k和N0。n值越大估计的准确性就越高。

第二章作业

1）总结软件可靠性定义以及相关度量参数，字数不少于1000字

2）总结系统级软件FMEA的定义以及分析步骤，并阐述每个步骤功能。

3）假定某一软件在测试过程中，已发现了10次失效，其时间分别为

失效编号	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
失效时间	15	13	9	10	21	23	18	22	17	16

基于J-M模型，估计第11次失效发生时，软件可靠度、不可靠度、失效密度和失效前平均时间MTTF的值。

2.2 软件可靠性分析--软件FTA

2、软件故障树分析（Software Fault Tree Analysis, 软件FTA）、

2.2 软件可靠性分析--软件FEMA与FTA综合分析

1、软件失效模式和影响分析（Software Failure Tree Modes and Effects Analysis, 软件FMEA）、

2、软件故障树分析（Software Fault Tree Analysis, 软件FTA）、

3、将二者结合的软件FEMA与FTA综合分析，以及

4、事件树分析

2.2 软件可靠性分析--软件事件树分析

1、软件失效模式和影响分析（Software Failure Tree Modes and Effects Analysis, 软件FMEA）、

2、软件故障树分析（Software Fault Tree Analysis, 软件FTA）、

3、将二者结合的软件FEMA与FTA综合分析，以及

4、事件树分析

2.2 软件可靠性分析--软件事件树分析

1、软件失效模式和影响分析（Software Failure Tree Modes and Effects Analysis, 软件FMEA）、

2、软件故障树分析（Software Fault Tree Analysis, 软件FTA）、

3、将二者结合的软件FEMA与FTA综合分析，以及

4、事件树分析

第二章作业

1）总结软件可靠性定义以及相关度量参数，字数不少于1000字

2）总结系统级软件FMEA的定义以及分析步骤，并阐述每个步骤功能。

3）

华东师范大学软件学院可信智能团队

Trustworthy Intelligence Group