华东师范大学软件学院课程作业

课程名称：软件质量分析	年级：2023 级本科	姓名：张梓卫
作业主题：总结软件可靠性定义	学号：10235101526	作业日期：2024/9/26
指导老师：陈仪香	组号：	

目录

一 软件可靠性分析

1 定义

按照 GB/T11457-95-软件工程术语，软件可靠性的定义为：

（一：定量定义）在规定的条件（软件运行的软、硬环境、操作剖面：即软件运行的输入空间及其概率分布）下，在规定的时间内软件不引起系统失效的概率。

其中：

- 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。
- 输入空间：软件所有可能的输入值构成的空间。
- 操作剖面：系统使用条件的定义。

（二：定性定义）在规定的时间（执行时间、日历时间、时钟时间）周期内所述条件下程序执行所要求的功能的能力。
其中：

- 执行时间：一个程序所用的实际时间或 CPU 时间（激励软件发生失效）【软件可靠性模型是针对执行时间建立的】
- 日历时间：编年时间，包括计算机可能未运行的时间。
- 时钟时间：程序执行开始到程序执行完毕所经过的钟表时间，该时间包括了其他程序运行的时间。

二 软件失效机理

1 基础概念

发生的失效数越多，或者发生失效的时间间隔越短，软件越不可靠。

2 软件失效机理

2 .1 失效的原因总结

其中，软件失效分为四个可能的部分。

- 错误（开发者）：指开发人员在开发过程中出现的失误、疏忽和错误。
  - 遗漏或误解软件需求规格说明中的用户需求
  - 不正确的翻译或遗漏设计规格说明的需求
- 缺陷（程序固有）：代码中能引起一个或多个失效的错误的编码。
  - 软件缺陷是程序固有的
  - 软件文档中不正确的描述也称为缺陷
  - 不正确的功能需求、遗漏的性能需求
- 故障（运行时）：是由于软件缺陷在运行时引起并产生的错误状态
  - 不正确的数值等
  - 数据在传输过程中产生的偏差
- 失效（用户经历）：程序的运行偏离了需求，是动态运行的结果。（遇到缺陷时）。
  - 死机、错误的输出结果
  - 超出规定时间的响应

3 失效的随机性

存在随机性的原因：

- 程序执行的条件不可预测
- 程序的使用千差万别
- 缺陷或错误在程序中的位置未知

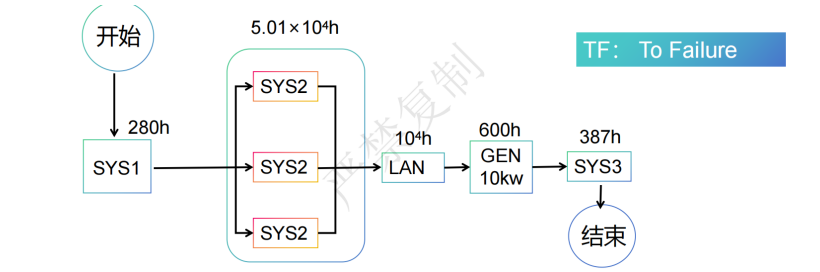
三 MTTF 计算公式

其中，TF 指的是 To Failure

1 当不考虑软件失效情况

$$\frac{1}{TFs} = \sum_{i=1}^{n} \frac{1}{TF_i}$$

Example:

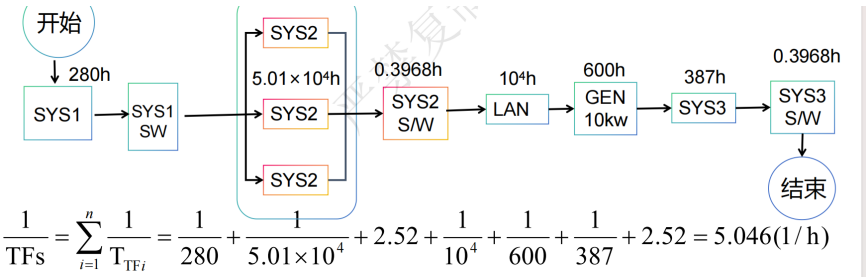


$$\frac{1}{\mathrm{TFs}} = \sum_{i=1}^{n} \frac{1}{\mathrm{T_{TF}}_i} = \frac{1}{280} + \frac{1}{5.01\times10^4} + \frac{1}{10^4} + \frac{1}{600} + \frac{1}{387} = 0.007942\,(1/\mathrm{h})$$

此时，$TFs = \frac{1}{0.007942} = 125.9h$

2 考虑软件失效情况

当考虑软件失效情况时（SYS2、SYS3 初始失效率为 2.52/h），则：



$$\frac{1}{\mathrm{TFs}} = \sum_{i=1}^{n} \frac{1}{\mathrm{T_{TF}}_i} = \frac{1}{280} + \frac{1}{5.01\times10^4} + 2.52 + \frac{1}{10^4} + \frac{1}{600} + \frac{1}{387} + 2.52 = 5.046\,(1/\mathrm{h})$$

此时，$MTTF = \frac{1}{\frac{1}{TFs}} = 11.9min.$

四 软件的失效

1 定义失效严重等级

应在软件需求说明中给出明确的项目特定的软件失效定义，因为软件可靠性是对失效行为特性的刻画，对于不同严重程度的失效用户常常有不同的要求。

Notice:

软件失效等级从对人员生命、成本、任务等方面考虑（分为 1~5 级，1 级最严重）：

失效严重等级	生命影响定义	成本影响/美元	系统能力影响	可靠性要求示例（跨网络交易）
1	危及生命或系统	>100,000	用户不能进行一项或多项关键操作	暂时的，有段损。一次跨网络的交易丢失导致了数据库损毁。失败率每 1000 次交易 1 次
2	对完成的任务有影响	10,000~100,000	用户不能进行一项或多项重要操作	暂时的，无段损。无法读取一张非损坏的卡的磁条数据。成功率 1/1000 次交易
3	可采取统计措施，影响极小，任务可达成	1,000~10,000	用户不能进行一项或多项操作，但有补救方法	永久的，无段损。系统对任何输入的卡都反应无效，软件必须更新来更正失效。失败率 1/1000 天
4	对需求或标准有轻微违反，运行中难以察觉	<1,000	一项或多项操作中的小缺陷	无
5	表面问题，将影响行为应列为注意或追踪，但不一定需要当前解决	无	无	无

五 度量参数

1 失效率与失效强度

1 .1 失效率

失效率的定义和硬件可靠性中瞬时失效率的定义完全一致的，基于寿命的观点给出的。它是一个条件概率密度。

失效率是指在 t 时刻尚未发生失效的条件下，在 t 时刻后单位时间内发生失效的概率。即：设 ξ 为发生失效的时间，$Z(t)$ 为失效率，则有

$$Z(t)=\lim_{\Delta t\to 0}\frac{P(t<\xi<t+\Delta t\mid \xi>t)}{\Delta t}=\lim_{\Delta t\to 0}\frac{P(t<\xi<t+\Delta t)}{P(\xi>t)\cdot\Delta t}=\lim_{\Delta t\to 0}\frac{R(t)-R(t+\Delta t)}{R(t)\cdot\Delta t}$$

因此：

$$Z(t)=-\frac{1}{R(t)}\cdot\frac{dR}{dt}=\frac{1}{R(t)}\cdot f(t)$$

1 .2 失效强度

失效强度则是基于随机过程定义的，是失效数均值的变化率。

假设软件在 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。设 $u(t)$ 为随机变量 $N(t)$ 的均值，即 $u(t)=E(N(t))$，则 t 时刻的失效强度 $\lambda(t)$ 定义为：

$$\lambda(t)=\frac{du(t)}{dt}$$

2 MTTF 与 MTBF

2 .1 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。

MTTF 用于不可修复产品

假设当前时间到下一次失效的时间为 ξ，ξ 具有累计算率密度函数 $F(t)=P(\xi\le t)$，即可靠度函数：

$$R(t)=1-F(t)=P(\xi>t)$$

则基于 MTTF 的软件可靠度 T_{TF} 为：

$$T_{TF} = \int_0^\infty R(t)dt$$

2 .2 平均失效间隔时间 MTBF

定义与**平均失效前时间 MTTF** 相近，只需将 ξ 转化为两次相邻失效时间间隔的均值即可。
MTBF 用于可修复产品

2 .3 修复软件失效对软硬件的不一致性

软件失效是可以修复的。但是，修复活动对失效特性的影响和硬件存在着很大的不同。

理论	实际
存在分析不可能得到的产品，如食品、电器	存在不能得到的文献，即实体档案都是可以得到的。
文献后续进行完全框架，则失效率不变，原 MTBF 不变	文献后续框架生态化，原 MTBF 变化
如果失效后不修而丢弃使用，修复对系统可靠性影响（与失效前之比），系统的失效率是变化的	失效后可以不修而丢弃，但与失效前相比，并不存在着避免修复。如果对原材料使用复杂且持续不变的话，材料失效率未必不变。这是因为回避不修，产品失效的可能与产品的使用频率有关，与产品的状况有关，而要满足一定的条件，确保回向导致失效的情况。

3 缺陷密度与故障密度

3 .1 缺陷密度（DD）

软件缺陷的基本度量，用于设定产品质量目标，支持软件可靠性模型预测潜藏的软件缺陷。
公式如下：

$$DD = \frac{D(\text{每个版本或模块规定严重性等级下的缺陷数})}{KSLOC(\text{代码的千行源代码数}(=\text{代码行数}/1000))}$$

3 .2 故障密度

按严重性分类将计算的故障密度与目标值比较来确定是否已经完成足够的测试。
公式如下：

$$FD = \frac{F(\text{导致严重性等级的失效的唯一故障数（相同故障记作一次）})}{KSLOC}$$

4 需求依从性

量反映软件需求分析工作的度量。根据该度量可以确定在需求分析阶段，软件需求规格说明中的需求不一致的比例，适用于**需求阶段**。
计算公式：
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。

公式：

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

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

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

注释：

- 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。
- $N_1 + N_2 + N_3$ 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。

5 度量参数的选择

- （1）对失效发生频率要求较低的系统，可靠性参数可选失效率或失效强度，如操作系统、电话交换系统软件等。
- （2）对在规定时间内能无失效工作要求比较高的系统，可选可靠度作为软件可靠性参数，如火力控制系统软件等。
- （3）对使用比较稳定的软件，可选平均失效前时间/平均失效间隔时间作为软件可靠性参数，如通用软件包等。