第6章

软件品质、IT的风险及管理

李俊杰
计算机与软件学院

引言

- “也许最后的答案是生活中没有东西能保证，但解决问题的第一步是认识到问题”--汤姆.福雷斯特（Tom Forester）和佩里.莫里森（Perry Morrison）

- 一次事故足以动摇前面千万次的成功计算留在人脑中的认识

- 需要培养：IT使用者的风险意识，IT设计者的告知义务

提纲

6.1 基本术语

6.2 复杂的软件

6.3 IT风险分析和控制方法

6.4 IT使用者的风险意识

6.5 IT设计者的风险意识

提纲

6.1 基本术语

1. 基于计算机系统的风险

- **计算机系统的安全包括**
  - 计算机系统本身的可靠性
  - 由计算机系统处理、传输和存储的信息的保密性、完整性和可用性

- **计算机系统的组成**
  - 软件、硬件、技术文档、通信线路
  - 人（最主要的组成元素）

6.1 基本术语

2. 风险管理

- 定义（英国企业信息安全标准RS7799）：在可接受的成本下，标识、控制和尽量减少（或消除）可能影响信息系统的安全风险的过程

3. 可靠性问题

- 计算机系统的可靠性是在规定的条件下和规定的时间内计算机系统能正确运行的概率，一般用平均无故障时间（Mean Time to Failure, MTTF）来度量
- 可靠性研究的四论域：失效（物理域）、故障（逻辑域）、差错（信息域）、失败（用户域）

6.1 基本术语

4. 电子文件的可靠性

- 电子文件的可靠性是指其内容的真实与完整，或者指电子文件的可信度与真实性。

5. 完整性

- 完整性被公认为是“一种未受损的状态”（一致性）。

6. 网络的完整性与安全性

- 局域网上面的数字资源无论是传输还是存储都存在着很高的风险。

6.1 基本术语

7. 数字资源的长久保存

- **数字资源的长久保存是指长久保存数字资源的信息内容和功能性的可存取性的一系列策略和手段，是实现信息共享的一个关键问题。**
  - 潜在的因素：系统自身的故障、出错或者升级、人为破坏等

8. 信息技术保护

- **编码化与密码化、资格检查、内存保护、外存保护**

9. 防火墙

- **数据包过滤器（Packet Filter）和应用层次防火墙（Application-level Firewall）**

提纲

- 6.1 基本术语
- 6.2 复杂的软件
  - 6.2.1 软件复杂性的概念
  - 6.2.2 软件复杂性的产生与后果
  - 6.2.3 大型软件开发的过程
  - 6.2.4 软件测试及其局限性
- 6.3 IT风险分析和控制方法
- 6.4 IT使用者的风险意识
- 6.5 IT设计者的风险意识

6.2.1 软件复杂性的概念

1. 定义

- **软件复杂性是指理解和处理软件的难易程度，包括程序复杂性和文档复杂性**
  - 软件复杂性主要体现在程序的复杂性中
  - 程序的复杂性关系到软件开发费用的多少、开发周期的长短、软件内部潜伏错误的多少
- **例：**
  - 一套航天飞机所用的软件：2560万行，22096人开发一年，成本达12亿美元
  - 花旗银行的自动柜员系统软件：78万行
  - 伦敦希斯罗机场使用的计算机管制系统：100万行，1600人开发一年，完善（500人年）

6.2.1 软件复杂性的概念

2. 程序复杂性度量

(1) 软件度量

- 规模，即总共的指令数，或源程序行数
- 难度，通常由程序中出现的操作数的数目所决定的量来表示
- 结构，通常用与程序结构有关的度量来表示
- 智能度，即算法的难易程度

- **例：毕业设计的代码量和难度等要求**

(2) 程序复杂性度量原则

- 程序理解的难度；纠错、维护程序的难度；向他人解释程序的难度；按指定方法修改程序的难度；根据设计文件编写程序的工作量；执行程序时需要资源的程度。

6.2.2 软件复杂性的产生与后果

1. 软件复杂性的产生原因

(1) 追求完美，想发明“银弹”

(2) 系统使用过程中逐步变得复杂（例：Java编程语言）

(3) 软件工程存在着“陷阱”（例：看似简单明了的东西，却有可能变成一个落后进度、超出预算、存在大量缺陷的怪物）

6.2.2 软件复杂性的产生与后果

1. 复杂性与实用性

- **复杂系统通常都会失败，不过失败后通常还会衍生出一个相对简单而易用的系统，例如：**
  - 为简化MULTICS而诞生的UNIX已被广泛使用了三十多年
  - 针对J2EE复杂性而出现的轻量级解决方案，Spring, Hibernate等
  - SMTP和POP3虽有安全隐患等，但比X.400协议简单很多，而X.400已很少见到
- **例：学术研究中的代码**

2. 软件复杂性带来的危害

- **增加开发难度，降低开发效率**
- **增加维护难度**
- **增加学习使用难度**

6.2.3 大型软件开发的过程

1. 大型软件开发概述

- **工程化的开发控制成为软件系统成功的保证**
  - 出色的编程能力和开发技巧+严格的软件工程思想

2. 软件可靠性设计

- **软件可靠性：在规定的条件下和规定的时间内，软件成功地完成规定功能的能力或不引起系统故障的能力**
  - 软件开发过程中所使用的软件开发学
  - 与验证方法有关
  - 使用的程序设计语言、软件运行环境条件、操作人员的素质等

6.2.3 大型软件开发的过程

3. 软件容错和避错

- 在一定程度上对自身故障的作用具有屏蔽能力，就称此软件为具有容错功能的软件，即容错软件（Fault-Tolerant Software, FTS）
- 例：航空飞行器、空间飞行器
- 例：AlphaGo中的自学习能力

4. 可测试性设计

- 可测试性设计是系统设计人员在设计计算机系统的同时就充分考虑到测试的要求，即用故障诊断的理论、方法和技术去指导系统设计，实现功能设计与测试设计的统一。
- 核心思想：提高系统的可控制性和可观测性

6.2.3 大型软件开发的过程

5. 大型软件的开发方法

- 结构化程序设计技术
- 面向对象技术
- 净室（Cleanroom）软件技术
- 螺旋式增量技术

6. 大型软件开发的成功案例--子弹列车控制系统

- 日本的新干线子弹列车（Bullet Train）开始运营后，多年来从未发生过人员死亡的事故，因此号称为全球最安全的高速铁路之一。其中很大程度上是在于软件开发过程中借鉴了先进科学的开发技术。软件开发商日立公司反复测试软件模块，并把程序员按流水线的形式组织起来严格监督。

6.2.4 软件测试及其局限性

1. 软件测试的概念

- 两类方法：证明程序是正确的；“证伪主义”
- 例：互联网公司中测试人员的工作

2. 软件测试局限之1：**不能证明**软件中还有没有其他错误和缺陷
3. 软件测试局限之2：从硬件领域移植过来的测试技术，**有效性不如硬件**…
4. 软件测试局限之3：如何有效提高机构的测试能力和水平**没有提供指导**…
5. 软件测试局限之4：设计者如果告诉用户软件有缺陷，**用户则不用了**…
6. 软件测试局限之5：任何测试都具有**不确定性**

提纲

- 6.1 基本术语
- 6.2 复杂的软件
- 6.3 IT风险分析和控制方法
  - 6.3.1 IT风险和可信计算的概念
  - 6.3.2 IT风险的管理过程
  - 6.3.3 信息系统风险评估
  - 6.3.4 项目管理
- 6.4 IT使用者的风险意识
- 6.5 IT设计者的风险意识

6.3.1 IT风险和可信计算的概念

1. 信息系统的安全隐患

(1) 关键组件失效

(2) 自然灾害

- 例：互联网公司的异地容灾备份

2. IT风险的主要类型

(1) 完整性风险：数据未经授权使用或不完整或不准确而造成的风险

(2) 存取风险：由于系统、数据或信息存取不当而导致的风险

(3) 获得性风险：影响数据或信息可获得性的风险

(4) 体系结构风险：信息体系结构规划不合理或未能与业务结构实现调配所带来的风险

(5) 其他相关风险：其他影响企业业务活动的技术性风险

6.3.1 IT风险和可信计算的概念

3. 可信计算（Trusted Computing）

(1) 可信计算的定义

- 定义（美国国防部）：可以违反安全策略的系统，也就是说“一个因为你没有选择而必须信任的系统”
- 可信计算平台是通过**硬件与软件技术**的结合，防范不同类型“不速之客”，**保证远程计算是可信的**

(2) 可信计算的组织

- 可信计算组织（TCPA）、微软公司的Palladium、英特尔的LaGrande等

6.3.2 IT风险的管理过程

1. 确定风险

2. 风险分析

3. 风险规划：制定风险解决方案，包括缓解方案、触发方案和应急方案

4. 风险跟踪：追踪风险的变化，撰写和提交风险状态报告，为后续的决策和行动提供信息支持

5. 风险控制

6.3.3 信息系统风险评估

1. 评估方法的类型

(1) 定性评估

(2) 半定量评估

(3) 定量评估

2. 信息系统风险评估的发展

- 我国信息系统风险评估工作目前还处于**起步阶段**，没有形成一套成形的专业规范，缺少一支能够全面开展信息系统风险评估的人才队伍。

6.3.4 项目管理

1. 软件开发标准
2. 项目管理：启动、计划、执行、控制、收尾
3. IT监理
4. 信息系统审计

- 定义（美国信息系统审计权威Ron Weber）：收集并评估证据以决定一个计算机系统（信息系统）是否有效做到保护资产、维护数据完整、完成组织目标，同时最经济地使用资源
  - 可用性、保密性、完整性
- 信息系统审计在国外已经发展成熟，而国内还处于起步阶段

提纲

- 6.1 基本术语
- 6.2 复杂的软件
- 6.3 IT风险分析和控制方法
- 6.4 IT使用者的风险意识
  - 6.4.1 IT使用者的风险意识的概念
  - 6.4.2 对风险认识的程度
  - 6.4.3 风险意识的培养
- 6.5 IT设计者的风险意识

6.4.1 IT使用者的风险意识的概念

- 认识计算机系统的风险性：研究、思考计算机在使用过程中有无风险、风险有多大、风险何时可能发生、风险发生的条件是什么、与所得的相比值不值、化解风险的方式方法有哪些等问题

- 规避计算机的风险性：采取切实有效的手段方法去化解风险、防范风险、规避风险，减少不必要的损失，把风险降到最低，提高工作绩效

6.4.2 对风险认识的程度

- **2009年“全球信息安全调查”的结果**
  - 24%的中国企业过去一年没有进行过风险评估
  - 44%的中国企业只是由IT部门的普通员工进行风险评估

- **从总体情况来看，虽然国内IT使用者的风险意识已经开始逐步建立，但是IT风险知识普及程度和自我风险意识没有得到深刻认识**

6.4.3 风险意识的培养

1. 风险意识的培养要从密码开始

- 例：密码保存问题

2. 不要安装非正式的无线访问接入点

3. 不要安装未授权的软件

4. 小心邮件接收

5. 授权用户的滥用（数据库滥用）

提纲

- 6.1 基本术语
- 6.2 复杂的软件
- 6.3 IT风险分析和控制方法
- 6.4 IT使用者的风险意识
- 6.5 IT设计者的风险意识
  - 6.5.1 何谓IT设计者的风险意识
  - 6.5.2 告知IT使用者一个真实的计算机系统
  - 6.5.3 IT灾难的警示

6.5.1 何谓IT设计者的风险意识

- 作为设计开发软件产品、直接为顾客进行软件服务的IT设计者有着**巨大的压力**，在开发各种软件时，更要注意提高软件的**安全性**
- IT设计者的风险意识：对漏洞要有一个**深刻的认识**

6.5.2 告知IT使用者一个真实的计算机系统

1. 二进制

2. 系统模拟

3. 不可靠的计算机系统

- 例：2003年8月14日美国东部时间下午4时20分，以纽约为中心的美国东北部和加拿大部分地区发生大面积停电事故，直到8月15日下午才恢复供电 … 著名的安全机构（SecurityFocus）的数据表明，停电事故是由**软件错误**所导致的。

6.5.3 IT灾难的警示

1. 系统灾难事件

(1) 交通工具方面

(2) 科学研究领域

(3) 其他潜在危险

2. Intel奔腾芯片产品之争

小结

- 软件的复杂性
- 软件的重要性
- 风险意识