

软件测试基础知识

知识点1：软件测试概念



内容

- ◉ 软件软件测试的目的
- ◉ 软件测试的缺陷案例
- ◉ 软件测试的概念

软件的问题

- ◉ 软件项目经常
 - 无法按期完成，
 - 超出经费的预算，
 - 软件质量难以控制。
- ◉ 开发人员及开发过程之间
 - 管理不规范，约定不严格，文档书写不完整，软件可维护性不好。
- ◉ 缺少严密有效的质量检验手段
 - 交付给用户的软件质量差，运行中出现许多问题，带来严重后果。
- ◉ 系统更新换代难度大
- ◉ 软件开发的产业模式已经不能适应信息时代。

软件缺陷案例- 来自Wired News

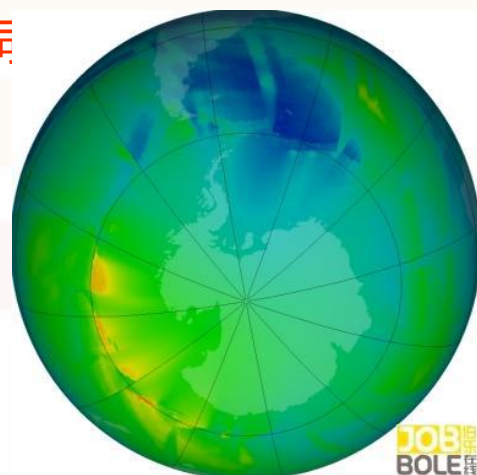
- 一触即发的第三次世界大战
 - 1980年，北美防空联合司令部曾报告称美国遭受导弹袭击。后来证实，这是反馈系统的电路故障问题，但反馈系统软件没有考虑故障问题引发的误报。
 - 1983年，苏联卫星报告有美国导弹入侵，但主管官员的直觉告诉他这是误报。后来事实证明的确是误报。
- 幸亏这些误报没有激活“核按钮”。在上述两个案例中，如果对方真的发起反击，核战争将全面爆发，后果不堪设想。



软件缺陷案例

“漏网”的臭氧层空洞

- 南极洲上方的臭氧层空洞一直存在但长期未被发现，这是为什么？
- 1978年，NASA启动臭氧层测绘的计划。在设计之时，用于该计划的数据分析软件忽略了和预测值有很大差距的数据。
- 直到1985年，才发现南极洲上方的臭氧层空洞，但不是NASA发现的（是英国科学家先发现的）。
- 直到NASA重新检测它们的数据，才发现这一错误。在修正错误后，NASA证实南极臭氧层的确有个很大的空洞



软件缺陷案例

- 消失在太空-美国航天局火星登陆系统的预设单位Bug
 - 在制造其火星气候轨道探测器时，一个NASA的工程小组使用的是英制单位，而不是预定的公制单位。这会造成探测器的推进器无法正常工作。正是因为这个 Bug，1999年探测器从距离火星表面130英尺的高度垂直坠毁。此项工程成本耗费3.27亿美元，这还不包括损失的时间（该探测器从发射到抵达火星将近一年时间。）



软件缺陷案例

◉ 阿丽亚娜5型火箭发射失败-代码重用Bug

- 1996年6月4日，阿丽亚娜5型运载火箭的首航，但因软件引发的问题导致火箭在发射39秒后偏轨，自我摧毁。
- 后来查明事故原因是：代码重用。在Ariane 5中重用4的代码，算法程序中触发了一个bug：存在于将64位浮点数转换为16位带符号整数的程序中。直接诱发了溢出条件，最终导致了航天计算机的崩溃。此次事故损失3.7亿美元。



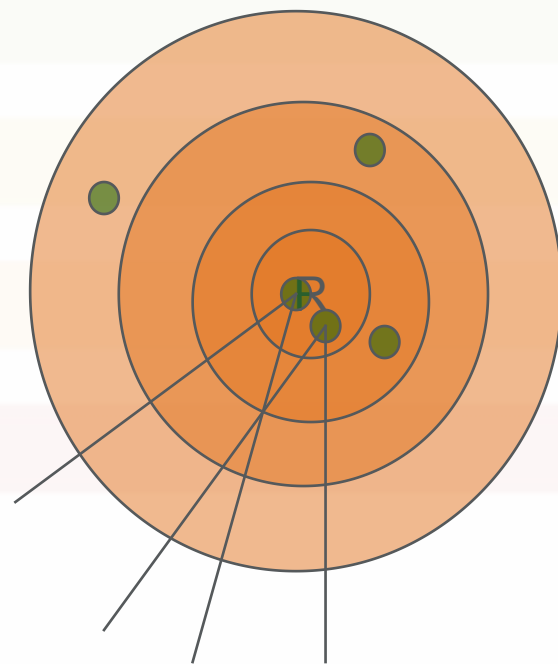
软件缺陷案例

◉ 英特尔奔腾芯片缺陷- Inter CPU的浮点除法Bug

- 在1994年，一个硅片上的错误导致Intel高性能奔腾芯片在一段范围内计算浮点指数除法时发生错误。例如 $4195835.0/3145727.0$ 产生的是1.33374而不是1.33382，产生了0.006偏差。
- 尽管该bug仅仅影响了几个用户，然而他却成了整个公众的噩梦。估计流通中的三百万到五百万的芯片存在着这样的缺陷，这就是英特尔（Intel）奔腾（Pentum）CPU芯片所带来的一个浮点触发缺陷，英特尔为该bug付出了4亿多美元的代价。

软件问题的原因

- ◎ 是软件缺陷的积累和放大效应
 - 失之毫厘，谬以千里
 - 成本急剧增加，质量大幅下降
 - 开发过程的混乱，效率很底



为何要进行软件测试

- ◉ 在开发软件过程中，我们使用了保证软件质量的方法分析、设计和实现软件，但难免还会在工作中犯错误
 - 在软件产品中就会隐藏着许多错误和缺陷
 - 规模大、复杂性高的软件更是如此
 - 这些错误有些是致命性的，如不排除，就会导致生命与财产的重大损失
- ◉ 软件测试是确保软件质量的重要一环，测试是手段，质量是目的,属于软件工程领域

软件测试的定义

- ◉ Hetzel：评价一个程序和系统的特性或能力，并确认它是否达到预期的结果，检查是否满足规定的需求。
- ◉ Myers：测试是为发现错误而执行程序的过程。
- ◉ 通用：是对软件需求分析、设计、编码的最终复查的一系列过程，是软件质量保证的关键步骤。

软件测试的定义

- ◉ 1983年IEEE提出的软件工程术语中的定义是：
 - 使用人工或自动的手段来运行或测定某个软件系统的过程，其目的在于检验它是否满足规定的需求或弄清预期结果与实际结果之间的差别
- ◉ 这个定义明确指出：
 - 软件测试的目的是检验软件系统是否满足需求。
 - 测试再也不是一个一次性的而且只是开发后期的活动，而是与整个开发流程融合成一体。
- ◉ 软件测试已成为一个专业，需要运用专门的方法和手段，需要专门人才和专家来承担



谢 谢