缺陷管理（上）
——缺陷的属性

Ron Patton关于软件缺陷的定义

- 软件测试员认为软件难以理解、不易使用、运行速度缓慢，或者最终用户认为不好。
- 软件未达到需求规格说明书中指明的功能。
- 软件出现了需求规格说明书中指明不会出现的错。
- 软件功能超出需求规格说明书中指明的范围。
- 软件未达到需求规格说明书中虽未指出但应达到的目标。

缺陷管理概述

- 缺陷管理：是在软件生命周期中识别和管理缺陷的过程（从缺陷的识别到缺陷的解决关闭），确保缺陷被跟踪管理而不丢失。
- 一般的，需要跟踪管理工具来帮助进行缺陷的全流程管理。

缺陷管理概述

- 缺陷的属性
- 缺陷报告
- 缺陷跟踪和管理

缺陷的属性

- 可重现性
- 严重性
- 优先级

1 可重现性

- 即缺陷应可以满足不止一次地触发出现
- 只有可以重现的缺陷，才有可能进行定位，以及修复，无法重现的缺陷是无法修复的

1 可重现性

- 所有缺陷都可以重现吗？
- 在如下情况下，缺陷可能是无法重现的
  - 受到误差累积的影响，只有误差累积到一定程度才能出现
  - 与特殊日期相关的
  - 与程序或功能运行次数相关的缺陷
  - 缺陷可能导致极其严重的后果而无法恢复

1 可重现性

- 只要发现程序执行中出错，意味着程序中一定有缺陷
- 但若不能让缺陷重现，则缺陷报告无法顺利提交给程序员
- 如何确保缺陷重现？如何确信缺陷可以重现？

1 可重现性

- 如何确保缺陷重现？如何确信缺陷可以重现？
  - 测试前对相关环境和数据备份
  - 测试过程中详细记录下每一个测试步骤和系统的响应
  - 使用不同的测试数据或操作步骤，或改变测试环境，看能否触发缺陷
  - 重复相同的测试，至少3次

2 严重性

- 即缺陷对系统造成的破坏力
  - 严重的：因重要功能丧失或致命错误而造成系统崩溃，重要数据丢失或破坏而可能引起用户财产损失或法律纠纷，甚至可能危及人身安全等
  - 一般的：不影响系统基本使用，能满足商业要求，但用户不常用的功能实现未达到预期效果，可能导致用户使用不方便等
  - 次要的：不会影响软件主要功能的使用，产品及其属性仍然可以使用，而且一般可以轻易处理的缺陷

3 优先级

- 即项目组对缺陷的处理优先级
  - 高：必须立即修复
  - 中 (一般)：缺陷需正常排队等待修复，但在产品发布之前必须修复
  - 低：在时间允许的时候进行修复

3 优先级

- 严重性高的的缺陷通常指定高优先级
- 非常严重的缺陷一定将指定为最高的处理优先级吗?

4 可修复性

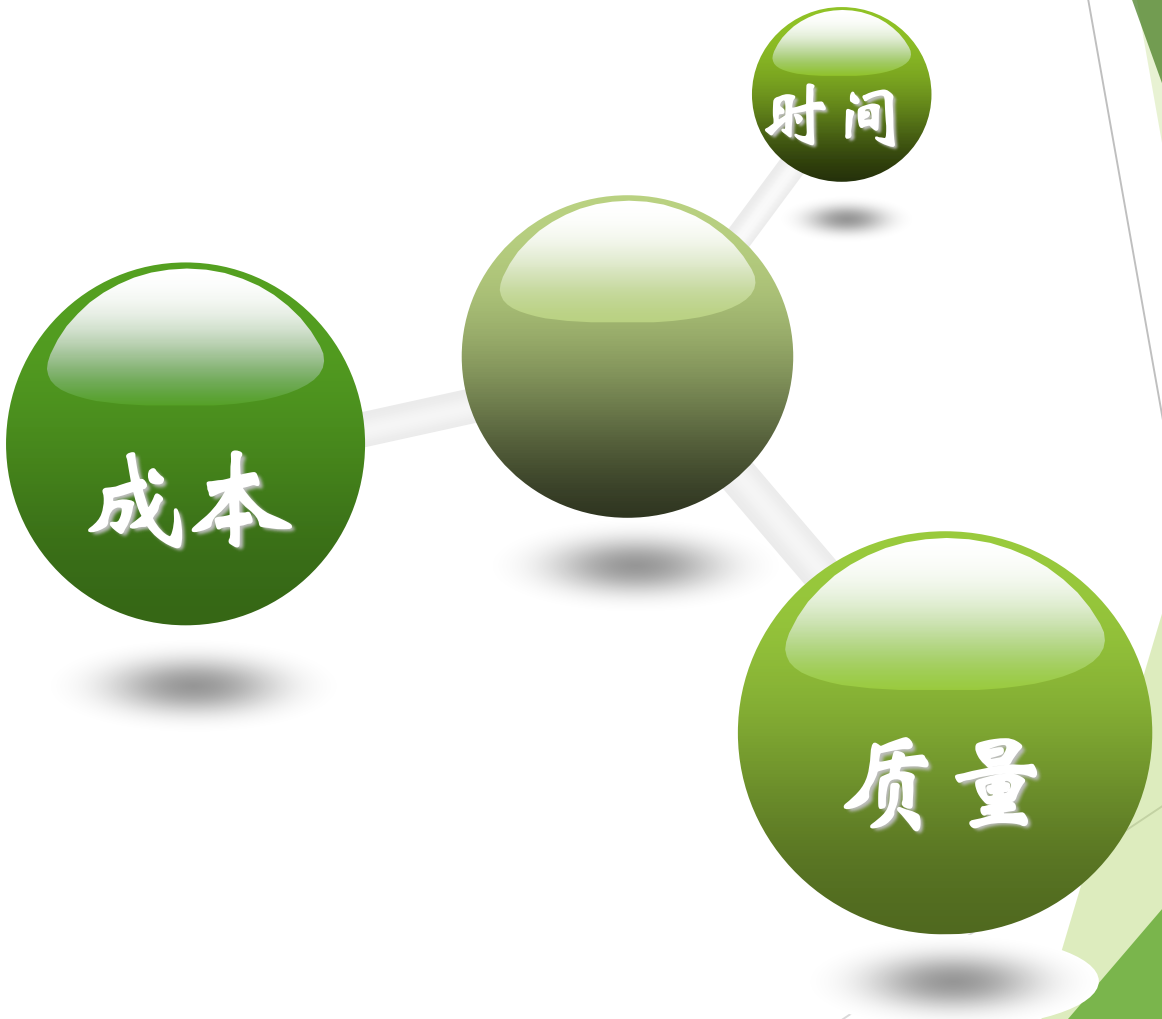
- 即缺陷在软件产品发布之前是否可以得到修复
- 所有缺陷在产品发布之前都可以得到修复吗？

4 可修复性

- 若时间允许，且缺陷修复涉及人员不多，修改的模块不多，则缺陷可能得到修复
- 否则，缺陷很可能不修复

4 可修复性

- 缺陷不被认可
- 缺陷无需修复
- 缺陷没时间修复
- 修复风险太高



缺陷的属性

- 需在报告中明确体现
  - 可重现性
  - 严重性
  - 优先级
- 不在报告中体现
  - 可修复性