

软件缺陷常常又被称为 Bug。所谓软件缺陷就是指计算机软件或者程序中存在的某种破坏正常运行能力的问题、错误或者隐藏的功能缺陷。

Bug 的存在会导致软件产品在某种程度上不能满足用户的需要。

在 IEEE 中对 Bug 有一个标准的定义：

1. 从产品内部看，是指软件产品开发或维护过程中存在的错误、毛病等各种问题。
2. 从产品外部看，是指系统所需要实现的某种功能的失效或违背。

缺陷种类

缺陷可以分为不同的种类：

1. 遗漏：指规定或预期的需求未体现在产品中。
2. 错误：指需求是明确的，在实现阶段未将规格说明正确实现。
3. 冗余：指需求规格说明未涉及的需求被实现了。
4. 不满意：除了上面 3 种情况外，用户对产品的实现不满意也称为缺陷。

缺陷的等级划分

在不同的企业对软件缺陷等级的划分大同小异，大致可分为五个等级：

1. 致命：指造成系统或应用程序死机、崩溃、非法退出等，会造成用户数据丢失或被破坏，功能设计与需求严重不符的问题。
2. 严重：指功能和特性没有实现，导致模块功能失效或异常退出，还有程序接口错误或者数据流错误等问题。
3. 一般：指主要功能丧失，提示信息不太正确，用户界面设计太差以及删除未提示等问题。
4. 提示：指对功能几乎没有影响，产品及属性仍可使用的的问题。
5. 建议：测试人员提出的建议、质疑等问题。

提交缺陷

不同的公司因为缺陷管理的流程不一样，可能有不同的缺陷报告模版。但是一个完整的缺陷报告通常应该包含以下内容：

1. 编号：用数字进行唯一标识缺陷，通常是在缺陷管理工具中新建 Bug 时会自动生成。
2. 状态：通常描述当前缺陷的状态，比如修复、延期等。
3. 标题：通常用一句比较简洁的话来概括 Bug，通过描述可以初步推测 Bug 原因，来提高处理的效率。
4. 类型：主要为了进一步描述缺陷产生的原因，比如功能错误、接口错误、数据库错误等。
5. 所属版本：描述当前 Bug 所在的测试版本，便于后期回归时注意测试版本。
6. 所属模块：描述 Bug 所在的业务模块，便于后期统计缺陷的分布情况，利于在进行回归测试的方法及测试策略的改进。
7. 严重级别：指 Bug 的严重程度。通常不同的严重程度给软件带来的后果、风险的影响都不一样，开发人员处理的优先级也不同。
8. 处理优先级：由开发人员根据 Bug 的严重级别来确定处理的优先级。

9. 发现人：Bug 的提交者。
10. 发现日期：一般在提交 Bug 时，由 Bug 管理工具自动生成，便于后续进行缺陷的跟踪。
11. 复现概率：指 Bug 重现的概率，便于开发定位分析。一般包括必现、偶现等。
12. 指定处理人员：根据 Bug 的类型指定处理人。通常指定具体的开发人员，如果是需求错误则需要指定产品或需求分析人员，便于后期进行跟踪。
13. 详细描述：详细描述缺陷引发的原因以及复现步骤。需要包含测试环境、前提条件、测试数据、复现步骤、预期结果、实际结果等内容。
14. 附件：为了加深描述可以添加一些附件信息，如截图、录屏、错误的日志信息等。