# 软件工程的可靠性和可维护性

## 计算机软件工程的可维护性分析

随着软件工程以及开发的复杂程度和系统完善程度的加深，逐步形成了相对较为科学的方式，对软件的可靠性、可维护性以及可懂性予以保障，同时生产率也逐步提高，效率提高的同时必然降低了成本。

在软件的整个生命周期中，软件技术的各个阶段所采用的技术以及管理都具有先进性，每个阶段在完成后都需要针对性的进行严格的审查、管理，只有在上一-阶段工作合格之后才能进行下个环节的开发，这就确保了软件开发的质量，尤其在可维护性.上得到了有效提高。

1维护性概述

软件的周期主要有两个重要的阶段，-一个是开发阶段另一一个则是运行阶段，实际上.系统是存在很多的错误和漏洞的，软件的开发是一项大投资， 在生产率提高、成本降低的同时，人们希望获得高品质的软件，从而快速的实现软件的应用和扩张。因此就需要在软件投入使用运行后，由软件人员进行进一一步的修改,针对系统中存在的错误和漏洞进行修正，这就是我们常说的软件维护。

常见的软件维护有三种:首先是纠正维护，主要针对软件中的错误进行修改;其次为完善维护，主要用于对软件的性能和功能予以扩大和提升;最后则是适应维护，该类工作则是为了提高软件对外部环境的适应性。

2一般维护方法

2.1模块化提升以及质量技术提升

在开发软件时，对软件可维护性提高最有效的方式之一便是对开发质量的提高，这不但会对成本予以降低，同时也能够提高技术的有效性。功能模央的独立性是该种方式的优势，若某一模块需要更改，那么只需要对该模块进行修正，而不是整体性修改，不会对其他模块造成影响;若程序需要增添新的功能，那么只需要对该功能进行新模块的添加即可;而在测试重复以及程序时更加方便，很容易发现序列错误，并对其进行修正，从而提高了运行效率。

2.2品质的创建以及优先级的创建

维护性对于程序来说应当是能够理解、修改、移植、使用并且可靠性以及效率较高的。.上述要求的实现，需要付出相当大的代价，并且即便是付出了也不一 定具有可行性。-一些软件在质量特性.上具有一定的互补性，例如可修改性、可理解性以及可理解和测试性等。但是在其他的特质上同样具有一定的矛盾性， 例如可移植性同效率之间或者效率同可变化性之间。所以，软件在满足可维护性要求的同时应当根据程序的使用和环境的要求而做出适当的调整。

3可维护性加强措施

在进行软件的维护过程中，会遇到诸多的问题，例如软件开发过程中人员变动较大，开发商的变动性也较大;由于文档资料的缺失，使得开发过程中对他人的开发体系无法全面

的掌握;程序问题或者稳当文件的不恰当，软件结构出现问题，难以理解，无法对不恰当的地方予以修改，或者即便修改也容易出现问题。

### 计算机软件工程可靠性分析

1. 软件开发人员应具有设计出高质量软件的思想意识  
   软件的实现不能仅以完成软件需求方提出的软件功能需求和性能需求为最终目标,还要在满足需求方要求的基础上,对于软件的易用性、健壮性、可靠性、安全性等都要有较完善的设计和实现，这样软件开发交付使用后出现缺陷的概率大大降低，也许就不再需要维护。这就要求开发人员具有设计高质量软件的意识。
2. 要具备“第一次做正确”的“零缺陷”思想  
   IBM公司给华为公司做过一次软件过程水平的诊断报告中提到：“(华为公司)没有时间在第一次就做对的事,有时间反复做同样的事”。这可以说是很多中国企业都存在的问题。由于任务节点的“后墙不倒”,导致在需求分析清楚,设计还不充分的情况下就匆匆进行编码实现,导致后期不断通过联试、测试来修复问题。如果从系统层面就树立“第一次做正确”的思想,软件开发前期做充分的分析和设计,那么一定会减少后期的缺陷,完成任务的节点也会得到保证。
3. “平衡”的艺术  
   现在军用软件非常重视和强调软件可靠性。但是,并不是说为了提高软件的可靠性就可以不顾任务节点,不考虑投入的人力资源。好的软件应该是在时间、成本、质量之间获得平衡。在确定软件质量能够达到交付要求的时间基础上,将软件的风险降至最低。  
   平衡的艺术不仅体现在实现软件可靠性这个层面,对于软件测试和评审的粒度也同样适用。软件测试和评审到什么程度,都应考虑平衡投入和取得的成效。  
   根据软件的可靠性、安全性要求对软件划分相应的等级,找出关键的软件部件(子系统、配置、部件、单元、功能等),进行重点管理,降低风险,这也是在保证一定可靠性、安全性要求的前提下,与投入成本、任务周期达成平衡的一种非常现实的做法。
4. 规范开发对于软件可靠性同样重要  
   在导致软件失效的事件链因素当中,有一些是由于软件使用错误造成的：包括缺少清晰的、良好结构化的用户文档,使得用户对于如何使用软件没有很好理解；软件设计的用户接口太复杂,使用过程中非常容易出错:缺乏必要的培训和技术支持,用户对软件不熟悉不了解;用户的教育或系统背景知识不足;用户对操作环境或产品用途了 解不足:产品规格说明和用户使用环境不ー致:使用了不适当的硬件配置等。这些问题都是可以通过实施一个严格有序的管理过程来避免或减少。
5. 建立缺陷管理系统  
   要做好软件可靠性,必须要建立一个缺陷管理系统。通过这样的系统,收集软件的缺陷描述、问题报告、原因分析、解决措施等信息,形成关键件判定和故障历史,从而产生软件的设计准则；形成使用、故障、维修信总报 告,从而产生软件可靠性评估报告;形成故障统计分析报告,从而产生软件可靠性增长计划。
6. 成立软件可靠性小组以帮助高质量软件实现  
   为了更好地做好软件可靠性工作,有必要成立一个软件可靠性小组。这个小组应包括可靠性验证和可靠性测试的人员,即做好软件的评审和测试。同时他们也要配合软件质量保证人员,做好软件过程的规范化管理。