

# 一种面向开发过程的软件可靠性预测方法

赵波<sup>1</sup>,王大翊<sup>2</sup>,荣霓<sup>3</sup>,董威<sup>4</sup>

(1. 国防科技大学 继教学院,湖南 长沙 410073;

2. 海军装备技术研究所,北京 100072;

3. 75753 部队,广东 广州 510600;

4. 国防科技大学 计算机学院,湖南 长沙 410073)

**摘要:**在进行软件开发时,该如何回答这样的问题,“当前开发的软件可靠性是什么?”这个对于用户来说是非常重要的信息,却难以表述。以此为方向,研究了一种面向开发过程的软件可靠性预测方法,结合软件开发各个阶段的特点,通过合理地采用软件可靠性早期预测模型,实现对开发中软件可靠性的预测。实践应用表明,该方法紧密结合软件开发的实际,反映了软件可靠性与开发人员的水平、开发规范的标准之间的联系,有助于采取必要的措施提高其以后的可靠性,从而也验证了该方法的工程实用性。

**关键词:**软件开发过程;软件可靠性工程;可靠性模型;可靠性预测

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2011)09-0014-05

## A Reliability Prediction Method for Process of Software Development

ZHAO Bo<sup>1</sup>, WANG Da-yi<sup>2</sup>, RONG Ni<sup>3</sup>, DONG Wei<sup>4</sup>

(1. College of Continuing Education, National University of Defense Technology, Changsha 410073, China;

2. Naval Institute of Technology Equipment, Beijing 100072, China;

3. Troops 75753 PLA, Guangzhou 510600, China;

4. Computer College, National University of Defense Technology, Changsha 410073, China)

**Abstract:** During software development, how to answer the question like “what is reliability about the software which is developing right now” is crucial for users whereas difficult to express. Based on this direction explored a reliability prediction method for the process of the software development, and through reasonable use of the software reliability prediction model, combining with the characteristics of every stages of software development, achieved the prediction of software reliability during the development. The practical application shows that the method closely connects with the actual software development, reflects the links between the software reliability, the level of developers and standards of norm during development, and it is helpful to take the necessary measures to improve the reliability in its future, which also proves the engineering practicability of the method.

**Key words:** software development; software reliability engineering; reliability model; reliability prediction

## 0 引言

随着软件产业的深入发展,软件可靠性工程作为软件开发过程中的有效手段而得以迅速的发展。软件可靠性工程定义为定量地按照用户对软件可靠性的需求,对基于软件系统的操作进行研究的行为<sup>[1]</sup>。软件可靠性工程主要利用软件可靠性模型,对软件进行可靠性预测和评估。软件可靠性模型作为实施可靠性评

估预测的有效手段,得到了广泛的探索。然而,对已确立的上百个模型的应用经验表明,没有一个普遍适用的模型能对所有的软件产品以及软件产品生命周期的各个阶段过程都能做出最佳的可靠性预测<sup>[2]</sup>。尤其是在软件开发过程中,对软件的早期可靠性预测具有非常现实的实践意义<sup>[3]</sup>。目前,面向软件开发过程的早期可靠性预测模型相对比较匮乏,而且预测精度普遍较低,操作性差,难以指导实际的软件开发<sup>[4]</sup>。

文中结合软件开发过程中各个阶段的特点,以软件开发过程中的可靠性预测为目标,探索了一种面向开发过程的软件可靠性早期预测方法,并设计和实现了面向开发过程的软件可靠性预测工具。其特点是:

收稿日期:2011-02-23;修回日期:2011-05-26

基金项目:国家自然科学基金资助项目(60970035,91018013)

作者简介:赵波(1981-),男,山东莱州人,硕士,研究方向为软件可靠性;董威,教授,硕士生导师,研究方向为高可信软件开发。

紧密结合软件开发的实际,能够直接地反映出软件可靠性与开发人员的水平、开发规范标准的联系,有助于开发人员采取必要的措施提高软件以后的可靠性。

## 1 面向开发过程的软件可靠性预测方法的过程分析

软件的开发过程包括软件需求分析、软件设计(总体设计和详细设计)编码实现和单元测试以及集成测试阶段<sup>[5]</sup>。这是一个复杂的过程,各种开发文档、软件需求、软件体系结构等大量的相关数据与软件可靠性之间必然存在联系,而传统方法在进行软件可靠性预测时,并未充分考虑。面向开发过程的软件可靠性预测的方法是在分析相关可靠性预测模型的基础上,在软件分析、设计、编码等尚未形成可运行代码的阶段对软件的可靠性指标进行度量,以此来评定软件开发过程的合理性。

下面结合一般软件开发过程来详细叙述在开发过程对软件进行可靠性预测的方法。

### 1.1 需求分析阶段的可靠性预测

需求分析阶段的特点是处于软件开发的早期,这一阶段的主要任务是对项目开发计划阶段所确定的软件项目的目的、条件、运行环境、软件产品要求、人员分工职责及进度,以及预计软件项目的可靠性进行任务区分,确定主要任务和次要任务,并以此来设计软件的基本流程、软件结构、模块的定义和输入输出数据、接口和数据结构等,同时还应对项目开发计划阶段做出的可靠性预计,做进一步细化,形成可靠性需求,建立具体的可靠性指标。

在需求分析,由于软件系统只停留在概念层,并且需求分析阶段涉及的内容主要以定性分析为主,所以这一阶段的预测,应该以对各种因素的一种综合考虑为优。

对需求分析阶段的软件可靠性预测过程为:

(1)根据软件系统应用类型确定初步的平均错误密度<sup>[6]</sup>。

查询相关的系统文档,确定开发软件系统的应用类型,对照表1,选择相适应的软件系统应用类型,得出初步的平均错误密度。所得结果为由软件应用类型所决定的错误密度的初步估计值 $A$ 。

(2)依据软件开发环境对基准错误密度进行修正<sup>[7]</sup>。

随着软件开发的不断深入,在对整个软件系统有了比较详细了解的基础上,进一步考虑软件的开发环境,分析软件开发过程中若干影响软件可靠性的特征量,通过分析软件开发项目的相关文档,得出整体的开发模式,并对照表2得到一个修正因子 $D$ ,最终依据公

式(1),得出需求分析阶段的错误密度预测值。

$$\lambda = A * D \quad (1)$$

表1 系统应用类型决定的错误密度对照表

| 系统应用类型   | 平均错误密度 A(错误/代码行) |
|----------|------------------|
| 机载系统     | 0.0128           |
| 战略信息指挥系统 | 0.0092           |
| 战术指挥系统   | 0.0078           |
| 过程控制系统   | 0.0018           |
| 生产系统     | 0.0085           |
| 开发工具     | 0.0123           |

表2 修正因子对照表

| 开发环境  | 说明                                            | D    |
|-------|-----------------------------------------------|------|
| 系统模式  | 软件的开发团队是所开发软件的整个组织中的一部分                       | 0.76 |
| 半分离模式 | 软件的开发团队有开发目标应用的经验,但是并不直接使用软件                  | 1.0  |
| 嵌入模式  | 软件的开发团队有软件开发方面的经验,但是并不熟悉所开发的软件针对的特定领域的知识,系统复杂 | 1.3  |

此过程是在系统设计的层上,对软件的可靠性作出的初步预测,预测的依据是诸多系统开发中积累的历史数据。

### 1.2 软件设计阶段的可靠性预测

软件设计阶段的主要任务是对需求分析阶段定义的功能模块逐步细化,确立系统结构,形成若干可实现的模块,并说明硬件与软件模块之间的接口及它们与外部环境的接口,详细描述各模块的输入、输出及处理过程。以此来产生一个作为软件构造基础的软件内部结构的描述。因此,软件的设计阶段对可靠性的影响,主要体现在:设计的规范性、模块化的思想、设计的复杂性、可溯性等。

对软件设计阶段的可靠性预测的具体预测过程为:

(1)异常管理 SA 评估。

异常管理 SA,表示软件系统对系统错误或是其他异常情况的反应处理能力,对这一特征量的试题主要从以下几个方面进行考虑:

- a)错误状态控制;
- b)输入数据检查;
- c)计算故障识别和恢复;
- d)硬件错误识别和恢复;
- e)设备错误识别和恢复;
- f)通信错误识别和恢复。

系统失效率会随着异常控制能力的提高而降低,而异常控制能力则体现在以上几个方面,比如错误的的数据,可能由硬件导致的失效,系统过载。对于在关键领域使用的软件,软件的设计和测试要保证不出现导

致任务失败的异常状态。

评估方法:

对照表 3, 对软件相关需求文档和设计文档进行评估, 根据软件系统是否符合所提出的要求, 做出“是”或“否”的判断, 并统计选择“是”的个数  $N$ 。根据下式得出修正因子  $SA$ 。

$$SA = \begin{cases} 0.9 (N \text{ 是 } < 3) \\ 1.0 (N \text{ 是 } = 3) \\ 1.1 (N \text{ 是 } > 3) \end{cases} \quad (2)$$

表 3 异常管理 SA 评估表

| 序号 | 要求描述                                |
|----|-------------------------------------|
| 1  | 是否有一个控制识别错误的标准, 使得所有的错误都能传递到相应的处理函数 |
| 2  | 是否对输入数据进行错误包容设计                     |
| 3  | 是否有从计算失效中恢复的需求                      |
| 4  | 是否有从 I/O 错误中恢复的需求                   |
| 5  | 是否有从所有和其他系统通信失效中恢复的需求               |
| 6  | 是否有提供替代路径策略的需求                      |
| 7  | 是否有当异常发生时保证数据完整性的需求                 |
| 8  | 是否有复制关键数据的需求                        |
| 9  | 是否有需求从异常中恢复系统各部分的连接, 以保证计算过程不被打断    |
| 10 | 是否有需求对传输误差进行控制                      |

(2) 可溯性 ST 评估。

可溯性是预测软件可靠性的一个特征量, 其目的是考察模型与需求及实现的功能间的联系是否密切。如果需求和所建立的模型之间的关系越紧密, 模型满足需求的程度就越高, 其可靠性也就越高。

特征值 ST 的评估也贯穿阶段需求分析、概要设计、详细设计三个阶段。根据可溯性的含义, 要在后一个开发阶段体现前一个开发阶段的设计和思想, 使得在前一个阶段的设计思想和要求, 在后一个阶段都有相应的设计与之对应。如对软件设计阶段的可溯性评估, 就是将需求分析的内容细化为一条条需求条款, 在概要设计阶段, 根据列出的需求条款, 逐一评审是否有相应的概要设计条目对需求进行实现。同样的, 在详细设计阶段, 也将详细设计的每一具体条目与概要设计条目比较, 评估其可塑性。

评估方法:

对照表 4, 分析软件开发各阶段的文档, 如果下述所列要求都满足, 则  $ST=1$ , 否则  $ST=1.1$ 。

(3) 质量评估 SQ。

质量评估的目标是对软件需求和设计中涉及的一些影响软件可靠性的因素进行综合评价, 得出特征量 SQ 的值, 对软件的可靠性进行进一步的修正。

度量方法:

对照表 5 质量评估表中列出的各项, 考查软件需

求分析文档, 评定软件系统是否符合所提出的要求, 做出“是”或“否”的判断, 并统计选择“否”的个数  $N$ , 根据公式(3)得出质量评估 SQ。

$$SQ = \begin{cases} 1.1 (N \text{ 否 } > 6) \\ 1.0 (N \text{ 否 } \leq 6) \end{cases} \quad (3)$$

表 4 可溯性评估表

| 序号 | 要求描述                            |
|----|---------------------------------|
| 1  | 软件需求分析所涉及到的各个功能都有相对应的系统或是子系统的设计 |
| 2  | 软件顶层的系统功能划分在概要设计中都有相应的模块与之对应    |
| 3  | 软件系统所有功能单元是否满足了上层设计的全部需求        |
| 4  | 从顶层设计到底层设计的分解过程是否通过图表清楚表示       |

表 5 质量评估表

| 序号 | 要求描述                       |
|----|----------------------------|
| 1  | 所有的处理函数对应的合法输入值是否有定量的准确性要求 |
| 2  | 所有的处理函数对应的输出是否有定量的准确性要求    |
| 3  | 所有的处理函数对应的常数是否有定量的准确性要求    |
| 4  | 现有的数学库程序是否能满足所要求的精度        |
| 5  | 是否要求工作的系统有独立的电源            |
| 6  | 所有的输入, 处理, 输出是否有明确的定义      |
| 7  | 所有的定义函数是否均被引用              |
| 8  | 是否所有对相同数据的引用都是用一个唯一的名字     |
| 9  | 是否有具体的标准来规范定义和使用全局变量       |
| 10 | 是否有具体的标准来规范设计中的数据表示        |
| 11 | 是否有具体的标准来规范所有数据的命名         |
| 12 | 是否有具体的标准来规范软件单元间的调用        |
| 13 | 是否所有被引用的函数都经过明确定义          |

(4) 通过上述过程所得到的异常管理 SA、可溯性 ST、质量评估 SQ 的结果, 依据公式(4)计算出软件设计阶段对软件失效率预测的修正因子  $D$ 。并结合在软件需求分析阶段得到的结果, 得出软件设计阶段的软件错误密度预测值。

$$D = SA * ST * SQ \quad (4)$$

(5) 结合在软件需求分析阶段得到的结果, 得出软件设计阶段的软件错误密度预测值。

$$\lambda(\text{设计}) = \lambda(\text{需求}) * D \quad (5)$$

### 1.3 编码实现阶段的可靠性预测

编码实现阶段的主要任务是把详细的设计转换成程序设计语言<sup>[8]</sup>。编码实现在软件开发阶段中是一个时间相对较长, 伴随着代码调试和模块单元测试不断寻找错误和改正错误, 随着时间的推移, 软件的错误密度不断减小、软件的可靠性不断上升的迭代过程。因

此,在这一过程中,开发者更关注编码实现过程中软件内部错误数量的变化情况<sup>[9]</sup>。

编码实现阶段的可靠性预测方法:

结合此阶段的特点,采用 Putnam 模型<sup>[10]</sup>对软件编码实现阶段的可靠性进行预测。Putnam 模型是由 IBM 和通用公司联合研究的软件可靠性早期预测模型,它把软件开发过程划分为以下十个里程碑,并记录到达每个里程碑所花费的自然月数。通过预测软件系统内部错误在软件编码实现过程中的变化情况,一方面可以有效地控制开发时间,严格控制进度,另一方面可以很好地指导软件的开发,如:预测给出了每个月份软件错误密度的变化情况,还可以给单元测试提供数据依据。

Putnam 模型应用标准瑞利分布对软件的错误密度进行预测,其计算公式为:

$$f(t) = \left(\frac{N}{t_d^2}\right)t \cdot \exp\left(-\frac{3t^2}{t_d^2}\right) \quad (6)$$

其中, $N$ 是固有的内在错误总数(每千行源代码所含的错误数); $t_d$ 是实现安装后初步使用所需的时间(自然月); $t$ 是进行可靠性预测时的累计开发时间; $f(t)$ 则是随着软件开发和测试的不断推进, $t$ 时间内发现的错误占内在错误总数的百分比的预测值。

编码实现阶段的预测要用到软件设计阶段的预测结果,即软件的错误密度,以此来计算出错误总数。在计算出软件已排除的错误占内在错误的百分比,即软件的可靠度之后,再计算出当前软件的错误密度,作为下一阶段软件预测的参考值。

#### 1.4 系统测试阶段的可靠性预测

系统测试的主要任务是根据软件系统开发各阶段的规格说明和程序内部结构而精确设计一批测试用例<sup>[11]</sup>(即输入数据和预期的输出结果),并利用这些测试用例去运行程序,以发现错误的过程,对于整个软件系统而言,软件可靠性测试是软件可靠性保证过程中重要的内容。早期测试可以降低生产高质量软件的成本,经过软件可靠性测试,用户更加关注的是软件在运行过程中所表现出来的可靠性特性。

系统测试阶段的可靠性预测方法:

结合此阶段的特点,采用 Musa 执行时间模型<sup>[12]</sup>对这一阶段的失效率进行预测。Musa 执行时间模型用于在软件系统测试开始时预测软件的失效率。又由于在编码实现阶段得到了软件错误密度的预测值,此时源代码行数已确定,可得出 Musa 模型计算所需要的参数初始错误总数。

$$\lambda_0 = k * \omega_0 * p \quad (7)$$

其中:通过对历史经验数据的分析, $k$ 取  $4.2 \times 10^{-7}$ ;  $p$  是单位时间内软件执行的代码行数; $\omega_0$  是程序

中初始错误数量的估计值,一般由经验获得,也可以通过公式估算。计算  $p$  的公式如下:

$$p = r / \text{SLOC} / \text{ER} \quad (8)$$

其中, $r$ 是常量,表示指令执行的平均速率,其值由生产商或标准程序决定;SLOC,源代码行数,是指软件程序代码中除去注释代码的代码总数,其中不包括再生代码;ER,扩展比率,是由程序设计语言决定的表示软件代码扩展比率的常数。

计算  $\omega_0$  的公式如下:

$$\omega_0 = N \times B \quad (9)$$

其中, $N$ 为内在错误总数的估计值,凭借软件开发的历史数据和经验估计; $B$ 是错误到失效的转换比率,具体的说,软件系统中的内在错误并不一定会引起软件系统运行时的失效,也就是在软件的内在错误和失效之间存在一个转换比率。

## 2 面向开发过程的软件可靠性预测工具

基于上述对面向软件开发过程的软件可靠性预测方法过程的描述,设计并实现了基于软件开发过程可靠性预测工具,该工具以用户对软件开发过程中特征和测试时收集的故障数据为基本输入,以软件开发各个阶段的失效率为输出,完整地描述软件开发各个阶段中软件所表现出的可靠性。

工具的过程分析及输出见表 6。

表 6 工具的过程分析及输出

| 开发阶段 | 评估内容                                     | 工具输入                                    | 工具输出                                                                            |
|------|------------------------------------------|-----------------------------------------|---------------------------------------------------------------------------------|
| 需求分析 | 软件开发计划<br>软件需求格式说明书                      | 系统应用类型<br>开发环境                          | 需求分析阶段的<br>基准错误密度                                                               |
| 软件设计 | 软件需求文档<br>软件程序结构文档<br>软件数据结构文档<br>软件过程文档 | 异常管理 SA<br>可溯性 ST<br>质量评估 SQ            | 软件设计阶段预<br>测软件错误密度                                                              |
| 代码实现 | 软件开发里程碑表<br>软件设计文档                       | 累计开发时间<br>安装后初步使用<br>时间<br>软件的源代码<br>行数 | 当月发现的错误<br>占内在错误总数<br>的百分比<br>已被发现和排除<br>的错误占内在错<br>误总数的百分比<br>代码实现阶段软<br>件错误密度 |
| 系统测试 | 编程语言<br>软件的规模                            | 软件程序代码<br>中代码总数                         | 系统测试开始时<br>预测软件失效率                                                              |

## 3 应用

为了确保本研究的工程实用性,选择了航天三院的某型号综合控制软件的开发作为背景,以该软件的全数字仿真环境作为软件可靠性预测的模拟环境,进行该方法的效能性实验。

该软件的开发与传统的软件开发相比,存在一些特殊要求,一些传统的软件可靠性模型并不适用,例如对于弹上软件,真实运行过程中的失效数据难以获取等。通过实际应用的效果,证明该方法在对指导软件开发的过程和质量保证计划的实施,确保软件系统高可靠性要求的时效方面,有良好的效果。

#### 4 结束语

文中主要研究了一种在软件开发过程中的不同阶段采用可靠性预测技术进行软件的可靠性预测的方法,并开发了一套面向开发过程的软件可靠性预测工具。通过实践应用,验证了方法的合理性和科学性。但此方法多基于历史数据的收集和分析,因而下一步的研究重点是收集和分析工程经验和历史数据,提取更精确的评估参数,以提高方法的预测精度和方法的适用性。

#### 参考文献:

- [1] 国家标准 GB 6583.1. 质量管理和质量保证术语[S]. 国家标准局,1986.
- [2] Lyu M F. Handbook of software reliability engineering[M]. USA: McGraw-Hill Companies, 1996.
- [3] 齐治昌. 软件工程[M]. 北京: 高等教育出版社, 2004.
- [4] 董威, 王戟, 毛晓光, 等. 软件可靠性工程框架和评估

系统实现[J]. 计算机工程与应用, 2000, 36(10): 71-74.

- [5] ANSI/AIAA R-013. 推荐的软件可靠性实践[S]. 1993.
- [6] Friedman M A, Tran P Y, Goddard P L. Reliability Techniques for Combined Hardware and Software Systems[R]. Rome Laboratory, Air Force Systems Command, Griffiss Air Force Base, 1992.
- [7] McCall J A, Randell W, Dunham J, et al. Software Reliability, Measurement, and Testing Guidebook for Software Reliability Measurement and Testing[R]. Rome Laboratory, Air Force Systems Command, Griffiss Air Force Base, 1992.
- [8] MIL-HDBK-338B. Military Handbook Electronic Reliability Design Handbook[S]. USA: Department of Defense, 1998.
- [9] Sommerville I. Software Engineering[M]. England: Addison-Wesley Publishers Ltd, 1984.
- [10] Condon E, Cukier M. Applying Software Reliability Models on Security Incidents[C]//18th IEEE International Symposium on Software Reliability Engineering. Toronto: IEEE Computer Society, 2007: 150-168.
- [11] Goseva-Popstojanova K, Trivedi K. Architecture-based approach to reliability assessment of software systems[J]. Performance Evaluation, 2001, 45(2-3): 179-204.
- [12] Hamlet D, Mason D, Woit D. Theory of software reliability based on components[C]// In: Proc. of the 3rd Int'l. Workshop on Component-Based Software Engineering. Toronto: IEEE Computer Society, 2001: 361-370.

(上接第 13 页)

迭代次数,引入的风险评估机制有效地避免了碰撞的发生。

#### 6 结束语

文中提出了一种并行的基于评估和蚁群分工合作的并行移动机器人实施导航方法。该方法利用并行机制加快蚁群的搜索速度,增加机器人搜索的实时性。在保证搜索快速性的同时引入了评估机制,保证搜索的安全性。并在每个处理单元的蚁群中引入分工合作的概念优化蚁群路径。

实验结果表明:该算法具有较好的搜索不碰撞最短路径的能力,能解决机器人在复杂环境中的导航问题。

#### 参考文献:

- [1] 段海滨. 蚁群算法原理及其应用[M]. 北京: 科学出版社, 2006.
- [2] 毕军, 付梦印, 张宇河. 一种改进的蚁群算法求解最短路径问题[J]. 计算机工程与应用, 2003, 39(3): 107-109.
- [3] 曾碧, 杨宜民. 动态环境下基于蚁群算法的实时路径规划方法[J]. 计算机应用研究, 2010, 27(3): 860-863.

- [4] 曾碧. 移动机器人的动态环境建模与路径规划研究[D]. 广州: 广东工业大学, 2010.
- [5] 凯文, 李春葆, 秦前清. 基于蚁群算法的最短路径搜索方法研究[J]. 公路交通科技, 2006, 23(3): 128-134.
- [6] Dorigo M, Maniezzo V, Colomi A. Ant system: Optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man and Cybernetics-part B, 1996, 26(1): 29-41.
- [7] 景兴建, 王越超, 谈大龙. 基于人工协调场的多移动机器人实时协调避碰规划[J]. 控制理论与应用, 2004, 21(5): 757-764.
- [8] Kennedy J, Ebethart R. Particle Swarm Optimization[C]// Proceeding of 1995 IEEE International Conference on Neural Network. [s. l.]: IEEE, 1995: 1942-1948.
- [9] 朱庆保. 全局未知环境下多机器人运动蚂蚁导航算法[J]. 软件学报, 2006, 17(9): 1890-1898.
- [10] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Trans on Evolutionary Computation, 1997, 1(1): 53-66.
- [11] 庄慧忠, 杜树新, 吴铁军. 机器人路径规划及相关算法研究[J]. 科学通报, 2004, 20(3): 210-215.
- [12] 席裕庚, 张纯刚. 一类动态不确定环境下机器人的滚动路径规划[J]. 自动化学报, 2002, 28(2): 161-175.