综合保障性技术

一种基于构件重要度的软件可靠性评估方法*

李迎博,谭黎立,王凯旋,梁卓,潘彦鹏

(中国运载火箭技术研究院,北京 100076)

摘要:基于构件开发的软件可靠性取决于构件可靠性,但构件可靠性受测试覆盖性、使用频次等影响较大,因此新研构件的可靠性评价往往较低。当成熟软件引入新研构件时,无论该构件承担的功能重要性高低,软件的可靠性评估值都会出现较大变动。针对此问题,建立一种基于功能场景的马尔可夫链软件模型,利用层次分析法评估得到模型中各构件的重要度,以此为基础建立基于构件重要度的软件可靠性评估模型。计算结果表明,该模型可以更准确地反映基于构件开发的软件可靠性水平。

关键词:软件;构件重要度;可靠性评估;马尔可夫链;层次分析法

doi:10.3969/j.issn.1009-086x.2022.06.013

中图分类号:N945.1;TP311.5;TJ0 **文献标志码**:A **文章编号**:1009-086X(2022)-06-0103-07

Software Reliability Evaluation Method Based on Component Importance

LI Ying-bo, TAN Li-li, WANG Kai-xuan, LIANG Zhuo, PAN Yan-peng

(China Academy of Launch Vehicle Technology, Beijing 100076, China)

Abstract: The software reliability of component-based development depends on component reliability, but component reliability is affected by test coverage and usage frequency. Therefore, the reliability evaluation of new components is often low. When a newly developed component is introduced into mature software, no matter the importance of component, the software reliability evaluation value will change greatly. To solve this problem, taking a software as the research object, a Markov chain software model based on functional scenario is established. the importance of each component in the model is evaluated by analytic hierarchy process. On this basis, a software reliability evaluation model based on component importance is established. The simulation results show that the model can more accurately reflect the software reliability level of component-based development.

Keywords: software; component importance; reliability evaluation; Markov chain; analytic hierarchy process (AHP)

* **收稿日期**:2022-03-08;**修回日期**:2022-07-01

第一作者简介:李迎博(1988-),男,山东济宁人。工程师,硕士,研究方向为软件可靠性设计。

通信地址:100076 北京市丰台区东高地街道益丰园小区 2 号楼 506 **E-mail**:2272liyingbo@163.com

0 引言

基于构件的软件开发技术是面向对象技术的进一步发展,软件开发人员可以直接使用已有的构件,通过组装来完成系统的开发,从而提高软件的研制效率和可靠性。基于构件开发的软件,其可靠性很大程度上取决于构件的可靠性。目前常用的可靠性评估方法为以构件为中心,建立构件之间的信息流图,根据构件的可靠性及构件间的转移概率得到软件的可靠性[1-5]。

构件的可靠性评估往往与软件分开进行,软件开发人员从构件库中取出构件使用时,构件的可靠性已经确定。且构件的可靠性评估与软件不同,其作为软件底层元素,需大量测试和应用数据支撑评估。尤其对于许多难以通过遍历进行测试覆盖的复杂构件,其可靠性评估与使用频次及使用场景有较大关系。如在航空、航天领域,与飞行相关的构件可靠性与其参加飞行试验的次数直接关联。当在成熟软件中引入新研制的构件时,若新研构件可靠性评估值不高,即便该构件完成的功能重要性较低,也往往会使成熟软件的可靠性评估值出现较大的下降。因此,需要建立一种与构件重要度相关的软件评估模型,以反映软件可靠性的真实水平。

本文针对某数据管理软件,首先构建了一种基于功能场景的马尔可夫链软件模型,之后利用层次分析法评估模型中各构件的重要度,以此为基础建立基于构件重要度的软件可靠性评估模型。通过实际可靠性计算结果,证明了模型的合理性。

1 模型建立

(1) 马尔可夫链[3]相关概念

在随机过程中把马尔可夫过程定义为:设有一随机过程$\{X(t), t \in T\}$,$t_1 < t_2 < \cdots < t_m < t_{m+1} \in T$,若在$t_1, t_2, \cdots, t_m, t_{m+1}$时,$X(t)$对应的观测值为$x_1, x_2, \cdots, x_m, x_{m+1}$,同时还满足表达式(1)的条件:

$$P\{X(t_{m+1}) \in A | X(t_i) \in A(i = 1, 2, \cdots, m)\} = P\{X(t_{m+1}) \in A | X(t_m) = x_m \in A\}, \quad (1)$$

则称此过程为马尔可夫过程。其中:A为马尔可夫过程的全体状态空间;$X(t)$为可能的取值,也称为状态。如果$\{X(t), t \in T\}$中,t取离散值,则该马尔可夫过程就是马尔可夫链,马尔可夫链具有无后效性,即当前所处的状态与当前时刻以前的状态无关。因此可以把软件的使用过程看作是马尔可夫链,利用马尔可夫链构建软件模型。

(2) 基于功能场景的马尔可夫链软件模型

目前常用的马尔可夫链软件模型中,一般以构件为中心,通过遍历或人工估计得到构件之间的转移概率,但此种方式耗时较长且易受主观因素影响导致准确度不高[3]。本文以软件功能为中心,进行构件分解及路径组合,提出了一种基于功能场景的马尔可夫链软件模型建立方法。

假设某数据管理软件,其主要功能见表1。

表1 软件主要功能

Table 1 Main functions of the software

序号	软件功能	软件名称
1	生成外部软件所需的本地数据文件	数据管理软件
2	调用外部软件	

根据功能将软件划分为6个应用场景,每种场景包含4个元素,分别为:场景名称、场景执行概率、场景前置条件、场景后置条件。则可得到软件场景元素表,见表2。

表2 软件功能场景元素表

Table 2 Software function scenario elements

序号	场景名称	执行概率/%	前置条件	后置条件
1	版本查询	5	版本查询	结果信息显示
2	数据录入与文件生成	20	数据正确性判断	本地文件生成
3	数据读取与文件生成	15	数据读取、数据正确性判断	本地文件生成
4	外部数据拷贝与文件生成	10	数据读取	本地文件生成
5	外部软件调用	20	软件调用	结果信息显示
6	线下数据比对	30	数据读取、数据比对	结果信息显示

其中场景的执行概率可以很容易通过需求文档和日常操作统计得到,前置、后置条件即为软件中的构件。可以得到软件中的构件组成见表3。

表3 软件构件组成

Table 3 Software component composition

序号	构件名称
1	版本查询
2	结果信息显示
3	数据正确性判断
4	本地文件生成
5	数据读取
6	软件调用
7	数据比对

由表2,3可以得到基于功能场景的马尔可夫链软件模型,其中软件模型的节点表示软件构件,节点之间的边为应用场景,见图1。

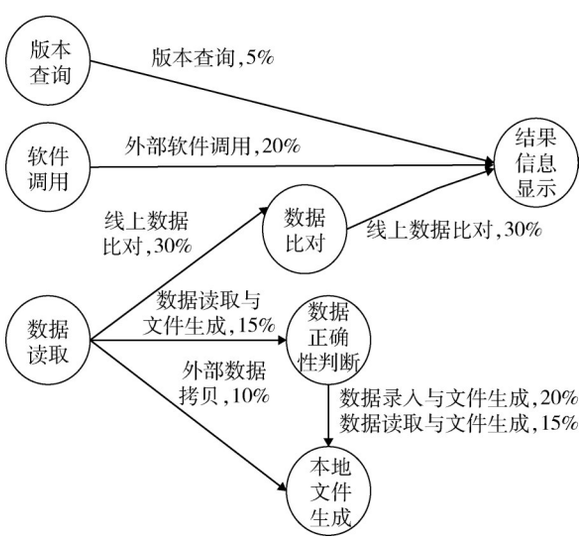

图1 马尔可夫链软件模型

Fig. 1 Markov chain software model

根据任务需求,对软件非主要功能场景进行拓展,增加线上数据比对场景,拓展后的软件主要功能、功能场景元素表及构件组成见表4~6。

同样得到马尔可夫链软件模型,见图2。

由上文中的表1~6及图1,2可知,功能拓展前后的软件主要功能未发生变化,拓展前的软件包含7个构件,拓展后的软件包含8个构件,新增数据收发构件。而根据软件主要功能可知,该软件重要度较高的构件为本地文件生成构件和软件调用构件,数据收发构件的重要度较低。

表4 软件主要功能(拓展后)

Table 4 Main functions of the software(after expansion)

序号	软件功能	软件名称
1	生成外部软件所需的本地数据文件	数据管理软件
2	调用外部软件	

表5 软件功能场景元素表(拓展后)

Table 5 Software function scenario element (after expansion)

序号	场景名称	执行概率/%	前置条件	后置条件
1	版本查询	5	版本查询	结果信息显示
2	数据录入与文件生成	20	数据正确性判断	本地文件生成
3	数据读取与文件生成	15	数据读取、数据正确性判断	本地文件生成
4	外部数据拷贝与文件生成	10	数据读取	本地文件生成
5	外部软件调用	20	软件调用	结果信息显示
6	线下数据比对	10	数据读取、数据比对	结果信息显示
7	线上数据比对	20	数据收发、数据比对	结果信息显示

表6 软件构件组成(拓展后)

Table 6 Software component composition(after expansion)

序号	构件名称
1	版本查询
2	结果信息显示
3	数据正确性判断
4	本地文件生成
5	数据读取
6	软件调用
7	数据比对
8	数据收发

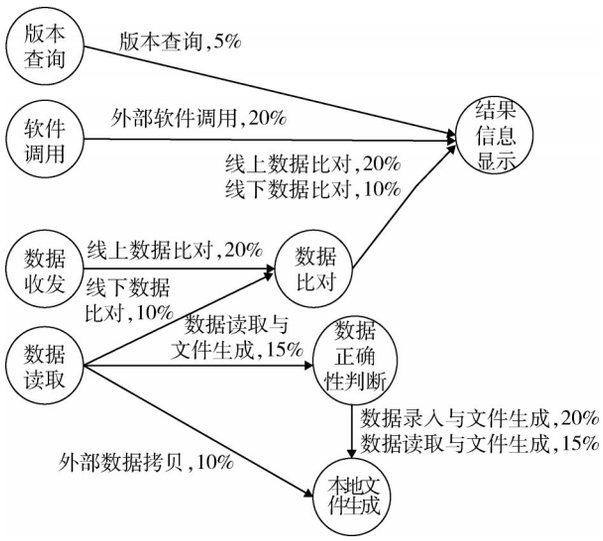

图 2 马尔可夫链软件模型(拓展后)

Fig. 2 Markov chain software model(after expansion)

2 构件重要度评估

构件重要度评估分为直接法和间接法 2 种。直接法一般建立与构件代码行数、危害程度等要素相关的数学模型,通过模型计算得到构件重要度[6-10],但此方法对不同类型构件的适应性较差,且很难将所有要素考虑在内。间接法一般使用人工给出构件之间的相对关系,但此种方法易受主观影响,当构件数目较多时,准确度较差。本文利用层次分析法[11-15]评估得到构件重要度,该方法可通过构件间重要度的两两比较,降低主观风险,且不受构件类型影响,具有良好的适应性和准确性。

首先,构建软件功能场景扩展前后的构件评价体系,如图 3,4 所示。

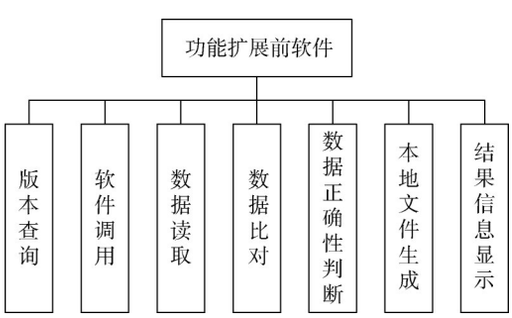

图 3 扩展前软件构件组成

Fig. 3 Composition of software components before expansion

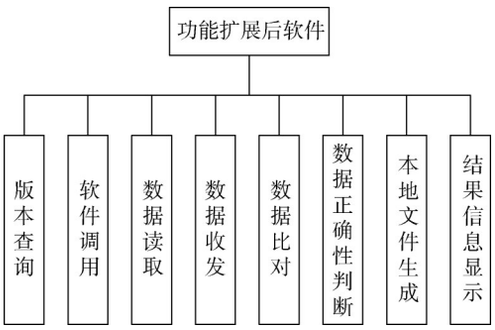

图 4 扩展后软件构件组成

Fig. 4 Composition of software components after expansion

其次,分别建立 2 个构件体系的判断矩阵 $\boldsymbol{A}_1$,$\boldsymbol{A}_2$,如下:

$$\boldsymbol{A}_1=\begin{bmatrix}1 & 1/6 & 1/4 & 1/2 & 1/4 & 1/7 & 1\\ 6 & 1 & 2 & 3 & 2 & 1 & 6\\ 4 & 1/2 & 1 & 2 & 1 & 1/2 & 4\\ 2 & 1/3 & 1/2 & 1 & 1/2 & 1/3 & 2\\ 4 & 1/2 & 1 & 2 & 1 & 1/2 & 4\\ 7 & 1 & 2 & 3 & 2 & 1 & 7\\ 1 & 1/6 & 1/4 & 1/2 & 1/4 & 1/7 & 1\end{bmatrix},$$

$$\boldsymbol{A}_2=\begin{bmatrix}1 & 1/6 & 1/4 & 1/2 & 1/2 & 1/4 & 1/7 & 1\\ 6 & 1 & 2 & 3 & 3 & 2 & 1 & 6\\ 4 & 1/2 & 1 & 2 & 2 & 1 & 1/2 & 4\\ 2 & 1/3 & 1/2 & 1 & 1 & 1/2 & 1/4 & 2\\ 2 & 1/3 & 1/2 & 1 & 1 & 1/2 & 1/3 & 2\\ 4 & 1/2 & 1 & 2 & 2 & 1 & 1/2 & 4\\ 7 & 1 & 2 & 4 & 3 & 2 & 1 & 7\\ 1 & 1/6 & 1/4 & 1/2 & 1/2 & 1/4 & 1/7 & 1\end{bmatrix},$$

式中:$\boldsymbol{A}_{1ij}$表示构件 i 相对构件 j 的重要性,重要性赋值采用 1–9 标度法。

之后,根据判断矩阵,计算得到对应的特征向量,特征向量使用对数最小二乘法计算得到,即 $\bar{w}_{1i}=\sqrt[7]{\prod_{j=1}^{7}A_{1ij}}$,$\bar{w}_{2i}=\sqrt[8]{\prod_{j=1}^{8}A_{2ij}}$,结果为

$\bar{\boldsymbol{w}}_1$=(0.357 3,2.379 6,1.345 9,0.730 6,1.345 9,2.486 7,0.357 3),

$\bar{\boldsymbol{w}}_2$=(0.352 7,2.449 5,1.414 2,0.733,0.759 8,1.141 2,2.638 9,0.372 7).

之后,进行归一化处理,得到权重系数即构件重要度数值向量 $\boldsymbol{w}_{1i}=\dfrac{\bar{\boldsymbol{w}}_{1i}}{\sum_{i=1}^{7}\bar{\boldsymbol{w}}_{1i}}$,$\boldsymbol{w}_{2i}=\dfrac{\bar{\boldsymbol{w}}_{2i}}{\sum_{i=1}^{8}\bar{\boldsymbol{w}}_{2i}}$,结果如下:

$$\boldsymbol{w}_1 = (0.039\,7, 0.264\,3, 0.149\,5, 0.081\,1, 0.149\,5, 0.276\,2, 0.039\,7),$$

$$\boldsymbol{w}_2 = (0.036\,7, 0.241\,2, 0.139, 3, 0.072\,2, 0.074\,8, 0.139\,3, 0.259\,9, 0.036\,7).$$

最后进行一致性检验,一致性校验指标根据判断矩阵 $\boldsymbol{A}_i$ 最大特征根 $\lambda_{\max_i}$ 计算得到,公式为 $C.I. = \dfrac{\lambda_{\max_i} - n}{n-1}$,可得 $CI_1 = 0.002\,4$, $CI_2 = 0.002\,3$,均满足要求。

则计算得到的 $\boldsymbol{w}_1, \boldsymbol{w}_2$ 中的要素即为2个软件中各构件的重要度。

3 基于构件重要度的软件可靠性模型

(1) 软件可靠性模型

本文提出一个函数体模型,该模型与构件重要度、构件数量、构件可靠性、构件在单个场景中执行概率、场景数量有关,即

$$\xi = f(\boldsymbol{P}, \boldsymbol{w}, \boldsymbol{C}, M, N),$$

式中:M 为软件构件数量;N 为软件功能场景数量;P_{ij} 为单个软件构件在单个场景中的执行概率;C_i 为单个软件构件的可靠度;w_i 为单个软件构件重要度。

根据构件在各个场景中的执行概率计算得到构件在软件中的执行概率,即

$$P_i = \sum_{j=1}^{N} P_{ij}. \tag{2}$$

将单个构件执行概率与构件重要度相结合,并进行归一化处理,得到单个构件综合重要度,即

$$Q_i = \frac{\left(\sum_{j=1}^{N} P_{ij}\right) \cdot w_i}{\sum_{i=1}^{M}\left(\sum_{j=1}^{N} P_{ij}\right) \cdot w_i}. \tag{3}$$

综合单个构件可靠性,最终可以计算得到软件可靠性模型,即

$$\xi = \sum_{i=1}^{M} \frac{\left(\sum_{j=1}^{N} P_{ij}\right) \cdot w_i}{\sum_{i=1}^{M}\left(\sum_{j=1}^{N} P_{ij}\right) \cdot w_i} \cdot C_i. \tag{4}$$

(2) 构件可靠性评价

得到表达式(4)后,还需要确定单个构件的可靠度 C_i,构件可靠度取决于构件测试、确认及应用情况,即

$$C_i = K_K(90.0 + 2.0K_S + 2.0K_Y + 3.0K_X + C_F + C_Y). \tag{5}$$

表达式(5)中各项取值标准及含义见表7。

表7 构件可靠性取值标准

Table 7 Component reliable value standard

表达式	取值标准	说明
K_K	进行了开发方测试取1,未进行开发方测试取0	开发方测试系数
K_S	进行了第三方测试取1,未进行第三方测试取0	第三方测试系数
K_Y	进行了验收确认取1,未进行验收确认取0	验收确认系数
K_X	进行了系统级测试(且考核到所有功能、性能等实现情况)取1,否则取0	系统测试考核系数
C_F	进行了一次飞行试验测试考核取1.0,后续每进行一次增加0.1,最大为2.0	飞行试验测试考核情况
C_Y	每使用一次增加0.1,最高为0.9	进入构件库后,被软件使用情况

4 计算结果

首先根据构件可靠性评价模型,得到软件中各构件可靠性见表8。

表8 构件可靠性

Table 8 Component reliability

序号	构件名称	可靠性/%
1	版本查询	99.5
2	软件调用	99.8
3	数据读取	99.5
4	数据收发	98.0
5	数据比对	99.6
6	数据正确性判断	99.8
7	本地文件生成	99.9
8	结果信息显示	99.7

其中,数据收发构件由于为新研模块,尚未进入构件库,已经进行了开发方测试、第三方测试、确认验收、系统测试及一次飞行试验考核,因此可靠性取

值为98.0%。其他构件均比较成熟,可靠性较高。

依据软件可靠性计算模型及构件可靠性,进行如下计算,以验证算法设计的正确性。

(1) 同等构件重要度计算

将所有构件重要度设为相同值,则软件可靠性只与构件可靠性及构件执行概率有关,可得到软件可靠性计算结果如表9所示。

表9 同等构件重要度条件下软件可靠性

Table 9 Software reliability under the condition of equal component importance %

功能场景拓展前软件可靠性	功能场景拓展后软件可靠性	计算条件
99.698	99.575	所有构件重要度设为相同值

可见功能场景拓展后的软件可靠性下降0.123。

(2) 不同构件重要度计算

将所有构件重要度设为前文实际计算值,则根据可靠性计算模型及构件可靠性,得到软件可靠性计算结果如表10所示。

表10 不同构件重要度条件下软件可靠性

Table 10 Software reliability under the condition of different component importance %

功能场景拓展前软件可靠性	功能场景拓展后软件可靠性	计算条件
99.744	99.688	所有构件重要度设为计算值

可见功能场景拓展后的软件可靠性下降值为0.056,而同等构件重要度条件下的软件可靠度下降值为0.123,下降值幅度大大降低。

而前文提到本地文件生成、软件调用2个构件为软件核心构件,在核心功能未发生变更,只增加一个不重要构件的情况下,软件可靠性下降程度越小越好。因此,基于构件重要度的软件可靠性模型可以更好地反映基于构件开发的软件的可靠性。

5 结束语

本文构建了一种基于功能场景的马尔可夫链软件模型,利用层次分析法评估模型中各构件的重要度,并建立基于构件重要度的软件可靠性评估模型。实际计算结果表明,该模型在基于构件的软件可靠性评估中,具有良好的应用效果。

参 考 文 献:

[1] 张娟. 基于架构的软件可靠性计算[D]. 杭州:浙江理工大学,2016.
ZHANG Juan. Architecture-Based Software Reliability Computing[D]. Hangzhou: Zhejiang Sci-Tech University,2016.

[2] 陈悦. 基于构件分析的软件可靠性评估与分配模型及应用[D]. 南京:南京航空航天大学,2020.
CHEN Yue. Research and Application of Component-Based Software Reliability Assessment and Allocation Model[D]. Nanjing: Nanjing University of Aeronautics and Astronautics,2020.

[3] 刘洋,于磊,徐炜珊,等. 基于层次结构markov链的软件可靠性建模方法[J]. 信息工程大学学报,2015,16(4):477-482.
LIU Yang, YU Lei, XU Wei-shan, et al. Software Reliability Modeling Based on Hierarchical Structure of Markov Chain[J]. Journal of Information Engineering University, 2015,16(4):477-482.

[4] 吴家伟. 面向复杂软件系统的可靠性评估方法[D]. 合肥:合肥工业大学,2018.
WU Jia-wei. Research of the Assessment Approach for Complex Software Reliability[D]. Hefei: Hefei University of Technology,2018.

[5] 吴彩华,马建朝,魏海涛,等. 基于Markov链的软件可靠性早期评估研究[J]. 空军预警学院学报,2014,28(3):199-202.
WU Cai-hua, MA Jian-chao, WEI Hai-tao, et al. Research on Early Evaluation of Software Reliability Based on Markov Chain[J]. Journal of Air Force Early Warning Academy, 2014,28(3):199-202.

[6] 唐佩佳,谢永杰,吴安波,等. 基于马尔可夫链的构件软件可靠性评估模型[J]. 计算机应用,2016,32(S2):262-265.
TANG Pei-jia,XIE Yong-jie,WU An-bo, et al. Reliability Evaluation Model Based on Markov Chain for Component-Based Software [J]. Journal of Computer Applications, 2016,32(S2):262-265.

[7] 刘志祥,刘杰,李丹,等. 一种基于马尔可夫模型的软件可靠性评估方法[J]. 软件可靠性与评测技术,2020,30(4):38-42.

LIU Zhi-xiang, LIU Jie, LI Dan, et al. Software Reliability Evaluation Based on Markov Model[J]. Electronic Product Reliability and Environmental Testing, 2012, 30(4):38-42.

[8] 幺强. 基于类分析与状态马尔科夫过程的Web构件软件可靠性建模[D]. 广州:华南理工大学,2013.
YAO Qiang. Software Reliability Model of Web Component System Based on Class Analysis and State Markov Procedure[D]. Guangzhou: South China University of Technology, 2013.

[9] 时圣革,常文兵. 马尔可夫链的模块化软件可靠性研究[J]. 火力指挥控制,2009,34(1):153-156,159.
SHI Sheng-ge, CHANG Wen-bing. Study on Reliability of Modularized Software based on Markov Chain[J]. Fire Control & Command Control, 2009, 34(1): 153-156, 159.

[10] 吴亮,方建勇. 基于优化Markov链使用模型的可靠性测试技术[J]. 指挥控制与仿真,2015, 37(5):124-127,138.
WU Liang, FANG Jian-yong. Software Reliability Testing Method Based on Optimizated Markov Chain Usage Model[J]. Command Control & Simulation, 2015, 37(5):124-127, 138.

[11] 杨东. 基于层次分析法在软件项目质量影响因素分析中的应用[J]. 项目管理技术,2021(6):155-158.
YANG Dong. The Application of Analytic Hierarchy Process in the Analysis of Influencing Factors of Software Project Quality[J]. Project Management Technology, 2021(6):155-158.

[12] 蒲文栋,阳红,王琰. 基于层次分析法的软件缺陷评估方法设计[J]. 电声技术,2021, 45(4):74-77.
PU Wen-dong, YANG Hong, WANG Yan. Design of Software Defect Evaluation Method Based on Analytic Hierarchy Process[J]. Audio Engineering, 2021, 45(4):74-77.

[13] 李波,黄新静. MATLAB软件算法在层次分析法中的应用[J]. 计算机应用,2018, 37(12):35-38,54.
LI Bo, HUANG Xin-jing. Application of MATLAB Software Algorithm in Analytic Hierarchy Process [J]. Techniques of Automation and Applications, 2018, 37(12):35-38, 54.

[14] 胡晓冉,左家平,王坤. 基于层次分析法的软件质量量化研究[J]. 计算机应用与软件,2013(11):138-141.
HU Xiao-ran, ZUO Jia-ping, WANG Kun. Study on AHP-Based Quantification of Software Quality[J]. Computer Applications and Software, 2013(11):138-141.

[15] 王志,刘艳辉,杨欢. 层次分析法在软件度量中的应用[J]. 计算机工程与设计,2017, 38(1):144-148.
WANG Zhi, LIU Yan-hui, YANG Huan. Application of AHP in Software Process Measurement[J]. Computer Engineering and Design, 2017, 38(1):144-148.