

An Approach Assessing Software Trustworthiness

Xinglong Xie* and Weixian Xue

Xi'an University of Technology, 710048, China

*Corresponding author

Abstract—This paper is aimed at presenting a technique appraising software trustworthiness grounded on Grey correlation analysis, which performs calculating of the weight for each index by virtue of trapezoidal fuzzy number. This paper initially works out at positive and negative ideal values for trustworthiness indicators, whereby to figure out the Grey correlation degree concerning the index value and the ideal value whereof. Grounded on this, the paper estimates the value for software trustworthiness and the empirical evidence reveals that the specified methodology is of validity and accuracy. Hence, this technique can be applied into more studies on software trustworthiness.

Keywords—trustworthiness measure; trustworthiness indicator; positive (negative) ideal value; trapezoidal fuzzy number; grey correlation analysis

I INTRODUCTION

Software trustworthiness, developed on the footing of correctness, dependability, security and real-time, drives at the totality of all features and peculiarities relevant to the capability of a software system satisfying expectations and its implied demand. With the advent of an ever-expanding software scale, this topic presently concerns a great deal, including market demand, modeling, testing, maintaining and run-time supporting for software, and has aroused extensive attention and witnessed a growth in related-studies. And has shifted into comprehensive consideration of multiple attributes such as accountability, security and practicality from focusing on the analyzing, testing and verifying of a single attribute amid the early studies [1, 2, 3, 4, 5, 6]. Especially, Immonen and Palviainen claimed that software worthiness needs to be gauged from three tiers: software component, architecture and system [7]. Ding, Lu and Yang designed a demand-driven model [8], and Zhao and Sun explored techniques concerning trustworthiness assessment [9]. Cai, Zou and Shao came up with a framework appraising the credibility of a supporting software [10]. Li et al. proposed a software trustworthiness model, integrating risk management and cost control [11]. D. Lizcano et al. put forward an automatic framework of software trustworthiness to support verification and validation, which has been demonstrated valid [12].

In general, existing methods are mainly divided into two classes: model-based, which focus on software framework modeling and perform the evaluating assignment prior to software delivery, and, application-based, which seek to make judgment of trustworthiness on evidence attained following software operations. The latter approaches think over multiple properties regarding evidence of trustworthiness during each

stage of the software life cycle yet most of property values suffer from poor information and expert judgment is inevitably subjective, this having triggered a departure from reality. Thus, notwithstanding orthodox theorems, which have set up considerable quantities of trustworthiness-evaluating frameworks amid efficiency-orientation, they are subjected to inadequacy in the number of trustworthiness attributes and draw certain static conclusions in spite of quantifying software trustworthiness. Concerning software trustworthiness assessment, quantitative and qualitative analysis should be employed as well as multiple properties. This would enhance translation of findings stemming from trustworthiness research into actual productivity, diminishing the costs for software failure.

The purpose of this paper is to contribute to a growing body of literature on software trustworthiness assessment through regarding software trustworthiness as a decision-making of multiple fuzzy Grey attributes. We design a framework by means of considering six first-grade indicators in the criteria tier and thirteen second-grade indicators in the sub-criteria tier, which gives our study superior in comparison with traditional methods. Furthermore, we solves the model by resorting to Grey correlation methodology, characterized by both trapezoidal fuzzy number deciding weights and comparison with ideal values. All this allows us to address issues set out in existing software trustworthiness studies.

The reminder of this paper is organized as follows. Section II contains the theoretic basis and modeling of software trustworthiness. Methodology is presented in section III. IV contains empirical analysis. We concluded in section V.

II THEORETIC BASIS AND MODELING

Software trustworthiness appraising is viewed as a systematic engineering, including evidence gleaning, indicator designing, trustworthy attribute measuring, methodology estimating trustworthiness, expert comments and user feedback (see Figure 1). Testimony collection runs through the whole life cycle of software, during which any figure, document or information, which mirrors a certain credibility property, can be treated as evidence. Thus, data concerning norms of all classes, reviews and tests in the course of software development, figures arising from entity analysis and testing, user satisfaction degree and remarks of a third party, all fall into the category of valid reliable proofs, marked by objectiveness, relevancy and availability.

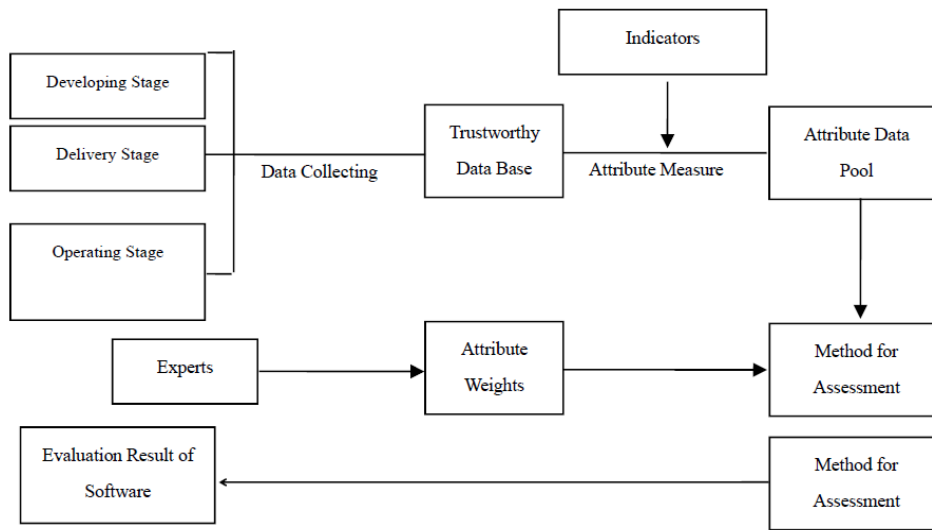


FIGURE I. RESEARCHING STRUCTURE OF SOFTWARE TRUSTWORTHINESS ESTIMATION

Worthiness attribute is understood to be direct information describing and appraising the reliability of a software system, gauged by utilizing indicators, heterogeneous in the relative importance. So, it is vital how to ascertain the weight of an index. This is generally left to experts and consumers, who can give the value of each indicator, whereby weights to be calculated. The core in software trustworthiness assessment lies in the algorithm, employed for the sake of calculating software system behavior, user-expected behavior, and actual results.

Index framework concerning software trustworthiness evaluation refers to an organism consisting of interrelated reliability attributes and metric values whereof. This has

aroused attention from scholars and registered an increase in the number of studies. Zhao et al. proposed a technique for confidence level gauging of software by virtue of Pi calculation [13]. Yang et al. brought forward a software-behavior-based dynamic structure of indexes, which is able to assess a behavior locus and examine the deviation degree regarding a point scene by performing calculation of context values invoked by the system and establishing the constraint rules regarding system-calling parameters [14]. In view of software attribute measurement and operability, this paper prefer to opt for usability, reliability, security, real-time and maintainability as indicators at the first rank, under which there are a certain number of sub-indicators(Figure II).

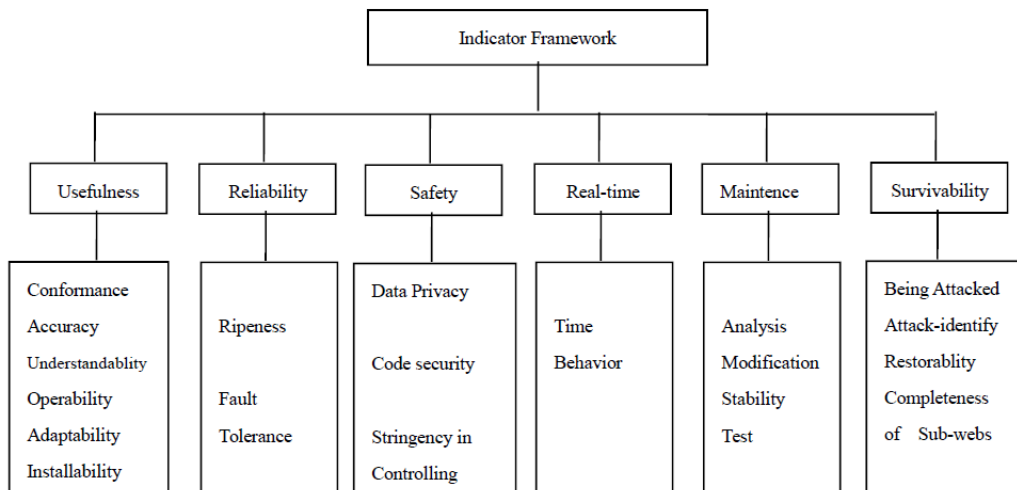


FIGURE II. FRAMEWORK REGARDING APPRAISING SOFTWARE TRUSTWORTHINESS

III FUZZY GREY RELATIONAL METHOD

A. Indicator Framework and Ideal Point

The framework evaluating software trustworthiness can be referred as a system characterized by uncertainties, small

samples and information insufficiency attributable to difficulties in the acquisition of indicator values. In this article, $G_j \in G(j=1,2,\dots,n)$ denotes the attributes concerning software trustworthiness, $A_i (i=1,2,\dots,m)$ a software remaining to be appraised. $S=(S_{ij})_{m \times n}$, in which the larger some indexes, the better they will be while other indicators are on the contrary:

the smaller, the better. Normalization is employed to process these indexes for eliminating consequences exerted on results to be estimated,

this yielding matrix $R=(r_{ij})m \times n$:

$$r_{ij} = \frac{s_{ij}}{\max\{s_{ij}\}} \quad (1 \leq i \leq m) \quad \text{for efficiency indexes} \quad (1a)$$

$$r_{ij} = \frac{\min\{s_{ij}\}}{s_{ij}} \quad (1 \leq i \leq m) \quad \text{for cost indexes} \quad (1b)$$

Utilizing ideal points can give rise to reference indicator series, for which, the maximum for an indicator acting as the ideal point while the minimum the negative counterpart.

$$r_i^+ = \max_j \{r_{ij}\} \quad (j=1,2,\dots,n) = (r_1^+, r_2^+, \dots, r_n^+) \quad (2a)$$

And

$$r_i^- = \min_j \{r_{ij}\} \quad (j=1,2,\dots,n) = (r_1^-, r_2^-, \dots, r_n^-) \quad (2b)$$

B. Attribute Weight Calculation

For fuzzy set F in the real number field, there exists $a < b < c < d$, if the membership function is:

$$\mu_F(x) = \frac{x-a}{b-a} \quad (a < x < b) \quad (3a)$$

$$\mu_F(x) = 1 \quad (b \leq x \leq c) \quad (3b)$$

$$\mu_F(x) = \frac{x-c}{d-c} \quad (c < x < d) \quad (3c)$$

$$\mu_F(x) = 0 \quad (else) \quad (3d)$$

Then this subjection function is viewed as the fuzzy number of trapezoid, signed by (a,b,c,d). It is employed to compute the weight of a trustworthiness property by taking following steps: let $W_k=w_j(k)$ be the weight for G_j given by the expert panel, this leading to formation of trustworthiness attribute weight matrix $W_k=(w_j)$. $E_k \in [0,1](k=1,2,\dots,p)$ indicates the decision-making weight of experts or users, satisfying $\sum E_k=1$. Table I reveals the principle concerning evaluation.

In conformity with the membership function of ladder-shaped vague numbers, we can find a fuzzy judgment of indicator G_j in the relative importance by specialist k:

$$w'_{jk} = (a_{jk}, b_{jk}, c_{jk}, d_{jk})$$

Whereby to secure the comprehensive weight for G_j by the expert team:

$$w'_j = (a_j, b_j, c_j, d_j) = \left(\sum_{k=1}^p E_k a_{jk}, \sum_{k=1}^p E_k b_{jk}, \sum_{k=1}^p E_k c_{jk}, \sum_{k=1}^p E_k d_{jk} \right) \quad (4)$$

Let the optimistic co-efficient of experts be 0.5, then there exists the following:

$$d(w'_j) = \frac{(a_j + b_j + c_j + d_j)}{4} \quad (5)$$

Weight is formalized by adopting the following specification to generate a weight vector:

$$w_j = \frac{d(w'_j)}{\sum_{j=1}^n d(w'_j)} \quad (6)$$

Thus, the weight vector for index G_j can be expressed in

the following form: $w=(w_1, \dots, w_j, \dots, w_n)$ where $\sum_{j=1}^n w_j=1$

C. Trustworthiness Grade

Formula 1 and 2, coupled with 7, can conclude distance V_j^+ between real and positively-ideal values and V_j^- between real and adversely-ideal values:

$$V_j^+ = |r_j - r_j^+| \quad (7a)$$

And

$$V_j^- = |r_j - r_j^-| \quad (7b)$$

Next, to attain the correlation coefficient between trustworthiness attributes and the ideal set through following specifications:

$$\beta_j^+ = \frac{\min_j V_j^+ + \rho \max_j V_j^+}{V_j^+ + \rho \max_j V_j^+} \quad (8a)$$

And

$$\beta_j^- = \frac{\min_j V_j^- + \rho \max_j V_j^-}{V_j^- + \rho \max_j V_j^-} \tag{8b}$$

Where β_j^+ refers the coefficient, which associates the trustworthiness property with and the positively ideal property, and, β_j^- means one, which connects the trustworthiness property to the negatively ideal property, and ρ is termed as distinguishing coefficient, falling into $[0,1]$, generally being fixed at 0.5. This, plus using formula (6), can contribute to production of relevancy degree between the scheme for the attribute set concerning software reliability and the positively (adversely) ideal attribute set:

$$\beta^+ = \frac{\sum_{j=1}^n w_j \times \beta_j^+}{\sum_{j=1}^n w_j} \tag{9a}$$

And

$$\beta^- = \frac{\sum_{j=1}^n w_j \times \beta_j^-}{\sum_{j=1}^n w_j} \tag{9b}$$

Synthesizing β^+ and β^- can give rise to generating of estimated numerical value for software trustworthiness:

$$\beta = \frac{\beta^+}{\beta^+ + \beta^-} \tag{10}$$

Comparing β with the criteria can find out at which level a software being gauged is positioned in the case of trustworthiness.

IV EMPIRICAL ANALYSIS

This paper would aim to measure the self-adapting temperature software at the machine room of a telecommunication firm, which places high requirements on software properties such as reliability and just-in-time. To achieve this end, the project team develops three sets of embedded software systems by virtue of resorting to compiling C and Java language. On the footing of software trustworthiness features, this paper designs an indicator framework via having some of indexes stemming from Figure II as references (see Table II).

TABLE II. FRAMEWORK

| Goal | Software Trustworthiness Value | | | | | | |
|--------------|-----------------------------------|--------------------------------------|-----------------------------------|--|---------------------------------|----------------------------------|---------------------------|
| Criteria | Usability/I ₁ | | | Reliability/I ₂ | | Security/I ₃ | |
| Sub-criteria | Function Accuracy/I ₁₁ | Operability/I ₁₂ | Easy Installation/I ₁₃ | Error-making Frequency/I ₂₁ | Fault Tolerance/I ₂₂ | Confidentiality/I ₃₁ | Integrity/I ₃₂ |
| Criteria | Real-time/I ₄ | Maintainability/I ₅ | | | Survivalality/I ₆ | | |
| Sub-criteria | Timeliness/I ₄₁ | Diagnosis difficulty/I ₅₁ | Modifiability/I ₅₂ | Stability/I ₅₃ | Reversibility/I ₆₁ | Self-improvement/I ₆₂ | |

Three experts are invited to assign each sub-criterion a weight by fuzzy words described on the Table I. For this specialist group, each is viewed as having the same position in the weight decision-making, namely, E1= E2= E3=1/3, and all first-grade indicator are equally treated in the weight distribution, coming at 1/6.

TABLE I. EVALUATION OF BLUR WORDS AND EXPRESSIONS

| Vague words | Unimportant | Less important | Important | More important | Extremely important |
|-------------|-------------|----------------|-----------|----------------|---------------------|
| (a,b,c,d) | (0,0,0,3) | (0.3,3,3,5) | (3,5,5,7) | (5,7,7,9) | (7,9,9,10) |

With regard to the initial procedure, it is dedicated to estimating the indicator vector. Take three sub-criteria under I1 for example. The weights concluded by three specialists and listed on table 3, the comprehensive weights of specialists generated by formula (4), the blur weights attained by (5), and post-formalized values stemming from (6) can lead to production of the final weight vector W1=(w11, w12, w13)=(0.42,0.31,0.27). In the same token, weight vectors for other second-level indicators can be available. This, in couple with the weights concerning first grade indicators, can bring about engendering of comprehensive weight vectors for second grade indexes, denoted by W=(W1, W2,...,Wn).

The next step is to find a matrix of assessing trustworthiness S3×13 through gauging each attribute value for software A1(i=1,2,3) to derive Grey number vector G_j (j=1,2,...,n):

$$S_{3 \times 13} = \begin{bmatrix} 0.91 & 0.93 & 0.92 & 0.95 & 0.84 & 5 & 5 & 0.99 & 0.94 & 0.90 & 4 & 0.93 & 0.84 \\ 0.87 & 0.91 & 0.96 & 0.97 & 0.83 & 3 & 5 & 0.90 & 0.95 & 0.93 & 3 & 0.98 & 0.91 \\ 0.96 & 0.94 & 0.95 & 0.97 & 0.95 & 4 & 4 & 0.94 & 0.95 & 0.85 & 4 & 0.95 & 0.90 \end{bmatrix}$$

normalizing which can generate:

$$R_{3 \times 13} = \begin{bmatrix} 0.95 & 0.98 & 0.94 & 1 & 0.89 & 1 & 1 & 0.91 & 1 & 0.96 & 1 & 0.95 & 0.92 \\ 0.91 & 0.96 & 1 & 0.98 & 0.88 & 0.6 & 1 & 1 & 0.98 & 1 & 0.75 & 1 & 1 \\ 1 & 1 & 0.97 & 0.98 & 1 & 0.8 & 0.8 & 0.95 & 0.99 & 0.91 & 1 & 0.97 & 0.99 \end{bmatrix}$$

Third, the positively ideal point and negatively are ascertained based on (2):

$$r^+ = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$$

$$r^- = (0.91, 0.96, 0.94, 0.98, 0.88, 0.6, 0.8, 0.91, 0.98, 0.91, 0.75, 0.95, 0.92)$$

Finally, we, recurring to these formularies from (7) to (10), figure out the composite correlative degree between A_i and the ideal scheme: $\beta=(\beta_1,\beta_2,\beta_3)=(0.92,0.94,0.93)$. Evidently, software A_2 enjoys the highest trustworthiness, confirmed by $\beta_2=0.94$, closest to 1, in conformity with the argument from some current literature.

V CONCLUSIONS AND PROPOSALS

The contribution of this paper is to provide an evaluation of software trustworthiness. We extend the current analysis by employing Grey correlation methodology as most current literature on software trustworthiness assessment, suffers from incompleteness in the estimation of trustworthiness values. This allows the study to improve accuracy through trapezoidal fuzzy number, lessening negative effects of expert subjectivity on indicator weight estimation. The proposed methodology is proved to be of validity, rationality and appropriate amount of calculation and worthwhile to be applied into more studies on software trustworthiness.

However, a number of issues should be addressed in software trustworthiness evaluation. Among these are critical technology development and research for modern methods such as meta model, data processing and quantification. Also of interest would be designing a set of decision-making rules, which need to be refined and optimized. With regard to a certain specific software, index framework should be established on the footing of demand for reliability as disparities in the selection of indicator frameworks may induce the degree of correlation to be variable, exerting impacts on final results. At last, more research can be done to better capture attributes' own interactions and mechanisms whereof.

ACKNOWLEDGMENT

The authors are thankful to the national social science foundation of China under grant No.17BJL005.

REFERENCES

- [1] B.Steffen, H.Wilhelm, P. Alexandra, Trustworthy software systems:a discussion of basic concepts and terminology, ACM SIGSOFT Software Engineering Notes Archive. 31(2006) 1-18.
- [2] T.Bao, S.F.Liu, X.Y.Wang, Research on trustworthiness evaluation method for domain software based on actual evidence, Chinese Journal Of Electronics. 20(2011) 824-851.
- [3] V. Basiliv, P. Donzelli, S.Asgari, A unified model of dependability: capture dependability in context, IEEE Software. 21(2004) 19-25.
- [4] B.Whitworth, J Fjermestad, D.Mahinda, The web of system performance, Communications of the ACM.49(2006) 93-99.
- [5] G.Water, Software dependability evaluation based on markov usage models, Performance Evaluation.40(2000) 199-222.
- [6] Y. Zhang, B.Fang, C.Y. Xu, Trustworthy Metrics Models for Internet Ware, Wuhan University Journal of Natural Sciences. 13(2008) pp. 547-552.
- [7] A. Immonen,M. Palviaine, Trustworthiness evaluation and testing of open source components, in: QSIC 07Proceedings of the Seventh International Conference on Quality Software, IEEE Computer Society ,Washington,2007, pp316-321.
- [8] Sh. Ding, F.J. Lu, Sh. L.Yang, A Requirement-driven software trustworthiness evaluation and evolution model, Journal of Computer Research and Development. 48(2011) 647-655.
- [9] H.Q.Zhao, J. Sun, A methodological study of evaluating dependability of SOA software system,Chinese Journal of Computers.33(2010)

2202-2210.

- [10] S.B.Cai, Y. Zh. Zou, L. S. Shao, Framework supporting software assets evaluation on trustworthiness, Journal of Software. 2(2010) 359-372.
- [11] J.P. Li, M.L. Li, D.S.Wu, H.Song, An integrated risk management and optimization model for trustworthy software process management, The Journal of Systems and Software. 89(2014) 63-75.
- [12] D. Lizcano, J.Soriano, G.lopez, J.J.Gutierrez, Automatic verification and validation wizard in web-centered end user software engineering, The Journal of Systems and Software.125(2017) 47-67.
- [13] Q. Zhao, H. Q.Wang, G.Sh. Feng, Measuring method of software dependability based on Pi calculus, Journal of Jilin University: Engineering and Technology Edition.41(2011) 1685-1689.
- [14] X.H.Yang, Zhou X.H. Zhou, J.F. Tian, Novel dynamic trusted evaluation model of software behavior, Journal of Chinese Computer Systems.31(2010) 2113-2120.