symmetry

MDPI

Article

A Novel Software Trustworthiness Evaluation Strategy via Relationships between Criteria

Hongwei Tao *, Qiaoling Cao, Haoran Chen, Yang Xian, Songtao Shang and Xiaoxu Niu

College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China
* Correspondence: hongweitao@zzuli.edu.cn

Abstract: Software trustworthiness evaluation is regarded as a multi-criteria decision-making problem. However, most current software trustworthiness evaluation methods do not consider the relationships between criteria. In this paper, we present a software trustworthiness evaluation strategy via the relationships between criteria. Because the trustworthy attribute degree is evaluated by a criterion, a trustworthy attribute measurement method based on fuzzy theory is first proposed, and the relationships between criteria are described by cooperative and conflicting degrees between criteria. Then, a measure formula for the symmetric substitutivity between criteria is proposed, and the cooperative degree between criteria is taken as the approximation of the symmetric substitutivity between criteria. With the help of the symmetric substitutivity between criteria, the software trustworthiness measurement model obtained by axiomatic approaches is applied to aggregate the degree to which each optional software product meets each objective. Finally, the candidate software products are sorted according to the trustworthiness aggregation results, and the optimal product is obtained from the alternative software products on the basis of the sorting results. The theoretical rationality of the measurement model is validated by proving that it satisfies the desirable properties of software trustworthiness measures, and its effectiveness is demonstrated through a case study.

Keywords: software trustworthiness evaluation; multi-criteria decision making; relationships between criteria; fuzzy theory; trustworthy attribute measurement; machine learning

check for updates

Citation: Tao, H.; Cao, Q.; Chen, H.; Xian, Y.; Shang, S.; Niu, X. A Novel Software Trustworthiness Evaluation Strategy via Relationships between Criteria. *Symmetry* **2022**, *14*, 2458. https://doi.org/10.3390/sym14112458

Academic Editor: Zhixun Su

Received: 14 October 2022
Accepted: 11 November 2022
Published: 20 November 2022

1. Introduction

Many concepts are related to software credibility, such as software dependability, software trustworthiness, and software quality. Software dependability refers to the ability of software to avoid the frequency and severity of service failure by exceeding the acceptable range [1], with a main focus on the acceptable service failure frequency and service failure severity of specific failure classes in a given use environment. Software quality indicates the ability of software to meet explicit or implicit requirements when used under specified conditions [2], with a main focus on system quality, taking into account the quality in use, usually only with respect to traditional quality attributes, such as correctness, reliability and safety, rarely considering the synthesis of different quality attributes. For example, the quality model defined in ISO/IEC 25010 only considers the measurement of each attribute rather than performing an overall evaluation of software quality according to these measures [2]. Although the McCall quality model [3] and Boehm quality model [4] consider the synthesis of each quality attribute, they only combine each quality attribute value as a measure of quality through a simple weighted average. The software quality models mentioned were introduced at different times and are the products of the evolution of such models. For example, ISO/IEC 25010 is the most recent standardized model, which was introduced in 2010, whereas the McCall model was proposed in 1977, and the Boehm model was proposed in 1978. This evolution continues at present in a slightly different and more specialized direction. Software trustworthiness refers to the extent which the dynamic behavior and results of software systems meet people's expectations and provide continuous services when disturbed [5]. In addition to traditional quality attributes, the concept of software trustworthiness also attaches importance to new software attributes, such as survivability and controllability, as well as the synthesis of these attributes. Furthermore, software trustworthiness not only considers the objective quality of software but also the recognition of this objective quality in people's minds. In this paper, we use the above definition of software trustworthiness.

Software trustworthiness is a new concept developed on the basis of many attributes, such as correctness, reliability, security, timeliness, integrity, availability, predictability, survivability, controllability, etc. [5]. It can be characterized by many attributes [6–9], which are referred to as trustworthy attributes. Because we deal only with software trustworthiness herein, the trustworthy attributes considered in this paper are only non-functional requirements related to software trustworthiness. Attribute-based measures of software trustworthiness typically translate the quantification problem of software trustworthiness into the selection of trustworthy attributes, measures of trustworthy attributes, and the aggregation of trustworthy attribute values [7]. There are three ways to choose trustworthy attributes. Given that different software has different trustworthy requirements, the first method dynamically constructs a software trustworthy attribute model. The second method is to establish the software trustworthy index system in advance. The third class integrates the first two methods. Some of the trustworthy attributes are provided up front, whereas others are provided by the user on demand. Most existing trustworthy attribute measurement methods first decompose high-level attributes into low-level sub-attributes that are easy to measure, with sub-attribute values then synthesized using mathematical tools, such as regression analysis, principal component analysis, and factor analysis [7,10]. The essence of this method is to predict the trustworthiness of attributes according to the internal attribute values. Typical models for the synthesis of trustworthy attribute values include machine learning [11–15], axiomatic approaches [16–20], uncertainty theory [21–24], system testing [25], the social-to-software framework [26], user feedback [27], heuristic-systematic processing [28], the development framework [29], crowd wisdom [30], etc.

The judgment degrees of trustworthy attributes are assessed by each criterion. Therefore, software trustworthiness evaluation can be regarded as a multi-criteria decision-making (MCDM) problem that consists of a series of criteria. The number of software trustworthy attributes is equal to the number of criteria used to evaluate software trustworthiness. Most existing approaches in MCDM are composed of two phases: (1) the aggregation of the judgements with respect to all criteria and per decision alternative and (2) the rank ordering of the decision alternatives according to the aggregated judgments. In practical decision-making problems, the criteria are often interrelated. For example, high performance and low power consumption are a pair of contradictory criteria. However, most approaches do not refer to the aspect of an explicit modeling of relationships between criteria, which makes the optimal solution obtained through MCDM discounted in use.

In this paper, a software trustworthiness evaluation strategy via the relationships between criteria is proposed. First, a trustworthy attribute measurement method based on fuzzy theory is presented. Then, we refer to the method described in [31,32], and the quantitative relationships between criteria are described by cooperative and conflicting degrees between criteria. Thirdly, the symmetric substitutivity between criteria is formulated, and the cooperative degree between criteria is used as the approximation of the symmetric substitutivity between criteria. By means of symmetric substitutivity between criteria, the software trustworthiness measurement model established by axiomatic approaches is used to aggregate the trustworthy attribute degree. Finally, the candidate software products are sorted on the basis of the results of trustworthiness aggregation, and the optimal product is obtained from the alternative software products according to the sorting results.

There are three main contributions in this work: (1) A novel trustworthy attribute measurement method based on fuzzy theory is given, which can more reasonably convert the fuzzy decision about trustworthy attribute to the determined real number. (2) The formal definitions of the quantitative relationships between criteria and the symmetric substitutivity between criteria are proposed, and a link between these two definitions is established. (3) A software trustworthiness evaluation model via the relationships between criteria is presented, which can not only be applied for the ranking of candidate software on trustworthiness, but also for the trustworthiness measurement of candidate software. Meanwhile, we theoretically validate this model by proving that it complies with the properties introduced in [19], and empirically verify it through a case study.

The rest of this paper is organized as follows. Section 2 describes the related work. A trustworthy attribute measurement method based on fuzzy set is given in Section 3. Section 4 proposes a software trustworthiness evaluation strategy via relationships between criteria. A case study is introduced to show the effectiveness of the presented methods in Section 5. Discussion and limitations are presented in Section 6. The conclusions and future work come in the last section.

2. Related Work

In this section some typical software trustworthiness models are selected for a detailed introduction.

Machine learning is widely used to solve complex problems in engineering applications and scientific fields, and it is also utilized to evaluate the software trustworthiness. Yuan et al. propose a partition consistency measurement method for applying software trustworthiness measurement in dynamic behavior feature datasets [11]. The dynamic behavior feature datasets are generated while the software is running. This method compares the datasets with the static attribute feature datasets generated during software testing. Medeiros et al. provide a comprehensive experiment to investigate how to effectively use software metrics to distinguish vulnerable code units from non-vulnerable code units [12]. To conduct this, they use Random Forest, Extreme Boosting, Decision Tree, SVM Linear, and SVM Radial to extract vulnerability-related knowledge from software metrics collected from the source code of Mozilla Firefox, Linux Kernel, Apache HTTPd, Xen, and Glibc. Lv et al., present an automatic online assessment of trustworthiness of cyber-physical system [13]. An evaluation framework based on machine learning knowledge is established, and an online ranking algorithm is designed to realize online real-time analysis and evaluation. Xu et al. build a QoS prediction model. In this model, they combine neural networks with matrix factorization to perform non-linear collaborative filtering on the potential feature vectors of users and services [14]. They also have conducted numerous experiments in a large-scale real QoS datasets, and the experimental results demonstrate the effectiveness of their method. Tian et al. build a software trustworthiness evaluation model based on a behavior trajectory matrix [15]. Checkpoints are set in the software behavior trajectory, and binary code is introduced to represent the software behavior trajectory tree. The scene information about the checkpoint is obtained and applied to establish the behavior trajectory matrix, which is used to represent the behavior trajectory. The behavioral trajectory matrix is transformed into gray scale images to train a deep residual network to classify the software behavior.

To more rigorously measure software trustworthiness and theoretically verify the model, Tao et al. evaluate software trustworthiness using axiomatic methods, present the expected properties of software trustworthiness measures, and build a series of models that satisfy these properties [16–20].

There are other classic software trustworthiness measurement models. Muhammad et al. classify software trustworthiness using values assigned to test case results. The rating strategy includes imposed test strategy, completeness of system test execution, test iterations, test case priority, and test case results from each iteration [25]. Yang et al. present a social-to-software framework of software trustworthiness measurement [26]. They first introduce a generalized index loss to unify the identity evidence, basic standard or norms evidence, and ability evidence of trustworthiness results. Then different methods are proposed for the three parts of software trustworthiness. Xu establishes a data-driven trust measurement model on the basis of the perceptual sources [33]. Direct trust calculation is implemented by sensing the relationship between nodes' own data, and recommended trust calculation is realized by the relationship between neighboring nodes in the monitoring module. Deng et al. propose a software trustworthiness model based on evidence, which is called TDT [34]. The basic idea of building TDT is to distill the main characteristics into key components and continue distilling until the basic facts such as evidence are reached. TDT can be used as a communication means for different stakeholders to reach an agreement on the system attributes in the requirements analysis phase, and can be used for deductive reasoning to verify whether the system achieves credibility in the product verification phase. A real-time trust metric theory on the basis of the non-interference model is put forward by Zhang et al. [35]. In this theory, system calls are processed as atomic operations, and system call sequences are constructed as the actual behavior of the process. The theoretical expected behavior is calculated according to the mutual non-interference relationship between the corresponding security domains in the actual behavior. The trustworthiness of a process is evaluated by determining whether actual and theoretical expected behavior deviate. Wang et al. extract the evidence of software trustworthiness from the following aspects of process entity: behavior and product. They provide a software process trustworthiness model composed of 37 credibility principles, 182 process credibility evidence and 108 product credibility evidence, and they present a software process trustworthiness evaluation method based on this evidence [36,37].

3. Trustworthy Attribute Measurement Method Based on Fuzzy Set

Fuzzy set theory [38,39] provides a good idea for evaluation. Shi et al. apply the fuzzy set theory to evaluate the trustworthy attributes [40]. They first establish the mapping between the trustworthy attribute language variable and the triangular fuzzy number (TFN), then invite the evaluators to construct the fuzzy decision matrix, and finally measure the trustworthy attribute by defuzzification. This approach does not require measuring internal attributes and can model uncertainty or inherent imprecision of expert judgments. In this paper, a similar method is adopted to measure trustworthy attributes. Considering that the greater the software trustworthiness, the more difficult it is to improve the trustworthiness, and the higher the requirements for the trustworthy attribute value. The classification interval is unequal when constructing the mapping between the trustworthy attribute language variable and the triangular fuzzy number. The value interval increasing from the lowest level to the highest level is approximately reduced according to the gold ratio.

The trustworthy attribute is divided into five levels, namely, very low, low, middle, high and very high. The mapping relationship between trustworthy attribute level and TFN is shown in Table 1.

Table 1. Mapping relationship between trustworthy attribute level and TFN.

Trustworthy Attribute	TFN
Very Low (VL)	(0.0, 0.0, 4.5)
Low (L)	(4.5, 6.0, 7.0)
Middle (M)	(7.0, 8.0, 8.5)
High (H)	(8.5, 9.0, 9.5)
Very High (VH)	(9.5, 10.0, 10.0)

Given the set of possible alternatives $A = \{a_1, a_2, \cdots a_m\}$, the set of criteria $C = \{c_1, c_2, \cdots c_n\}$ and l evaluators, each evaluator rates the trustworthy attribute with the TFNs given in Table 1, and obtains the fuzzy decision matrix D^k as shown in the following:

$$D^k = \begin{array}{c} \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{array} \begin{array}{c} \begin{array}{cccc} c_1 & c_2 & \cdots & c_n \end{array} \\ \begin{bmatrix} e_{11}^k & e_{12}^k & \cdots & e_{1n}^k \\ e_{21}^k & e_{22}^k & \cdots & e_{2n}^k \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1}^k & e_{m2}^k & \cdots & e_{mn}^k \end{bmatrix} \end{array}$$

where $e_{ij}^k = (l_{ij}^k, m_{ij}^k, u_{ij}^k)$ is a TFN and expresses the fuzzy judgment rating of alternative a_i concerning the criterion c_j given by k-th evaluator. Suppose that each evaluator has the same importance, then the final fuzzy decision matrix D can be determined, which is expressed as:

$$D = \begin{array}{c} \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{array} \begin{array}{c} \begin{array}{cccc} c_1 & c_2 & \cdots & c_n \end{array} \\ \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m1} & e_{m2} & \cdots & e_{mn} \end{bmatrix} \end{array}$$

where $e_{ij} = (l_{ij}, m_{ij}, u_{ij}) = (\sum_{k=1}^{l} l_{ij}^k/l, \sum_{k=1}^{l} m_{ij}^k/l, \sum_{k=1}^{l} u_{ij}^k/l)$.

In order to integrate the trustworthy attribute values with axiomatic approaches in the future, the fuzzy number is needed to convert to the determined real number through defuzzification technology. The graded mean integration representation method presented by Chen and Hsieh [41] is used in this paper. Denote the defuzzification value of $e_{ij} = (l_{ij}, m_{ij}, u_{ij})$ as $\mu_{c_j}(a_i)$, and $\mu_{c_j}(a_i)$ is obtained by Equation (1).

$$\mu_{c_j}(a_i) = \frac{l_{ij} + 4m_{ij} + u_{ij}}{6} \tag{1}$$

$\mu_{c_j}(a_i)$ is the degree to which the alternative a_i meets criterion c_j. Since $0 \le l_{ij}, m_{ij}, u_{ij} \le 10$, then it follows that $0 \le \mu_{c_j}(a_i) \le 10$.

4. Software Trustworthiness Evaluation Based on Relationships between Criteria

For a given alternative, intuitively speaking, the two criteria are contradictory if the increase (or decrease) in the satisfaction degree of one criterion will lead to the decrease (or increase) in the satisfaction degree of the other criterion. The two criteria promote each other if the increase (or decrease) in the satisfaction degree of one criterion will lead to the increase (or decrease) in the satisfaction degree of the other criterion. The two criteria are independent of each other if the change of the satisfaction degree of one criterion will not lead to the change of the satisfaction degree of the other criterion. On the basis of the above description, the formal definitions of conflicting, cooperative and irrelevant pairs are given, as shown in Definition 1.

Before providing this definition, the notations used in this paper are first presented as follows: Denote the set of possible alternatives as $A = \{a_1, a_2, \cdots a_m\}$ and the set of criteria as $C = \{c_1, c_2, \cdots c_n\}$. Suppose $\mu_{c_j}(a_i)$ is the degree to which the alternative a_i meets criterion c_j, which satisfies $0 \le \mu_{c_j}(a_i) \le 10$.

Definition 1 (Conflicting, cooperative and irrelevant alternative pairs [31,32])**.**

Suppose that cand $c'$are the two criteria, and $A = \{a_1, a_2, \cdots a_n\}$ is a set of alternatives, $\forall a_i, a_j \in A, i \ne j$. A set of conflicting alternative pairs about c and c' is defined as,

$$CF(c, c') = \{(a_i, a_j) | (\mu_c(a_i) - \mu_c(a_j)) \times (\mu_{c'}(a_i) - \mu_{c'}(a_j)) < 0\}$$

A set of cooperative alternative pairs about candc′is defined as,

$$CP(c,c\prime) = \left\{(a_i,a_j)\middle|(\mu_c(a_i) - \mu_c(a_j)) \times (\mu_{c\prime}(a_i) - \mu_{c\prime}(a_j)) > 0\right\}$$

A set of irrelevant alternative pairs about candc′is defined as,

$$IR(c,c') = \left\{(a_i,a_j)\middle|(\mu_c(a_i) - \mu_c(a_j)) \times (\mu_{c\prime}(a_i) - \mu_{c\prime}(a_j)) = 0\right\}$$

Hence, for the two criteria c and c', the set of pairs of alternatives $A_P(c,c') = \left\{(a_i,a_j)\middle|\forall a_i,a_j \in A\right\}$ can be divided into three categories: conflicting, cooperative and irrelevant, and it is easy to obtain that

$$A_P(c,c') = CF(c,c') \cup CP(c,c') \cup IR(c,c')$$

and

$$CF(c,c') \cap CP(c,c') \cap IR(c,c') = \varnothing$$

Based on Definition 1, we present the conflicting and cooperative degrees between two criteria, as shown in Definition 2.

Definition 2 (Conflicting and cooperative degrees between two criteria)**.**
Suppose that c and c′ are the two criteria, and $A_P(c,c') = \left\{(a_i,a_j)\middle|\forall a_i,a_j \in A\right\}$ *is the set of pairs of alternatives. Let* $CF(c,c')$ *denote the set of conflicting pairs about c and c′, and* $CP(c,c')$ *represent the set of cooperative pairs about c and c′. The conflicting degree between the two criteria c and c′, is defined as:*

$$cf(c,c') = \frac{\sum\limits_{(a_i,a_j)\in CF(c,c')}\left(\left|\frac{\mu_c(a_i)-\mu_c(a_j)}{\mu_c(a_i)}\right| + \left|\frac{\mu_{c'}(a_i)-\mu_{c'}(a_j)}{\mu_{c'}(a_i)}\right|\right)}{\sum\limits_{(a_i,a_j)\in A_p(c,c')}\left(\left|\frac{\mu_c(a_i)-\mu_c(a_j)}{\mu_c(a_i)}\right| + \left|\frac{\mu_{c'}(a_i)-\mu_{c'}(a_j)}{\mu_{c'}(a_i)}\right|\right)}$$

The cooperative degree between the two criteria, cand c′, is defined as:

$$cp(c,c') = \frac{\sum\limits_{(a_i,a_j)\in CP(c,c')}\left(\left|\frac{\mu_c(a_i)-\mu_c(a_j)}{\mu_c(a_i)}\right| + \left|\frac{\mu_{c'}(a_i)-\mu_{c'}(a_j)}{\mu_{c'}(a_i)}\right|\right)}{\sum\limits_{(a_i,a_j)\in A_p(c,c')}\left(\left|\frac{\mu_c(a_i)-\mu_c(a_j)}{\mu_c(a_i)}\right| + \left|\frac{\mu_{c'}(a_i)-\mu_{c'}(a_j)}{\mu_{c'}(a_i)}\right|\right)}$$

In the following, the set of criteria $C = \{c_1, c_2, \cdots c_n\}$ is divided based on the cooperative degrees. Among all the criteria, the two criteria with the highest cooperative degree are selected as a group, and then the two criteria with the highest cooperative degree are selected from the remaining criteria as a group, and so on until the division is completed. Suppose n is an even number, and the division result is $\{c_1, c_2\}$ as a group, $\{c_3, c_4\}$ as a group, … , $\{c_{n-1}, c_n\}$ as a group. When n is odd, the division results are similar, narrowly making c_n itself a group.

The symmetric substitutivity between criteria is used to describe the quantitative change relationships between two criteria when the total criterion remains unchanged and only these two criteria values are changed, which is defined in Definition 3.

Definition 3 (Symmetric substitutivity between criteria)**.**

For a given alternative a, assume that c_i and c_j are the two criteria, $\mu_{c_j}(a)$ is the degree to which the alternative a meets criterion c_j and T is a differentiable function about $\mu_{c_j}(a)$, the symmetric substitutivity between criteria c_i and c_j is defined as Equation (2).

$$\sigma_{ij} = \frac{d\left(\mu_{c_i}(a)/\mu_{c_j}(a)\right)}{d(h_{ij})} \times \frac{h_{ij}}{\mu_{c_i}(a)/\mu_{c_j}(a)} \tag{2}$$

where

$$h_{ij} = -\frac{\partial T/\partial \mu_{c_j}(a)}{\partial T/\partial \mu_{c_i}(a)} = \frac{d\mu_{c_i}(a)}{d\mu_{c_j}(a)}.$$

Axiomatic approaches formalize the empirical understanding of software attributes through defining ideal metric properties. They can offer precise terms for the software attributes' quantification. They have been utilized to assess software trustworthiness [16–20,42]. In this paper, a simplified software trustworthiness metric model presented in [42] is applied to aggregate $\mu_{c_j}(a)$.

Definition 4 (Simplified software trustworthiness metric model given in [42])**.**

The simplified software trustworthiness metric model used in [41] *is shown as Equation (3).*

$$\begin{aligned} T = {} & \left(\left(\tfrac{\alpha_1}{\alpha_1+\alpha_2}(\mu_{c_1}(a))^{-\rho_{12}} + \tfrac{\alpha_2}{\alpha_1+\alpha_2}(\mu_{c_2}(a))^{-\rho_{12}}\right)^{-\frac{1}{\rho_{12}}}\right)^{\alpha_1+\alpha_2} \times \left(\left(\tfrac{\alpha_3}{\alpha_3+\alpha_4}(\mu_{c_3}(a))^{-\rho_{34}} + \tfrac{\alpha_4}{\alpha_3+\alpha_4}(\mu_{c_4}(a))^{-\rho_{34}}\right)^{-\frac{1}{\rho_{34}}}\right)^{\alpha_3+\alpha_4} \\ & \times \cdots \times \left(\left(\tfrac{\alpha_{n-1}}{\alpha_{n-1}+\alpha_n}(\mu_{c_{n-1}}(a))^{-\rho_{n-1,n}} + \tfrac{\alpha_{n-1}}{\alpha_{n-1}+\alpha_n}(\mu_{c_n}(a))^{-\rho_{n-1,n}}\right)^{-\frac{1}{\rho_{n-1,n}}}\right)^{\alpha_{n-1}+\alpha_n} \end{aligned} \tag{3}$$

where

(1) *a is a possible alternative;*
(2) *n is the number of criteria;*
(3) *$c_j(1 \le j \le n)$ are the criteria;*
(4) *$\mu_{c_j}(a)$ is the degree to which the alternative a satisfies criterion c_j, such that $0 \le \mu_{c_j}(a) \le 10$;*
(5) *α_j represents the weight of criterion c_j with $0 \le \alpha_j \le 1$ and $\sum_{j=1}^{m} \alpha_j = 1$;*
(6) *$\rho_{i,i+1}(1 \le i \le n-1)$ are parameters related to the symmetric substitutivity between the criteria c_i and c_{i+1};*
(7) *T is the software trustworthiness measure function of $\mu_{c_1}(a), \ldots, \mu_{c_n}(a)$.*

By Equation (2), the symmetric substitutivity between criteria c_i and c_j can be determined as follows:

$$\sigma_{12} = \tfrac{1}{1+\rho_{12}}$$

$$\sigma_{34} = \tfrac{1}{1+\rho_{34}}$$

$$\cdots\cdots$$

$$\sigma_{n-1,n} = \tfrac{1}{1+\rho_{n-1,n}}\,.$$

Using the cooperative degree between criteria c_i and c_j as an approximation of the symmetric substitutivity between criteria c_i and c_j, we can determine

$$\sigma_{12} = \tfrac{1}{1+\rho_{12}} = cp(c_1, c_2)$$
$$\sigma_{34} = \tfrac{1}{1+\rho_{34}} = cp(c_3, c_4)$$
$$\cdots\cdots$$
$$\sigma_{n-1,n} = \tfrac{1}{1+\rho_{n-1,n}} = cp(c_{n-1}, c_n).$$

It follows that

$$\rho_{12} = \tfrac{1}{cp(c_1,c_2)} - 1$$
$$\rho_{34} = \tfrac{1}{cp(c_3,c_4)} - 1$$
$$\cdots\cdots$$
$$\rho_{n-1,n} = \tfrac{1}{cp(c_{n-1},c_n)} - 1.$$

Then the following software trustworthiness metric model based on relationships between criteria can be obtained.

Definition 5 (Software trustworthiness metric model based on relationships between criteria)**.** *Software trustworthiness metric model based on relationships between criteria is defined as Equation (*4*).*

$$\begin{gathered} T = \left(\left(\tfrac{\alpha_1}{\alpha_1+\alpha_2} (\mu_{c_1}(a))^{-(\frac{1}{cp(c_1,c_2)}-1)} + \tfrac{\alpha_2}{\alpha_1+\alpha_2} (\mu_{c_2}(a))^{-(\frac{1}{cp(c_1,c_2)}-1)} \right)^{-\frac{1}{\frac{1}{cp(c_1,c_2)}-1}} \right)^{\alpha_1+\alpha_2} \times \\ \left(\left(\tfrac{\alpha_3}{\alpha_3+\alpha_4} (\mu_{c_3}(a))^{-(\frac{1}{cp(c_3,c_4)}-1)} + \tfrac{\alpha_4}{\alpha_3+\alpha_4} (\mu_{c_4}(a))^{-(\frac{1}{cp(c_3,c_4)}-1)} \right)^{-\frac{1}{\frac{1}{cp(c_3,c_4)}-1}} \right)^{\alpha_3+\alpha_4} \times \\ \cdots\cdots \times \\ \left(\left(\tfrac{\alpha_{n-1}}{\alpha_{n-1}+\alpha_n} (\mu_{c_{n-1}}(a))^{-(\frac{1}{cp(c_{n-1},c_n)}-1)} + \tfrac{\alpha_{n-1}}{\alpha_{n-1}+\alpha_n} (\mu_{c_n}(a))^{-(\frac{1}{cp(c_{n-1},c_n)}-1)} \right)^{-\frac{1}{\frac{1}{cp(c_{n-1},c_n)}-1}} \right)^{\alpha_{n-1}+\alpha_n} \end{gathered} \tag{4}$$

where

1. *a is a possible alternative;*
2. *n is the number of criteria;*
3. *$c_j (1 \leq j \leq n)$ are the criteria;*
4. *$\mu_{c_j}(a)$ is the degree to which the alternative a satisfies criterion c_j, such that $0 \leq \mu_{c_j}(a) \leq 10$;*
5. *α_j represents the weight of criterion c_j with $0 \leq \alpha_j \leq 1$ and $\sum_{j=1}^{m} \alpha_j = 1$;*
6. *$cp(c_i, c_{i+1}) (1 \leq i \leq n-1)$ are cooperative degrees between criteria c_i and c_{i+1};*
7. *T is the software trustworthiness measure function of $\mu_{c_1}(a), \ldots, \mu_{c_n}(a)$.*

We once presented the expected properties of software trustworthiness measure, including monotonicity, acceleration, sensitivity and substitutivity [16]. Here we prove that the software trustworthiness metric model based on relationships between criteria conforms to these four properties. For convenience, in the following of this paper, let

$$b = \frac{\alpha_i}{\alpha_i + \alpha_{i+1}} (\mu_{c_i}(a))^{-(\frac{1}{cp(c_i,c_{i+1})}-1)} + \frac{\alpha_{i+1}}{\alpha_i + \alpha_{i+1}} (\mu_{c_{i+1}}(a))^{-(\frac{1}{cp(c_i,c_{i+1})}-1)}$$

Proposition 1. *T satisfies monotonicity, i.e., $\frac{\partial T}{\partial \mu_{c_i}(a)} \geq 0, 1 \leq i \leq n$.*

Proof of Proposition 1. Since

$$\frac{\partial T}{\partial \mu_{c_i}(a)} = \alpha_i T b^{-1} \mu_{c_i}(a)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)-1};\ 1 \le i \le n$$

and

$$\begin{cases} 0 \le cp(c_i, c_{i+1}) \le 1, & 1 \le i \le n-1 \\ 0 \le \mu_{c_i}(a) \le 10, & 1 \le i \le n \\ 0 \le \alpha_i \le 1, & 1 \le i \le n \end{cases}$$

then we can get

$$\frac{\partial T}{\partial \mu_{c_i}(a)} \ge 0,\ 1 \le i \le n$$

Therefore, T complies with monotonicity. □

Proposition 2. *Acceleration holds for T, i.e.,* $\frac{\partial^2 T}{\partial \mu_{c_i}(a)^2} \le 0, 1 \le i \le n.$

Proof of Proposition 2. By calculating the second derivative of T with respect to each $\mu_{c_i}(a)$ $(1 \le i \le n)$, we can obtain

$$\frac{\partial^2 T}{\partial \mu_{c_i}(a)^2} = \alpha_i T b^{-1} \mu_{c_i}(a)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)-2} \times$$
$$\left(\alpha_i b^{-1} \mu_{c_i}(a)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)} + \frac{\alpha_i}{\alpha_i+\alpha_{i+1}} \left(\frac{1}{cp(c_i,c_{i+1})} - 1\right) b^{-1} \mu_{c_i}(a)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)-1} - \frac{1}{cp(c_i,c_{i+1})}\right)$$

Due to

$$\alpha_i b^{-1} \mu_{c_i}(a)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)} = \frac{\alpha_i \mu_{c_i}(a)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)}}{\frac{\alpha_i}{\alpha_i+\alpha_{i+1}} \left(\mu_{c_i}(a)\right)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)} + \frac{\alpha_{i+1}}{\alpha_i+\alpha_{i+1}} \left(\mu_{c_{i+1}}(a)\right)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)}}$$
$$= (\alpha_i + \alpha_{i+1}) \frac{\alpha_i \mu_{c_i}(a)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)}}{\alpha_i \left(\mu_{c_i}(a)\right)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)} + \alpha_{i+1} \left(\mu_{c_{i+1}}(a)\right)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)}} \le 1$$

and

$$\frac{\alpha_i}{\alpha_i+\alpha_{i+1}} \left(\frac{1}{cp(c_i,c_{i+1})} - 1\right) b^{-1} \mu_{c_i}(a)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)-1}$$
$$= \left(\frac{1}{cp(c_i,c_{i+1})} - 1\right) \frac{\frac{\alpha_i}{\alpha_i+\alpha_{i+1}} \mu_{c_i}(a)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)-1}}{\frac{\alpha_i}{\alpha_i+\alpha_{i+1}} \left(\mu_{c_i}(a)\right)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)} + \frac{\alpha_{i+1}}{\alpha_i+\alpha_{i+1}} \left(\mu_{c_{i+1}}(a)\right)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)}}$$
$$\le \left(\frac{1}{cp(c_i,c_{i+1})} - 1\right)$$

it follows that

$$\left(\alpha_i b^{-1} \mu_{c_i}(a)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)} + \frac{\alpha_i}{\alpha_i + \alpha_{i+1}} \left(\frac{1}{cp(c_i, c_{i+1})} - 1\right) b^{-1} \mu_{c_i}(a)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)-1} - \frac{1}{cp(c_i, c_{i+1})}\right) \le 0$$

Notice that

$$\begin{cases} 0 \le cp(c_i, c_{i+1}) \le 1, & 1 \le i \le n-1 \\ 0 \le \mu_{c_i}(a) \le 10, & 1 \le i \le n \\ 0 \le \alpha_i \le 1, & 1 \le i \le n \end{cases}$$

Therefore

$$\alpha_i T b^{-1} \mu_{c_i}(a)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)-2} \ge 0$$

Then, for $1 \leq i \leq n$, we have

$$\frac{\partial^2 T}{\partial \mu_{c_i}(a)^2} \leq 0$$

□

Proposition 3. *T meets sensitivity.*

Proof of Proposition 3. By computing, it is easy to obtain that for $1 \leq i \leq n$,

$$\frac{\partial T}{\partial \mu_{c_i}(a)} \frac{\mu_{c_i}(a)}{T} = \alpha_i T b^{-1} \mu_{c_i}(a)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)-1} \frac{\mu_{c_i}(a)}{T} = \alpha_i b^{-1} \mu_{c_i}(a)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)}$$

Because of

$$\begin{cases} 0 \leq cp(c_i, c_{i+1}) \leq 1, & 1 \leq i \leq n-1 \\ 0 \leq \mu_{c_i}(a) \leq 10, & 1 \leq i \leq n \\ 0 \leq \alpha_i \leq 1, & 1 \leq i \leq n \end{cases}$$

it follows that

$$\frac{\partial T}{\partial \mu_{c_i}(a)} \frac{\mu_{c_i}(a)}{T} \geq 0$$

Then we obtain the conclusion that T is sensitive to $\mu_{c_i}(a)$. □

Proposition 4. *T complies with substitutivity.*

Proof of Proposition 4. Because this paper only focuses on the substitutivity of criteria within the same group, therefore only the substitutivity between criteria within the same group is calculated here.

Observe that for $1 \leq i \leq n-1$

$$h_{i,i+1} = \frac{\alpha_{i+1} \left(\mu_{c_{i+1}}(a)\right)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)-1}}{\alpha_i \left(\mu_{c_i}(a)\right)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)-1}}$$

Then the substitutivity between criteria c_i *and* c_{i+1} which belong to the i-th group can be derived as

$$\sigma_{i,i+1} = \frac{d\left(\mu_{c_i}(a)/\mu_{c_{i+1}}(a)\right)}{d(h_{i,i+1})} \times \frac{h_{i,i+1}}{\mu_{c_i}(a)/\mu_{c_{i+1}}(a)}$$

$$= \frac{\left(\mu_{c_{i+1}}(a)\right)^{-1} d\mu_{c_i}(a) - \mu_{c_i}(a)\left(\mu_{c_{i+1}}(a)\right)^{-2} d\mu_{c_{i+1}}(a)}{\frac{\alpha_{i+1}}{\alpha_i} \frac{1}{cp(c_i,c_{i+1})} \left(\mu_{c_i}(a)\right)^{(\frac{1}{cp(c_i,c_{i+1})}-1)} \left(\mu_{c_{i+1}}(a)\right)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)} \left(\left(\mu_{c_{i+1}}(a)\right)^{-1} d\mu_{c_i}(a) - \mu_{c_i}(a)\left(\mu_{c_{i+1}}(a)\right)^{-2} d\mu_{c_{i+1}}(a)\right)} \times$$

$$\frac{\frac{\alpha_{i+1}}{\alpha_i} \left(\mu_{c_i}(a)\right)^{(\frac{1}{cp(c_i,c_{i+1})}-1)+1} \left(\mu_{c_{i+1}}(a)\right)^{-(\frac{1}{cp(c_i,c_{i+1})}-1)-1}}{\mu_{c_i}(a)\left(\mu_{c_{i+1}}(a)\right)^{-1}} = cp(c_i, c_{i+1}).$$

The proposition follows immediately from what we have proved. □

For the given set of possible alternatives $A = \{a_1, a_2, \cdots, a_m\}$, the set of criteria $C = \{c_1, c_2, \cdots c_n\}$, the set of the weights of the criteria $\{\alpha_1, \alpha_2, \cdots, \alpha_n\}$, and the decision matrix about $\mu_{c_j}(a_i)$ $(1 \leq i \leq m, 1 \leq j \leq n)$, $CP(c_i, c_j)$ are first calculated according to Definition 1 and $cp(c_i, c_j)$ are computed based on $CP(c_i, c_j)$ and Definition 2. Then, the criteria are grouped through $cp(c_i, c_j)$, the two criteria with the highest cooperative degree are grouped together, and then the two criteria with the highest cooperative degree of the remaining criteria are grouped together, and so on until the division is completed.

Assume that c_i and c_j are in the same group, then $\rho_{ij} = \frac{1}{cp(c_i,c_j)} - 1$. Considering that when $cp(c_i, c_j) = 1$, the corresponding ρ_{ij} is 0, and the contribution of $\mu_{c_i}(a)$, $\mu_{c_j}(a)$ to the trustworthiness of the candidate alternative a cannot be reflected, so simple preprocessing is performed. When $cp(c_i, c_j) = 1$, the value of the corresponding ρ_{ij} is set to 0.01. Finally, the metric model given in Definition 5 is used to evaluate each alternative in A, and the measurement results are arranged in descending order. The alternative corresponding to the first measurement result is the optimal alternative.

Without losing generality, assume that n is an even number. The algorithm for software trustworthiness evaluation based on relationships between criteria is given in Algorithm 1.

Algorithm 1 Algorithm for software trustworthiness evaluation based on relationships between criteria: for the given set of possible alternatives $A = \{a_1, a_2, \cdots, a_m\}$, the set of criteria $C = \{c_1, c_2, \cdots c_n\}$, the set of the weights of the criteria $\{\alpha_1, \alpha_2, \cdots, \alpha_n\}$, and the decision matrix about $\mu_{c_j}(a_i)$ $(1 \le i \le m, 1 \le j \le n)$, output the optimal alternative of $A = \{a_1, a_2, \cdots, a_m\}$.

```
Input: $A = \{a_1, a_2, \cdots, a_m\}, C = \{c_1, c_2, \cdots c_n\}, \{\alpha_1, \alpha_2, \cdots, \alpha_n\}, \mu_{c_j}(a_i)\ (1 \le i \le m, 1 \le j \le n)$
Output: the optimal alternative of $A = \{a_1, a_2, \cdots, a_m\}$
Let $M = \{(i,j) | 1 \le i < j \le n\}$;
for $(i,j) \in M$ do
    Calculate $CP(c_i, c_j)$ according to Definition 1;
end for
for $(i,j) \in M$ do
    Compute $cp(c_i, c_j)$based on $CP(c_i, c_j)$ and Definition 2;
end for
while $M \neq \varnothing$ do
    $cp = 0$;
    for $(i,j) \in M$ do
        if $cp(c_i, c_j) \ge cp$ then
            $cp = cp(c_i, c_j)$;
        end if
    end for
    $N = \varnothing$;
    for $(i,j) \in M$ do
        if $cp(c_i, c_j) == cp$ then
            $N = N \cup \{(i,j)\}$;
        end if
    end for
    Randomly select $(i,j) \in N$ and make $c_i$ and $c_j$ a group;
    if $cp = 1$ then
        $\rho_{ij} = 0.01$;
    else
        $\rho_{ij} = \frac{1}{cp} - 1$;
    end if
    $M = M - \{(i,j)\}$;
end while
According to the grouping results, substitute $\rho_{ij}$ into Equation (3) and obtain Equation (4);
for $i \in \{1, 2, \ldots, m\}$ do
    for $j \in \{1, 2, \ldots, n\}$ do
        Substitute $\mu_{c_j}(a_i)$and $w_j$ into Equation (4);
    end for
    Obtain the software trustworthiness measurement result $T(a_i)$ of $a_i$;
end for
Sort $\{T(a_1), T(a_2), \cdots, T(a_m)\}$ in descending order using the quick sorting algorithm;
return the alternative corresponding to the first measurement result.
```

Theorem 1. *The time complexity of Algorithm* 1 *is* $O(n^4 + mn)$.

Proof of Theorem 1. Steps 2–4 are a for loop, and they are applied to calculate the set of cooperative alternative pairs according to Definition 1. The number of loops is $|M|$. Since $|M| = n(n-1)/2$, the time complexity of Steps 2–4 is $O(n^2)$. Steps 5–7 are also a for loop, which are used to compute the set of cooperative degrees based on the results of Steps 2–4 and Definition 2. The number of loops is also $|M|$. Therefore, the time complexity of Steps 5–7 is also $O(n^2)$.

Steps 8–28 are a double nested loop, which are utilized to group the set of criteria $C = \{c_1, c_2, \cdots c_n\}$ and compute ρ_{ij}. The number of loops in first while loop is $|M|$. Steps 10–14 are the first for loop in the second loop, and are used to find the maximum cooperative degree in the set $\{cp(c_i, c_j)|(i,j) \in M\}$, and the number of loops is $|M|$. Steps 16–20 are the second for loop in the second loop, constructing the set composed of subscripts of maximum cooperative degree in the set $\{cp(c_i, c_j)|(i,j) \in M\}$, and the number of loops is also $|M|$. Step 21 randomly selects (i,j) from N and makes c_i and c_j a group. Steps 22–26 are an if-else conditional statement, computing ρ_{ij}, and Step 27 is used to update the loop variable of the while loop. It is easy to determine that the time complexity of Steps 21–27 is $O(1)$ and the time complexity of Steps 9–27 is $2|M| + O(1)$. Therefore, the time complexity of Steps 8–28 is $M(2|M| + O(1))$. Notice that $|M| = n(n-1)/2$, and it follows that the time complexity of Steps 8–28 is $O(n^4)$.

Steps 30–35 are also a double nested loop, calculating the software trustworthiness measurement results $\{T(a_1), T(a_2), \cdots, T(a_m)\}$. The number of loops in the first loop is m, and in the second loop is n. Consequently, we infer that the time complexity of Steps 30–35 is $O(mn)$. The software trustworthiness measurement results $\{T(a_1), T(a_2), \cdots, T(a_m)\}$ are sorted in descending order using the quick sorting algorithm in Step 36, and it takes $O(n \log n)$.

In summary, we can obtain the time complexity of Algorithm 1 is $O(n^4 + mn)$. □

5. Case Study

With the development of enterprise informatization, Product Lifecycle Management (PLM) software is becoming more and more important to improve the informatization level and core competitiveness of enterprises. Reference [40] presents a software trustworthiness evaluation approach based on combination weights and improved TOPSIS methods, and this approach is applied to evaluate the candidate PLM software trustworthiness for an aircraft equipment manufacturer. The candidate PLM software set consists of three PLM software, and the criterion set is composed of functionality, learnability, operability, co-existence, maintainability and portability. In this section, the method proposed in this paper is used to evaluate this case. Denote the three candidate PLM software as a_1, a_2,a_3 in turn, and functionality, learnability, operability, co-existence, maintainability, and portability as c_1, c_2, c_3, c_4, c_5, c_6. The combination weights of these six criteria α_1, α_2, α_3, α_4, α_5, α_6 obtained through the combination weighting method established in [40] are given in Table 2. They are the weighted sum of the objective and subjective weights, where the objective weights are calculated by entropy weighting method and the subjective weights are determined by FAHP method.

Table 2. Combination weights of six criteria.

α_1	α_2	α_3	α_4	α_5	α_6
0.239	0.0799	0.1164	0.1997	0.2443	0.1206

Four exports D_1, D_2, D_3, D_4 are invited to give the fuzzy decision matrix about a_1, a_2,a_3 [36], as given in Table 3.

Table 3. Fuzzy decision matrix.

c_1				c_2				c_3				c_4				c_5				c_6				
D_1	D_2	D_3	D_4	D_1	D_2	D_3	D_4	D_1	D_2	D_3	D_4	D_1	D_2	D_3	D_4	D_1	D_2	D_3	D_4	D_1	D_2	D_3	D_4	
a_1	M	M	M	M	H	M	M	H	H	VH	VH	H	H	M	H	H	H	M	M	L	M	M	M	H
a_2	H	H	H	H	M	M	L	H	M	H	H	M	M	M	M	M	VH	VH	H	H	H	H	H	VH
a_3	VH	VH	VH	VH	VH	H	H	VH	M	H	H	L	M	L	M	H	H	H	M	M	M	L	L	H

Based on the mapping relationship between trustworthy attribute level and TFN defined in Table 1, the exports' fuzzy decision can be transferred to the corresponding fuzzy numbers, and the integrated fuzzy decision matrix using TFNs can be obtained, as presented in Table 4.

Table 4. Integrated fuzzy decision matrix.

	c_1	c_2	c_3	c_4	c_5	c_6
a_1	(7.000, 8.000, 8.5000)	(7.750, 8.500, 9.000)	(9.000, 9.500, 9.750)	(8.125, 8.750, 9.250)	(6.750, 7.750, 8.375)	(7.375, 8.250, 8.750)
a_2	(8.500, 9.000, 9.5000)	(6.750, 7.750, 8.375)	(7.750, 8.500, 9.000)	(9.000, 9.500, 9.750)	(9.000, 9.500, 9.750)	(8.750, 9.250, 9.625)
a_3	(9.500, 10.000, 10, 00)	(9.000, 9.500, 9.750)	(7.125, 8.000, 8.625)	(6.750, 7.750, 8.375)	(7.250, 8.500, 9.000)	(6.125, 7.250, 8.000)

Through defuzzying the integrated decision matrix with Equation (1), the decision matrix about $\mu_{c_j}(a_i)$ can be established as demonstrated in Table 5, where each element in the matrix is a real number between 0 and 10.

Table 5. Decision matrix.

$\mu_{c_j}(a_i)$	c_1	c_2	c_3	c_4	c_5	c_6
a_1	7.9167	8.4583	9.4583	8.7292	7.6875	8.1875
a_2	9.0000	7.6875	8.4583	9.4583	9.4583	9.2292
a_3	9.9167	9.4583	7.9583	7.6875	8.3750	7.1875

For any of the two criteria c_i and c_j $(1 \leq i < j \leq 6)$, according to Definition 1 and Table 5, calculate the set of cooperative alternative pairs, and the calculation results are shown in Equation (5).

$$\begin{aligned}
&CP(c_1, c_2) = \{(a_1, a_3), (a_2, a_3)\} \\
&CP(c_1, c_3) = \varnothing \\
&CP(c_1, c_4) = \{(a_1, a_2)\} \\
&CP(c_1, c_5) = \{(a_1, a_2), (a_1, a_3)\} \\
&CP(c_1, c_6) = \{(a_1, a_2)\} \\
&CP(c_2, c_3) = \{(a_1, a_2)\} \\
&CP(c_2, c_4) = \varnothing \\
&CP(c_2, c_5) = \{(a_1, a_3)\} \\
&CP(c_2, c_6) = \varnothing \\
&CP(c_3, c_4) = \{(a_1, a_3), (a_2, a_3)\} \\
&CP(c_3, c_5) = \{(a_2, a_3)\} \\
&CP(c_3, c_6) = \{(a_1, a_3), (a_2, a_3)\} \\
&CP(c_4, c_5) = \{(a_1, a_2), (a_2, a_3)\} \\
&CP(c_4, c_6) = \{(a_1, a_2), (a_1, a_3), (a_2, a_3)\} \\
&CP(c_5, c_6) = \{(a_1, a_2), (a_2, a_3)\}
\end{aligned} \tag{5}$$

For any of the two criteria c_i and c_j $(1 \leq i < j \leq 6)$, according to Definition 2 and the above calculation results, the cooperative degrees $cp(c_i, c_j)(1 \leq i < j \leq 6)$ can be determined, as demonstrated in Table 6.

Table 6. Cooperative degrees between criteria.

$\boldsymbol{cp(c_i,c_j)}$	c_2	c_3	c_4	c_5	c_6
c_1	0.7551	0	0.2500	0.7662	0.2745
c_2		0.2580	0	0.2376	0
c_3			0.7348	0.2292	0.7066
c_4				0.7468	1
c_5					0.7662

It can be seen from Table 6 that the maximum value of cp is $cp(c_4, c_6)$, whose value is 1. Therefore, c_4 and c_6 are selected as the first group. In the remaining criterion, the maximum value of cp is $cp(c_1, c_5) = 0.7662$, and c_1 and c_5 are chosen as the second group. Finally, c_2 and c_3 are taken as the third group. According to the grouping results, the weights in Table 2 and Definition 5, Equation (6) can be obtained. It should be noted that because $cp(c_4, c_6) = 1$, ρ_{46} is set to 0.01.

$$
\begin{aligned}
T = {} & \left(\left(\frac{0.2390}{0.2390+0.2443}(\mu_{c_1}(a))^{-(\frac{1}{0.7662}-1)} + \frac{0.2443}{0.2390+0.2443}(\mu_{c_5}(a))^{-(\frac{1}{0.7662}-1)}\right)^{-\frac{1}{\frac{1}{0.7662}-1}}\right)^{0.2390+0.2443} \times \\
& \left(\left(\frac{0.1997}{0.1997+0.1206}(\mu_{c_4}(a))^{-0.01} + \frac{0.1206}{0.1997+0.1206}(\mu_{c_6}(a))^{-0.01}\right)^{-\frac{1}{0.01}}\right)^{0.1997+0.1206} \times \\
& \left(\left(\frac{0.0799}{0.0799+0.1164}(\mu_{c_2}(a))^{-(\frac{1}{0.2580}-1)} + \frac{0.1164}{0.0799+0.1164}(\mu_{c_3}(a))^{-(\frac{1}{0.2580}-1)}\right)^{-\frac{1}{\frac{1}{0.2580}-1}}\right)^{0.0799+0.1164}
\end{aligned}
\tag{6}
$$

Substituting the data in Table 5 into Equation (6), it follows that

$$
\begin{aligned}
T(a_1) &= 8.2505 \\
T(a_2) &= 9.0394 \\
T(a_3) &= 8.4248
\end{aligned}
$$

Thus, among the three candidate PLM software, a_2 is the optimal software, and the result obtained is consistent with that in [40].

6. Discussion and Limitations

Software trustworthiness is a concept related to human cognition, and uncertainty theory can model human subjectivity well. Therefore, they are often used in software trustworthiness measurements. However, most of the existing methods use uncertainty theory to calculate the weight of trustworthy attributes. For example, references [21] and [22] both use fuzzy comprehensive evaluation methods to evaluate software trustworthiness, but they differ from the methods of selecting trustworthy attribute sets and computing the weights of trustworthy attributes. In reference [21], experts are invited to establish the trustworthy attribute set, and the weights of trustworthy attributes are computed based on the information entropy. In reference [22], the author's trustworthy attribute set is determined in advance, and the weights of trustworthy attributes are obtained by using rough set theory and expert opinion. Gao et al. present a new weight distribution method by combining fuzzy analytic hierarchy process with standard importance index correlation, and establish a component-based software trustworthiness measurement model according to the four component composition structures on the basis of their weight distribution method [23]. Shi et al. also provide a calculation method of combination weight based on fuzzy analytic hierarchy process and entropy [40].

The fuzzy set theory is adopted by Shi et al. to measure the trustworthy attributes [40]. The mapping between the trustworthy attribute language variable and the triangular fuzzy number is first built, then the evaluators are invited to establish the fuzzy decision matrix, and finally the trustworthy attribute is measured by defuzzification. In this paper, a similar method is applied to measure trustworthy attributes; however, a more reasonable mapping between the trustworthy attribute language variable and the triangular fuzzy number is given. When building this new mapping, it is taken into account that the greater the software trustworthiness, the more difficult it is to improve the trustworthiness, and the higher the requirement for trustworthy attribute values. Shi et al. present a method to utilize an improved TOPSIS based on vertical projection distance [43] to evaluate the software trustworthiness. The advantage of the improved TOPSIS is that the alternative closest to the positive idea solution is farthest from the negative ideal solution. However, the evaluation method given by Shi et al. can only be applied for ranking the candidate software on trustworthiness. The method presented in this paper can not only be used for the ranking of candidate software on trustworthiness, but can also be used for the trustworthiness measurement of candidate software.

It should be noted that the method presented in this paper can be used for software trustworthiness evaluation, but only when multiple candidate software products must exist. Furthermore, the trustworthy attribute measurement method based on fuzzy theory can model the uncertainty or inherent imprecision of experts' judgment. However, this method only gives the identification measurement of trustworthy attribute in the experts' mind and does not give the measurement model of trustworthy attribute.

7. Conclusions and Future Work

To begin with, a trustworthy attribute measurement method based on fuzzy theory is presented in this paper, which is composed of the mapping relationship between trustworthy attribute level and TFN and defuzzification technology. This method can more reasonably transform the fuzzy decision of trustworthy attribute into a certain real number. Moreover, a software trustworthiness evaluation strategy based on the relationships between criteria is given, including the quantitative relationships between criteria described by the cooperative degrees between criteria, the symmetric substitutivity between criteria approximated by the cooperative degrees between criteria, and a software trustworthiness measurement model via the relationships between criteria. Lastly, we verify the theoretical rationality of the software trustworthiness measurement model by showing that it satisfies the expected properties of software trustworthiness measure. Meanwhile, the case study shows the effectiveness and practicality of the model. This strategy can be used not only to rank candidate software about trustworthiness, but also for the trustworthiness measure of software.

In the future, we will further improve the trustworthy attribute measurement method based on fuzzy theory and establish the measurement model of trustworthy attribute. We will also study the cooperative degrees between criteria for software in different fields, then estimate the symmetric substitutivity related parameters in Definition 3, and construct software trustworthiness measurement models for different fields. Further optimization of grouping methods based on cooperative degrees is also important work for the future.

Author Contributions: Conceptualization, H.T. and Q.C.; methodology, H.T. and Q.C.; algorithm, H.C., Y.X. and S.S.; writing—original draft preparation, H.T.; writing—review and editing, Q.C., H.C., Y.X., S.S. and X.N.; funding acquisition, H.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the National Natural Science Foundation of China (61906175), Doctoral Research Fund of Zhengzhou University of Light Industry (2016BSJJ037) and Science and Technology Project of Henan Province (212102210076, 202102210351).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: A preliminary version of this work was presented at the 9th International Conference on Dependable Systems and Their Applications (DSA2022) and its title is "software trustworthiness evaluation method based on relationships between criteria".

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Avizienis, A.; Laprie, J.C.; Randell, B.; Landwehr, C. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Trans. Dependable Secur. Comput.* **2004**, *1*, 11–33. [CrossRef]
2. *ISO/IEC 25010:2011*; Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—System and Software Quality Models. ISO: Geneva, Switzerland, 2011.
3. McCall, J.A.; Richards, P.K.; Walters, G.F. *Factors in Software Quality: Concepts and Definitions of Software Quality*; General Electric Co: Sunnyvale, CA, USA, 1977.
4. Boehm, B.W.; Brown, J.R.; Kaspar, H.; Lipow, M.; MacLeod, G.J.; Merritt, M.J. *Characteristics of Software Quality*; North-Holland Publishing Company: New York, NY, USA, 1978.
5. He, J.F.; Shan, Z.G.; Wang, J.; Pu, G.G.; Fang, Y.F.; Liu, K.; Zhao, R.Z.; Zhang, Z.T. Review of the Achievements of Major Research Plan of Trustworthy Software. *Bull. Natl. Nat. Sci. Found. China* **2018**, *32*, 291–296. [CrossRef]
6. Tao, H.W.; Fu, L.Y.; Chen, Y.X.; Han, L.; Wang, X. Improved Allocation and Reallocation Approaches for Software Trustworthiness Based on Mathematical Programming. *Symmetry* **2022**, *14*, 628. [CrossRef]
7. Tao, H.W.; Chen, Y.X.; Wu, H.Y.; Deng, R.M. A Survey of Software Trustworthiness Measurements. *Int. J. Perform. Eng.* **2019**, *15*, 2364–2372. [CrossRef]
8. Tao, H.W.; Wu, H.Y.; Chen, Y.X. An Approach of Trustworthy Measurement Allocation Based on Sub-Attributes of Software. *Mathematics* **2019**, *7*, 237. [CrossRef]
9. Tao, H.W.; Chen, Y.X.; Wu, H.Y. A Reallocation Approach for Software Trustworthiness Based on Trustworthy Attributes. *Mathematics* **2020**, *8*, 14. [CrossRef]
10. Oman, P.; Hagemeister, J. Construction and Testing of Polynomials Predicting Software Maintainability. *J. Syst. Softw.* **1994**, *24*, 251–266. [CrossRef]
11. Yuan, Y.; Han, H. A Software Behavior Trustworthiness Measurement Method Based on Data Mining. *Int. J. Comput. Intell. Syst.* **2012**, *4*, 817–825. [CrossRef]
12. Medeiros, N.; Ivaki, N.; Costa, P.; Vieira, M. Vulnerable Code Detection Using Software Metrics and Machine Learning. *IEEE Acc.* **2020**, *8*, 219174–219198. [CrossRef]
13. Lv, Z.H.; Han, Y.; Singh, K.A.; Manogaran, G.; Lv, H.B. Trustworthiness in Industrial IoT Systems Based on Artificial Intelligence. *IEEE Trans. Industr. Inform.* **2021**, *17*, 1496–1504. [CrossRef]
14. Xu, J.L.; Xiao, L.J.; Li, Y.H.; Huang, M.W.; Zhuang, Z.C.; Weng, T.H.; Liang, W. NFMF: Neural Fusion Matrix Factorisation for QoS Prediction in Service Selection. *Connect. Sci.* **2021**, *33*, 753–768. [CrossRef]
15. Tian, F.; Guo, Y.H. Software Trustworthiness Evaluation Model Based on Behavior Trajectory Matrix. *Info. Softw. Technol.* **2020**, *119*, 106233. [CrossRef]
16. Tao, H.W.; Chen, Y.X. A New Metric Model for Trustworthiness of Softwares. *Telecommun. Sys.* **2012**, *51*, 95–105. [CrossRef]
17. Tao, H.W.; Chen, Y.X.; Wu, H.Y. Decomposition of Attributes Oriented Software Trustworthiness Measure Based on Axiomatic Approaches. In Proceedings of the IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C), Macau, China, 11–14 December 2020; pp. 308–315. [CrossRef]
18. Liu, H.; Tao, H.W.; Chen, Y.X. An Approach for Trustworthy Evidence of Source Code Oriented Aerospace Software Trustworthiness Measurement. *AER Control Appl.* **2021**, *47*, 32–41. [CrossRef]
19. Tao, H.W.; Chen, Y.X.; Wu, H.Y. Theoretical and Empirical Validation of Software Trustworthiness Measure Based on Decomposition of Attributes. *Connect. Sci.* **2022**, *34*, 1181–1200. [CrossRef]
20. Tao, H.W.; Chen, Y.X.; Liu, H.; Wu, H.Y.; Hu, Y.H. Attributes Oriented Software Trustworthiness Measure Based on Axiomatic Approaches. *J. Int. Technol.* **2022**, *23*, 583–592. [CrossRef]
21. Shi, H.L.; Ma, J.; Zou, F.Y. A Fuzzy Comprehensive Evaluation Model for Software Dependability Based on Entropy Weight. In Proceedings of the 2008 International Conference on Computer Science and Software Engineering, Wuhan, China, 12–14 December 2008; pp. 683–685. [CrossRef]
22. Li, B.; Cao, Y. An Improved Comprehensive Evaluation Model of Software Dependability based on Rough Set Theory. *J. Softw.* **2009**, *4*, 1152–1159. [CrossRef]
23. Gao, X.T.; Ma, Y.F.; Zhou, W. The Trustworthiness Measurement Model of Component-based Software Based on the Subjective and Objective Weight Allocation Method. In Proceedings of the IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C), Macau, China, 11–14 December 2020; pp. 478–486. [CrossRef]
24. Yang, M.; Gao, T.L.; Xie, W.Y.; Jia, L.; Zhang, T. The Assessment of Cloud Service Trustworthiness State Based on D-S Theory and Markov Chain. *IEEE Acc.* **2022**, *10*, 68618–68632. [CrossRef]
25. Muhammad, D.M.S.; Fairul, R.F.; Loo, F.A.; Nur, F.A.; Norzamzarini, B. Rating of Software Trustworthiness Via Scoring of System Testing Results. *Int. J. Dig. Enterp. Technol.* **2018**, *1*, 121–134. [CrossRef]

26. Yang, X.; Jabeen, G.; Luo, P.; Zhu, X.L.; Liu, M.H. A Unified Measurement Solution of Software Trustworthiness Based on Social-to-Software Framework. *J. Comput. Sci. Technol.* **2018**, *33*, 603–620. [CrossRef]
27. Wang, B.H.; Chen, Y.X.; Zhang, S.; Wu, H.Y. Updating Model of Software Component Trustworthiness Based on Users Feedback. *IEEE Acc.* **2019**, *7*, 60199–60205. [CrossRef]
28. Alarcon, G.M.; Ryan, T.J. Trustworthiness Perceptions of Computer Code: A Heuristic-Systematic Processing Model. In Proceedings of the 51st Hawaii International Conference on System Sciences, Waikoloa Village, HI, USA, 2–6 January 2018; pp. 5384–5393. [CrossRef]
29. Maza, S.; Megouas, O. Framework for Trustworthiness in Software Development. *Int. J. Perform. Eng.* **2021**, *17*, 241–252. [CrossRef]
30. Wang, H.M. Harnessing the Crowd Wisdom for Software Trustworthiness: Practices in China. *ACM SIGSOFT Softw. Eng. Notes* **2018**, *43*, 6–11. [CrossRef]
31. Lee, J.; Kuo, J.Y. Fuzzy Decision Making Through Trade-off Analysis Between Criteria. *Inf. Sci.* **1998**, *107*, 107–126. [CrossRef]
32. Lee, J.; Kuo, J.Y.; Huang, W.T. Fuzzy Decision Making Through Relationships Analysis Between Criteria. In Proceedings of the 1996 Asian Fuzzy Systems Symposium, Kenting, Taiwan, 11–14 December 1996; pp. 296–301. [CrossRef]
33. Xu, Z.S. Research on Software Trustworthiness Measurement Evaluation Model Based on Data Driven. In Proceedings of the 2nd International Conference on Computer Science Communication And Network Security (CSCNS2020), Sanya, China, 22–23 December 2020; pp. 1–4. [CrossRef]
34. Deng, Y.X.; Chen, Z.Z.; Du, W.J.; Mao, B.F.; Liang, Z.Z.; Lin, Q.S.; Li, J.H. Trustworthiness Derivation Tree: A Model of Evidence-Based Software Trustworthiness. In Proceedings of the IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C), Macau, China, 11–14 December 2020; pp. 487–493. [CrossRef]
35. Zhang, F.; Xu, M.D.; Chao, H.C.; Zhang, C.; Liu, X.L.; Hu, F.N. Real-time Trust Measurement of Software: Behavior Trust Analysis Approach Based on Noninterference. *J. Softw.* **2019**, *30*, 2268–2286. [CrossRef]
36. Wang, D.X.; Wang, Q.; He, J. Evidence-Based Software Process Trustworthiness Model and Evaluation Method. *J. Softw.* **2017**, *28*, 1713–1731. [CrossRef]
37. Wang, D.X.; Wang, Q. Trustworthiness Evidence Supporting Evaluation of Software Process Trustworthiness. *J. Softw.* **2018**, *29*, 3412–3434. [CrossRef]
38. Tang, Y.; Pan, Z.; Pedrycz, W.; Ren, F.; Song, X. Viewpoint-based Kernel Fuzzy Clustering with Weight Information Granules. *IEEE Trans. Emerg. Top. Comput. Intell.* **2022**, in press. [CrossRef]
39. Tang, Y.; Zhang, L.; Bao, G.; Ren, F.J.; Pedrycz, W. Symmetric Implicational Algorithm Derived from Intuitionistic Fuzzy Entropy. *Iran. J. Fuzzy Syst.* **2022**, *19*, 27–44. [CrossRef]
40. Li, S.; Yang, S.L.; Li, K. Developing an Evaluation Approach for Software Trustworthiness Using Combination Weights and TOPSIS. *J. Softw.* **2012**, *7*, 532–543. [CrossRef]
41. Chen, S.H.; Hsieh, C.H. Representation, Ranking, Distance, and Similarity of L-R Type Fuzzy Number and Application. *Aust. J. Intell. Process. Syst.* **2000**, *6*, 217–229.
42. Chen, Y.X.; Tao, H.W. *Software Trustworthiness Measurement Evaluation and Enhancement Specification*; Science Press: Beijing, China, 2019.
43. Hua, X.Y.; Tan, J.X. Revised TOPSIS Method Based on Vertical Projection Distance-Vertical Projection Method. *Syst. Eng. Theory Pract.* **2004**, *24*, 114–119. [CrossRef]