Are These Requirements Risky: A Proposal of an IoT-Based Requirements Risk Estimation Framework

Chetna Gupta 1,* and Varun Gupta 2

1 Department of Computer Science and Engineering, Jaypee Institute of Information Technology, Noida 201305, India
2 Department of Economics and Business Administration, University of Alcala, 28802 Madrid, Spain; varun.gupta@uah.es
* Correspondence: chetna.gupta@jiit.ac.in

Abstract: Internet of Things (IoT) systems are revolutionizing traditional living to a new digital living style. In the past, a lot of investigations have been carried out to improve the technological challenges and issues of IoT and have focused on achieving the full potential of IoT. The foremost requisite for IoT software system developers seeking a competitive edge is to include project-specific features and meet customer expectations effectively and accurately. Any failures during the Requirements Engineering (RE) phase can result in direct or indirect consequences for each succeeding phase of development. The challenge is far more immense because of the lack of approaches for IoT-based RE. The objective of this paper is to propose a requirements risk management model for IoT systems. The method regarding the proposed model estimates requirements risk by considering both customers’ and developers’ perceptions. It uses multiple criteria using intuitionistic fuzzy logic and analytical technique. This will help to handle the uncertainty and vagueness of human perception, providing a well-defined two-dimensional indication of customer value and risk. The validity of the approach is tested on real project data and is supported with a user study. To the best of our understanding, literature lacks the trade-off analysis at the RE level in IoT systems and this presented work fills this prerequisite in a novel way by improving (i) requirements risk assessment for IoT systems and (ii) handling developers’ subjective judgments of multiple conflicting criteria, yielding more concrete and more observable results.

Keywords: Internet of Things; intuitionistic fuzzy logic; software engineering; requirements engineering; risk estimation

1. Introduction

Internet of Things (IoT) devices and systems are touching every aspect of our society, ranging from their usage in smart and connected cities, smart nations, smart buildings/homes, smart mobility, transportation, ambient assisted living, smart farms, industry 4.0, and supply chain management, among others [1,2]. Furthermore, IoT technology offers its advantage in other sectors such as healthcare (e.g., IoT-based medical devices or smart health devices) and in the government sector, which uses IoT applications as national security applications. According to the latest report of Forbes [3], the market of IoT is expected to rise to $123 billion by 2021 and up to 25.1 billion units according to reports of [4]. This rapid increase in reliance on IoT devices, combined with the wide range of potential IoT applications, might dramatically amplify the potential vulnerabilities [2].

For the successful implementation and usage of any software system, requirements for elicitation, specification, and management play a key role in development. Requirement Engineering (RE) is the first phase of the software development lifecycle dealing with all kinds of anomalies such as inconsistencies and ambiguities in the requirements specification. This helps in reducing the after-effects of wrong requirement elicitation and
specification leading to rework, increase in cost and time of development, and overall user satisfaction [5]. IoT system development is much more complex than traditional software development and presents new challenges to requirements engineering due to the involvement of heterogeneous technologies [6–9]. However, likewise, traditional software development IoT system development faces higher levels of ambiguity related to market demand, user expectations, working ecosystem (aptitude, tools, intellectual property), and technology. The majority of the existing work addresses attributes of IoT covering issues related to security and connectivity and its representation on the digital platform [10]. To the best of the authors’ awareness, the literature lacks studies dealing with requirement-related issues such as elicitation of hardware and software requirements, their mapping to a given platform, interaction and behavior-related issues between hardware and software for IoT system development, etc. [11].

Understanding, quantifying, and prioritizing threats to organizational assets and operations are all part of risk assessment [9]. However, there is a lack of studies addressing the issue of handling risk in IoT in requirements engineering. The presented work focuses on topics linked to requirements and their specification instead of assessing/examining requirements for risk (discussed in Section 2). Requirement risk assessment is one of the profound areas lacking such practices, particularly in the RE phase of development. Without formal techniques to make an informed decision on critical choices, software engineers can face surprising outcomes. The cost to repair a requirement anomaly varies with how far the development has progressed. The fixing expenditure is 3 times greater in design, 10 times greater during coding, 50 times greater during testing, and around 100 times higher subsequently at the release [12,13]. Well-defined, explicit knowledge about various risks in requirements helps software developers manage the project more effectively and efficiently. Especially in today’s time-to-market era, it is essential to know what risks requirements can have so that a corrective measure/decision can be taken to plan successive releases.

International standard systems and software engineering ISO/IEC/IEEE 15288:2015 [14] and ISO/IEC/IEEE 12207:2017 [15], which explain the software life cycle process, are applied to homogenous systems and can be integrated with other complex systems (e.g., IoT systems). However, these standards are not a one-fit model for all systems. A tailor-made system in this regard can be a good option to satisfy software development needs. Existing risk-based research indicates that a limited number of studies have addressed anomalies in requirements (also referred to as defects), their discovery, verification, and validation of requirements with existing defect taxonomy, and analysis of dependency between requirements, which is still to be explored for IoT systems [11]. The task of measuring risk becomes challenging as requirements interrelate and can have dissimilar dimensions concerning the integration of software and hardware components in IoT systems [6–9]. Therefore, multiple criteria should be considered while measuring the risk related to implementing requirements.

This paper presents a risk assessment framework that considers multiple criteria to support software developers in measuring and mapping the initial discovery of possible risks of executing requirements in IoT platforms. To interleave human actions, Intuitionistic Fuzzy Set (IFS) [16] is used to rank risks by combining several conflicting criteria. From the available catalog of multi-criteria decision models, IFS is a powerful method that leverages the benefits of uncertain and vague human decisions, noticeably and intuitively. It is a scalar value that considers both the best and worst choices with good computational competence. The main contribution of this paper is the presentation of a requirements risk assessment model for IoT systems that considers multiple criteria, in particular, requirements deficiencies, their inter-relationships, complexity, and customer priority to offer robust solutions with reduced overhead in software maintenance. Specifically, the following research questions are addressed: (a) How effective is a multi-criteria IFS-based requirements risk assessment method for IoT systems? Can it improve the expert’s manual judgment, in terms of minimizing software maintenance cost and time? (b) How valid,
concrete, and feasible are the solutions provided by a proposed approach, from the human expert’s perspective?

The remainder of the paper is structured as follows: related work is presented in Section 2 followed by a discussion of the proposed approach in Section 3. Empirical validation is provided in Section 4, and Section 5 presents the conclusion and future work.

2. Related Work

Both academia and industry are expending effort regarding the effective depiction of IoT systems. The literature lacks studies particularly focusing on risk analysis in RE for IoT systems. The core of these systems is software and therefore in this section, both IoT and in general software systems are discussed. Past studies [17–21] have reported a review of the benefits, challenges, and impact of corrective requirements elicitation for IoT applications. Commercial and standardization bodies have also undertaken initiatives promoting IoT requirement analysis and related reference architectures (e.g., Industrial Internet Reference Architecture (IIRA) [22]). Existing systematic reviews on requirement elicitation [18,21,23–26] have summarized empirical evidence, challenges, methods, and outcomes of requirement elicitation, but the authors have not discussed the elicitation of IoT applications. Lim et al. [18] represent the survey on IoT-based elicitation techniques for a handful of methods and conclude that interviews and prototypes are commonly used elicitation techniques, and that users are the common sources of requirements. They also report a lack of methodical understanding of elicitation techniques for IoT where most of the studies are limited to requirements elicitation and specification, the importance and conflicting nature of functional and non-functional requirements, specification using use case diagrams, using modeling language for conceiving requirements, and a scenario-based technique [5,7,27–31]. IoTReq—a Unified Modeling Language (UML) modeling method that supports non-functional requirements is another approach proposed by Reggio [7] for elicitation and specification of IoT system requirements. The work reported by Laplante et al. [27] also focuses on non-functional requirements analysis in IoT systems. Costa et al. [30] reported conflicts between functional and non-functional requirements. A recent study by Ferraris et al. [28] has proposed a trust-based requirements elicitation method focusing on security, privacy, and usability to enhance the level of trust in IoT systems. A study presented by Hassan et al. [32] presents an outline and assessment of six development methods to address feasibility analysis, requirements engineering, analysis and design, and deployment, etc. The results of experimentation, conducted on real-world projects, stated that none of the six approaches covers all the stages necessary for IoT system development.

Requirements risk assessment is closely related to requirement elicitation. The majority of the methods listed in the literature perform quantitative and qualitative analysis on elementary risk processes, including risk discovery, investigation, planning, etc. to ascertain and control the risks [33–37]. These models are suitable for varied projects in small or large organizations used in different fields. According to [38,39], only 5% of the project budget is needed to commence risk management to prevent the probabilities of schedule overrun by 50–70%. Furthermore, researchers have also presented models of risk estimation using the analytic hierarchy process, Bayesian belief network, machine learning, risk metrics, fuzzy entropy, goal-oriented methodologies, decision trees, UML, etc. [40,41].These methods cumulatively state that evaluation of risk can efficiently help in producing software of quality by lessening the exposure to software risk. Furthermore, some researchers [42,43] have addressed in particular requirements risk management, risk-based testing, project risk dependencies, etc. These models have a negligible emphasis on considering software risks in the initial stage of the Software Development Life Cycle (SDLC). More recently, Fahmideh et al. [44] conducted a mixed quantitative and qualitative study on key tasks with industry experts to highlight the most important development process tasks, challenges, and recommendations related to the development of IoT systems. It can be concluded
that the acceptance of early risk detection and mitigation can help in curtailing the loss in software projects.

3. Proposed Approach

Multi-criteria decision-making techniques are beneficial in selecting choices in discrete problems. Risks can strike in any phase of the SDLC and should be controlled suitably using strategies designed for individual phases of SDLC. The IFS-based risk assessment framework will help in finding the best value of decision variables, which will be used as a measure of the effectiveness of a decision. The whole process is sketched in Figure 1.

![Figure 1. Basic steps in the proposed risk assessment method.](image-url)

The foremost prerequisite considered for the proposed approach is to simplify with a quick analysis process yielding accurate and trustworthy results. If both of these conditions are not met, the process is unlikely to be used in the requirements risk assessment process. To achieve this aim, a two-fold process is proposed.

Phase I: Development and application of an intuitionistic fuzzy logic-based model.

Real-world situations are frequently complex and the data gathered is not always complete. In many decision-making situations, we have more than two possibilities, yet we do not always have complete information about those options. Simple set theory and Boolean logic are insufficient to model these situations. Fuzzy set theory [45] and fuzzy logic are commonly used to deal with such circumstances [45]. Despite their wide range of applications [46], fuzzy sets are limited by the uncertainty element due to the lack of comprehensive information. In addition to membership values, Atanassov’s intuitionistic fuzzy sets [16] give an intuitive framework for dealing with vagueness from imprecise information by taking non-membership values into consideration. In this phase, the risk score using IFS on multiple criteria are computed, and later the analytical technique is used to analyze the relationship between the risk score and customer priority value for a specific requirement.

**Brief Introduction of Intuitionistic Fuzzy Sets**

The following section discusses a concise overview of IFS from Atanassov’s intuitionistic fuzzy sets [16, 47].

**Definition 1.** Let $X$ be a non-empty set. An intuitionistic fuzzy set $A$ in $X$ represents an entity with the following form:

$$A := \{ (x, \mu_A(x), \nu_A(x)) | x \in X \}$$  

(1)
where the function $\mu_A : X \to (0, 1)$ and $v_A : X \to (0, 1)$ represents the element’s degree of membership and non-membership $x \in X$, respectively. For every $x \in X$, $\mu_A$ and $v_A$ satisfy, $0 \leq \mu_A(x) + v_A(x) \leq 1$. Contrary to traditional fuzzy sets, the sum of $\mu_A$ and $v_A$ does not always have to be 1. This is especially important when the system does not have all of the information it requires. Besides, we have $\pi_A(x) = 1 - \mu_A(x) - v_A(x)$ as the intuitionistic fuzzy set index or the hesitation margin of $x$ in $A$. $\pi_A(x)$ is the degree of indeterminacy of $x \in X$ to the IFS $A$ and $\pi_A(x) \in (0, 1)$ and $0 \leq \pi_A \leq 1$ for every $x \in X$. $\pi_A(x)$ represents a lack of understanding about whether or not $x$ belongs to IFS $A$.

**Definition 2.** Let $A \in X$ be IFS, then;
1. $\pi(x) = 1 - \mu_A(x) - v_A(x)$ is called the degree of indeterminacy of the element $x \in A$.
2. $\partial(x) = \mu_A(x) + \pi_A(x)\mu_A(x)$ is called the degree of favor of $x \in A$.
3. $\eta(x) = v_A(x) + \pi_A(x)v_A(x)$ is called the degree of opposition to $x \in A$.

For example, let $A$ be an intuitionistic fuzzy set with $\mu_A(x) = 0.5$ and $v_A(x) = 0.3$. Then, $\pi_A(x) = 0.2$, $\partial_A(x) = 0.6$, and $\eta_A(x) = 0.36$. It can be interpreted as “the degree that the object $x$ belongs to IFS $A$ is 0.5, the degree that the object $x$ does not belong to IFS $A$ is 0.3, the degree of hesitancy or indeterminacy of $x$ belonging to IFS $A$ is 0.2, the degree of favor of $x$ belonging to IFS $A$ is 0.6, and the degree of opposition to $x$ not belonging to IFS $A$ is 0.36”.

**Definition 3.** Two IFS $A$ and $B$ are said to be similar or cognate if $\exists \mu_A(x) = \mu_B(x)$ or $v_A(x) = v_B(x)$.

**Definition 4.** Two IFS $A$ and $B$ are said to be equal or comparable if $\mu_A(x) = \mu_B(x)$ and $v_A(x) = v_B(x)$.

**Definition 5.** Two IFS $A$ and $B$ are said to be equivalent to each other i.e., $A$ is equivalent to $B$, denoted by $A \rightarrow B$ if $\exists$ functions $f$: $\mu(x) \rightarrow \mu_B(x)$ and $f$: $v_A(x) \rightarrow v_B(x)$, which are both injection and surjection. Then, the functions define a one-to-one correspondence between $A$ and $B$.

**Definition 6.** Let $A$ and $B$ be two IFS, $A \subseteq B \rightarrow \mu(x) \leq \mu_B(x)$ and $v_A(x) \geq v_B(x)$ for $x \in X$. Then $A$ is a subset of $B$ and $B$ is a superset of $A$.

**Definition 7.** A is a proper subset of $B$, i.e., $A \subset B$ if $A \subseteq B$ and $A \neq B$. It means $\mu(x) \leq \mu_B(x)$ and $v_A(x) \geq v_B(x)$ but $\mu_A(x) \neq \mu_B(x)$ and $v_A(x) \neq v_B(x)$ for $x \in X$.

**Definition 8.** An IFS $A$ is dominated by another IFS $B$ (i.e., $A \leq B$), if there exists an injection from $A$ to $B$. $A$ is strictly dominated by $B$ (i.e., $A \prec B$), if (i) $A \leq B$ and (ii) $A$ is not equinumerous with $B$.

**Definition 9.** Let $A$, $B$, and $C$ be IFSs. Then;
1. $A \trianglelefteq A$, i.e., $A$ is a reflexive relation,
2. $A \trianglelefteq B$ and $B \trianglelefteq A$, i.e., a symmetric relation,
3. $A \trianglelefteq B$ and $B \trianglelefteq C \Rightarrow A \trianglelefteq C$, i.e., a transitive relation.

**Definition 10.** For every two IFSs $A$ and $B$, the following operations can be defined:

\[
A \subseteq B \iff (\forall x \in X)(\mu_A(x) \leq \mu_B(x) \text{ and } v_A(x) \geq v_B(x));
\]

\[
A = B \iff (\forall x \in X)(\mu_A(x) = \mu_B(x) \text{ and } v_A(x) = v_B(x));
\]

\[
A^c = \{<x, \mu_A(x), \mu_A(x) > | x \in X\};
\]

\[
A \cap B = \{<x, \min(\mu_A(x), \mu_B(x)), \max(v_A(x), v_B(x)) > | x \in X\};
\]

\[
A \cup B = \{<x, \min (\mu_A(x), \mu_B(x)), \max(\nu_A(x), \nu_B(x)) > | x \in X\};
\]

\[
\cup A_i = \{<x, \max(\mu_i), \min(\nu_i) > | x \in X \} i \in I;
\]
\( \cap A_i = \{ \langle x, \min(\mu_i) \rangle, \ max(\nu_i) > | x \in X \} \ i \in I; \)

\([A] = \{ \langle x, \mu_A(x) \rangle, \ 1 - \mu_A(x) > | x \in X \} \) (Necessity operator);

\( (\cap) A = \{ \langle x, 1 - \nu_A(x) \rangle, \ \nu_A(x) > | x \in X \} \) (Possibility operator)

The intuitionistic fuzzy set \( B_{ij} \) can be expressed as follows to obtain values from developers for criteria:

\( B_{ij} = \{ r_i, c_j, \mu_{ij}, \nu_{ij} \} \) (2)

where, \( 0 \leq \mu_{ij} \leq 1, \ 0 \leq \nu_{ij} \leq 1 \) and \( 0 \leq \mu_{ij} + \nu_{ij} \leq 1. \)

Also, \( 0 < i \leq n \) and \( 0 < j \leq m, \) where \( n = \) total number of requirements in the given set; \( m = \) total number of criteria, \( \mu_{ij} \) and \( \nu_{ij} \) represent the degree of membership and the degree of non-membership for the requirements \( r_i \in R \) for the criteria \( c_j \in C. \) The greater the hesitation index number, the more hesitant the developers were to choose \( r_i \in R \) for the criteria \( c_j \in C. \)

\( \pi_{ij} = 1 - \mu_{ij} - \nu_{ij} \) (3)

where, \( 0 < i \leq n \) and \( 0 < j \leq m. \)

Each criterion is given a weight based on intuitionistic fuzzy values. For each alternative requirement \( r_i \in R, \) the optimal rank value can be computed using the following equation:

\[ \text{max} \left\{ p_i = \sum_{j=1}^{m} \alpha_{ij} \omega_j \right\} \] (4)

such that: \( i = (1, 2, 3, \ldots, n). \)

\[ \mu_{ij} \leq \alpha_{ij} \leq \mu_{ij}^u \] (5)

\[ \omega_j^l \leq \omega_j \leq \omega_j^u \] (6)

\[ \sum_{j=1}^{m} \omega_j = 1 \] (7)

The following linear programming equations are generated to obtain the optimal rank value using (4):

\[ \text{min} \left\{ p_i^l = \sum_{j=1}^{m} \mu_{ij} \omega_j \right\} \] (8)

such that:

\[ \omega_j^l \leq \omega_j \leq \omega_j^u \]

\[ \sum_{j=1}^{m} \omega_j = 1 \]

and

\[ \text{max} \left\{ p_i^u = \sum_{j=1}^{m} \mu_{ij}^u \omega_j \right\} \] (9)

such that:

\[ \omega_j^l \leq \omega_j \leq \omega_j^u \]

\[ \sum_{j=1}^{m} \omega_j = 1 \]

for each \( i = (1, 2, 3, \ldots, n). \)

Using the Simplex Method, the following optimal weights of the criteria can be calculated.

\[ \omega^* = (\omega_1^*; \omega_2^*; \omega_3^* \ldots \omega_m^*) \]
and

$$\omega'' = (\omega_1''; \omega_2''; \omega_3'' \ldots \omega_m'').$$

With the help of these, the rank value of the requirements can be found as:

$$p'_i = \mu_i \omega_j = \sum_{j=1}^{m} \mu_i \omega_j$$  \hspace{1cm} (10)

and

$$p''_i = \mu_i \omega_j = 1 - \sum_{j=1}^{m} \omega_j \omega_j''$$  \hspace{1cm} (11)

for each \(i = (1, 2, 3, \ldots, n)\).

The objective function \(p'_i\) can be re-written for every requirement \(r_i \in R\) as

$$\min \left\{ p'_i = \frac{\sum_{j=1}^{m} \mu_i \omega_j}{n} \right\},$$  \hspace{1cm} (12)

also,

$$\max \left\{ p'_i = \frac{\sum_{j=1}^{m} \mu_i \omega_j}{n} \right\}$$  \hspace{1cm} (13)

$$\left\{ \omega_j \leq \omega_j'' \leq \omega_j'' \right\} \sum_{j=1}^{m} \omega_j = 1.$$

Using these weight vectors, the following formula can be used to obtain the final optimal value for all the alternatives:

$$\beta_i = \frac{\sum_{j=1}^{m} \mu_i \omega_j'}{1 + \sum_{j=1}^{m} \left( \mu_i - \mu_i \right) \omega_j'}.$$

Phase II: The relationships to be analyzed are divided into three distinct categories, discussed in Table 1.

<table>
<thead>
<tr>
<th>Category 1</th>
<th>Relationships depicting the high ratio of customer priority value to the development risk score.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category 2</td>
<td>Relationships depicting the medium ratio of customer priority value to the development risk score.</td>
</tr>
<tr>
<td>Category 3</td>
<td>Relationships depicting the low ratio of customer priority value to the development risk score.</td>
</tr>
</tbody>
</table>

Using this analytical visualization, the relationship analysis can be performed in two dimensions: according to their development risk score computed and customer priority value elicited from customers. The proposed method is applied and tested on real project data, and later the results are compared to evaluate the effectiveness of the proposed approach. Based on the above-mentioned relationship categories, the software developers can effectively and accurately make an informed decision on the inclusion or exclusion of a particular requirement. It is a scalable framework wherein decisions on addition and omission of multiple criteria can be acquired subject to the project/organizational requirements. Let \(R\) represent a pool of requirements having \(n\) requirements such that \(r_i \in R\). Table 2 below presents the criteria for risk assessment used in this work.
Table 2. Risk estimation criteria.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cm(ri)</td>
<td>refers to the complexity of the requirement.</td>
</tr>
<tr>
<td>Ct(ri)</td>
<td>refers to the approximation of funding including analyses, maintenance, tools, and software, etc. to implement and investigate the specific requirement for the IoT system.</td>
</tr>
<tr>
<td>CPV(ri)</td>
<td>refers to the opinion of value, significance, and the need of development to users. It is a multidimensional concept depending on the viewpoint of the users during elicitation.</td>
</tr>
<tr>
<td>Pr(ri)</td>
<td>refers to the penalty initiated when a certain requirement/feature of it is not implemented or is missed.</td>
</tr>
<tr>
<td>Df(ri)</td>
<td>refers to the ambiguities in requirements related to the goal or scope of the project. This work follows Hayes’ [48] taxonomy because it emphasizes requirement ambiguities. It is computed as the number of defects in a requirement divided by the total number of defects.</td>
</tr>
</tbody>
</table>

These risk estimation criteria are elected through knowledge abstraction from the Project Management Body of Knowledge (PMBOK), blogs, and discussion with software industry practitioners [49–51].

4. Empirical Validation

To perform and validate the proposed approach, it was tested on real project data from [52]. The summary of the project is as follows: it is a web portal assisting clients by providing on-demand and reserved parking slots through an application build over iPhone Operating System (IOS) and Android platforms. Currently, this allocation is performed manually. The company profile is shown in Table 3.

Table 3. Company Profile.

<table>
<thead>
<tr>
<th>Company Profile</th>
<th>More than 800 employees in a company with an experience of more than ten years with a product partnership with intermediate service providers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounts/Engagements</td>
<td>Sixty percent of the projects are end-to-end, starting from requirements analysis, User Experience (UX) design, product management, product architecture, product engineering, Quality Assurance (QA), support, and operations; 20% product engineering and strategy, 10% QA Automation and Verification activities, and 10% support and operations.</td>
</tr>
</tbody>
</table>

This project was developed as a part of global software development in India and the US during 2018–2019.

A list of 12 high-level requirements comprising 84 lower-level requirements was finalized for the project during the RE phase. The customer priority value was elicited, and the developer’s analysis was recorded using IFS. The process started with the gathering of customers; preferences for each requirement (CPV (ri)). These preferences were recorded on a five-points Likert scale using interactive Graphical User Interface (GUI) support.

4.1. Results and Discussion

The steps below were performed to come to a conclusion on the utilization of the recommended method:

(1) Using the intuitionistic inputs (discussed above) the criterion was calculated using linear programming.

(2) The weights for the membership and non-membership values for the selected criteria were calculated using (8) and (9). The best rankings for each requirement were determined by utilizing (14).
(3) Finally, these weights were used to calculate \( p^u \) and \( p^l \) using to (10) and (11). Thereafter, the normalized weights (WRs) were computed for each requirement.

(4) The next risk–value relationship was conducted. Figure 2 shows the Risk–Customer priority value diagram for 10 requirements collected from a sub module of real project data. The relationships are analyzed within three distinct categories:

- Relationship depicting the high ratio of customer priority value to development risk score (a priority value–risk ratio exceeding 2).
- Relationships depicting the medium ratio of customer priority value to development risk score (a priority value–risk ratio between 0.5 and 2).
- Relationships depicting the low ratio of customer priority value to development risk score (a priority value–risk ratio lower than 0.5).

![Figure 2. Risk–Customer priority value diagram presenting relationship for project data. R1–R9 are the software requirements.](image-url)

The customer priority value–risk value diagram helps requirements risk assessment. It can be concluded that if the requirements with high and medium ratios were selected for implementation over the requirements that contribute little to customers’ priority value, risk, cost, and duration of the project can be reduced significantly.

Generally, dependency is represented over time, which specifies that an activity or task cannot begin until a preceding activity has been completed. Analysts look at the data and keep track of the needs’ dependencies. There is a lack of trade-off analysis in software engineering, notably for requirements analysis for IoT software systems, which is a vital aspect of multidisciplinary systems engineering. The authors believe that employing IFS to implement a customer priority value–risk value approach is a good starting step toward meeting this requirement. However, due to the mathematical foundation of IFS, the proposed approach yields more concrete results complementing the win–win situation. The proposed method is a valuable addition to the literature because it is more observable, robust, and easier to use.
4.2. Survey and Feedback

To answer the research question, a survey was conducted with 35 industry experts with more than 8 years of rich experience. Respondents’ scores were gathered on a five-point Likert scale. The purpose of the survey questionnaire was to gather information in order to answer the following question—“Can the proposed approach effectively be used to assess risk quantitatively and without ambiguity?”

The data was collected online and analyzed to see how well the proposed solution fits in real-world projects in producing the expected result. Figure 3 summarizes the results of the survey for formulated questions to capture the response. The following are the questions used in the survey:

1. Does the proposed solution for risk assessment make sense, in general?
2. Are there any constraints or dependencies that make the solution infeasible?
3. Is it costly to include the solution? Are there any difficulties that would make it a long process?
4. Is there any difficulty in terms of missing or misleading values?
5. Is it feasible to change your inputs? In such a situation, will it affect the solution? Do you find it positive?
6. In comparison to the approach followed in your workplace, will this solution have the potential to make a difference?

![Analysis of Survey Results](image)

Figure 3. Analysis of survey results.

It can be concluded from the survey results that the majority of the respondents agreed that the proposed approach is a promising solution assisting developers to measure risk early in the RE phase, thereby reducing the overhead of cost and time of software maintenance.

5. Conclusions and Future Work

This paper presents a multi-criteria requirements risk estimation model for IoT systems, helping software developers to assess the probable risk of implementing gathered requirements. This will help in administering preventive actions to mitigate risk as early as
possible. This model captures customer value and computes requirements risk using the intuitionistic fuzzy logic method, a multi-criteria decision analysis method. It is combined with an analytical method to analyze customer value and requirements risks. It is essential to balance customer satisfaction and risk associated with implementing requirements as both vary by order of magnitude. IFS is a powerful method to handle the uncertainty and vagueness of human perception. Results of experimentation and surveys on real project data and with IT experts conclude that the proposed method is an effective measure of assessing requirements risk by minimizing maintenance time and fostering customer value. The work presented is novel in improving (i) requirements risk assessment and (ii) handling the developers’ subjective judgments of multiple conflicting criteria to provide robust solutions.

As a part of future work, the authors plan to store the analysis using a cloud computing framework about the people responsible, their rationale, their assumptions, and their initial and final decision values to support software requirement reuse for distributed software development/global software development.

Author Contributions: Conceptualization, C.G; methodology, C.G. and V.G.; software, C.G. and V.G.; validation C.G. and V.G.; formal analysis, C.G.; investigation, C.G. and V.G.; resources, C.G. and V.G.; data curation, C.G. and V.G.; writing—original draft preparation, C.G. and V.G.; writing—review and editing, C.G. and V.G.; visualization, C.G. and V.G.; supervision, C.G. and V.G.; project administration, C.G. and V.G.; funding acquisition, C.G. and V.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to express their sincere thanks to the software industry practitioners for taking part in the validation. Their expertise has been very useful in the initial validation.

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

References

23. Pacheco, C.; Garcia, I.; Reyes, M. Requirements elicitation techniques: A systematic literature review based on the maturity of the techniques. *IET Softw.* 2018, 12, 365–378. [CrossRef]
33. Al-Sarayreh, K.T.; Meridji, K.; Abran, A. Software engineering principles: A systematic mapping study and a quantitative literature review. *Inf. Softw. Technol.* 2017, 92, 194–221. [CrossRef]