

Software Reliability Allocation using FAHP and Trapezoidal Fuzzy Set Theory

Ankita

Department of Mathematics, Kumaun University, Nainital, Uttarakhand, India
E-mail: chaudharyankita145@gmail.com
ORCID iD: <https://orcid.org/0009-0007-7926-6651>

Deepak Kumar*

Department of Mathematics, Kumaun University, Nainital, Uttarakhand, India
E-mail: deepakdev16@gmail.com
ORCID iD: <https://orcid.org/0000-0002-7979-9307>

*Corresponding author

Received: December 23, 2025; Revised: February 26, 2026; Accepted: March 10, 2026; Published: August 8, 2026

Abstract: Reliability of a software system is an important attribute of software quality, and its importance has increased significantly due to the growing demand for high-quality software systems. Evaluating reliability during the design phase is essential for improving system performance and ensuring effective planning. At this stage, an important challenge is to identify suitable methods for achieving the desired reliability of the software system. Reliability allocation methods can be applied to assign reliability targets to individual components prior to the actual system design. To address this, a new hierarchical model is proposed that integrates the perspectives of users, software programmers, and software managers. The system is decomposed into multiple levels, including functions, programs, modules, and submodules, to enable systematic analysis. To handle uncertainty and vagueness in human judgments, the fuzzy analytic hierarchy process (FAHP) is employed, incorporating the geometric mean method with trapezoidal fuzzy numbers. The proposed approach determines reliable weights and effectively allocates reliability targets at each level. A comparison with the classical AHP method demonstrates that FAHP provides a more flexible and realistic representation by capturing uncertainty, making it more suitable for reliability allocation in complex software systems.

Index Terms: Software Reliability, Reliability Allocation, Geometric Mean Method, Trapezoidal Fuzzy Number, Fuzzy Analytic Hierarchy Process (FAHP).

1. Introduction

Reliability plays a vital role in the development of a system, as it determines how well the system performs its job and how often it fails. A reliable system can provide consistent and satisfactory results, whereas unreliable system can lead to errors, defects, delays, accidents or losses. This can have an effect on performance, safety, cost and customer satisfaction. Therefore, the reliability of a system significantly impacts on the efficiency, quality and effectiveness of the system output, along with the user experience. It also reduces the risk of harm or damage to the system. Additionally, it can lower the expenses of operating, maintaining, repairing or replacing the system or its components. Most importantly, software reliability can enhance the reputation, trust and loyalty of the customers or users of the system.

Software reliability is the probability of a software system executing its assigned tasks effectively in a particular environment for a certain period of time. There are various software reliability growth models that are used to achieve the reliability target/goal of software to reduce the probability of failure of any complex system. To implement these models, it is essential to understand their compatibility with actual failure data. At this stage of development, an efficient, emphatic and methodical software development process is needed to create software that is extremely reliable. The method of allocating reliability plays a pivotal role in the design stage, as it assists in determining and distributing reliability targets/goals. Initially, research on reliability allocation focused primarily on hardware systems; however, over time, attention has shifted toward software reliability allocation. Zahedi et al. (1991) [1], Poore et al. (1993) [2], and Leung (2004) [3] were among the early contributors in this area. Zahedi et al. (1991) [1] applied the analytic hierarchy process (AHP) to model software architecture, incorporating cost constraints and proposing a method to maximize overall software reliability. Poore (1993) [2] introduced a spreadsheet-based methodology to explore

strategies for distributing reliability across software systems effectively. Mirchandari C. (2018) [4] developed a model for evaluating software reliability throughout the entire software life cycle, providing insights into the system's reliability over time. Later, Pradhan S.K. et al. (2021) [5] proposed an optimal resource allocation model aimed at reducing software costs during testing and improvement phases. Luo H. and Xu L. (2023) [6] introduced a software reliability growth model based on a NHPP framework. Fang C. (2025) [7] integrated Bayesian analysis into SRGMs to optimize software release strategies. In recent years, several innovative approaches have further enhanced software reliability. Xie D. (2025) [8] developed a method using natural language processing techniques to improve software reliability, while Samal (2025) [9] focused on efficient fault detection mechanisms. Additionally, Hussain et al. (2025) [10] proposed a novel SRGM incorporating particle swarm optimization for parameter estimation, effectively addressing uncertainty and dependent failure conditions. Most recently, Dhavakumar P. N. (2026) [11] developed a software defect prediction model to further enhance software reliability.

On the other hand, the software development process involves multiple stages such as design, coding, maintenance, and other user requirements, all of which are carried out by human beings. Since human behavior is inherently uncertain and imprecise, it can be effectively represented using fuzzy concepts. Moreover, the software environment itself exhibits a significant degree of uncertainty. Therefore, fuzzy set theory plays a crucial role in the advancement of software reliability models. Early contributions by Cai, Wen, and Zhang (1991, 1993) [12, 13] and Singh (1995) [14] explored software reliability modeling using fuzzy set theory. Subsequently, Chatterjee et al. [15] proposed a FAHP-based model for distributing target reliability during the design and planning stages of software systems. In recent years, several studies have further extended the application of FAHP in software reliability and decision-making problems. For instance, Upadhyay et al. (2022) [16] developed a FAHP-based model incorporating operational profiles for more realistic reliability allocation. Babu (2024) [17] utilized a FAHP-based mathematical model to enhance software reliability evaluation. In 2024, Yuen [18] developed a fuzzy cognitive network process for evaluating software reliability and quality, and performed a comparative analysis with FAHP, highlighting its effectiveness in handling complex decision-making environments. Aliyeva (2025) [19] applied FAHP in facility location selection for software systems, while Khan et al. (2025) [20, 21] employed a FAHP-based framework to improve software development security. These studies highlight the effectiveness of FAHP in handling uncertainty and improving decision-making in software engineering applications.

In the context of reliability allocation using FAHP, Chatterjee et al. [15] proposed a model for distributing target reliability during the design and planning stages of software systems, employing triangular fuzzy numbers to represent expert judgments. Chakraborty also used triangular fuzzy numbers to enhance reliability evaluation in hierarchical software systems. In contrast, the present study employs trapezoidal fuzzy numbers in the FAHP, which provide greater flexibility and a more accurate representation of uncertainty. Additionally, a comparison with the standard AHP method has been conducted to highlight the effectiveness of the proposed approach. The proposed FAHP-based model integrates user requirements and preferences with the technical aspects of software, including program, module, and submodule reliabilities. It establishes reliability targets during the design and planning phases and features a hierarchical structure linking the user's perspective on reliability with that of software managers and programmers. This framework aims to extract the necessary parameters for developing a robust reliability allocation model and is particularly advantageous for addressing the allocation of reliability in hierarchically organized software systems, ensuring systematic distribution of reliability goals across programs, modules, and sub modules.

This paper is structured as follows: Section 2 provides fundamental definitions related to fuzzy set theory. Section 3 discusses the fuzzy analytic hierarchy process (FAHP). Section 4 presents a discussion on the proposed reliability allocation model, which is structured hierarchically. Section 5 presents an example for justifying our proposed method. Section 6 presents a comparison between classical AHP and FAHP. Lastly, Section 6 concludes the paper with future research directions.

2. Preliminaries

In 1965, Zadeh [22] introduced the concept of fuzzy set theory to address the uncertainties arising from vagueness in systems or data. This theory encompasses fuzzy sets, membership functions and fuzzy numbers, which collectively transform ambiguous data into valuable information effectively.

2.1. Fuzzy set [22]:

If X is a collection of objects and x is any particular element of X , then fuzzy set A is defined as:

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

where $\mu_A(x): X \rightarrow [0,1]$ is called the membership function or grade of membership (also degree of compatibility or degree of truth) of x .

2.2. Fuzzy number [23]:

A fuzzy set A on R must possess at least the following three properties to satisfy as a fuzzy number:

1. A must be a normal fuzzy set.
2. The α -cut of A , must be closed interval for every $\alpha \in (0, 1]$.
3. The support of A must be bounded.

2.3. Trapezoidal fuzzy number [23]:

A fuzzy number $A = (a, b, c, d)$ is called trapezoidal fuzzy number (graphically represented in Figure 1) if its membership function is given by:

$$\mu_A(x) = \begin{cases} \frac{x-a}{b-a}, & a \leq x \leq b, \\ 1, & b \leq x \leq c, \\ \frac{c-x}{c-b}, & c \leq x \leq d, \\ 0, & \text{otherwise} \end{cases}$$

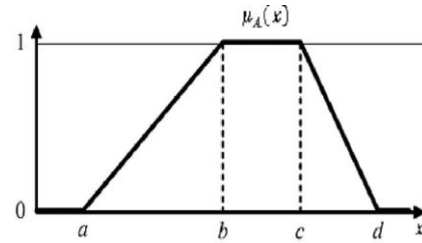


Fig. 1. Trapezoidal fuzzy number.

3. Fuzzy Analytic Hierarchy Process (FAHP)

The Thomas Saaty first introduced the analytic hierarchy process (AHP) in 1980 [24]. It provides a systematic frame work for decomposing a complex decision problem into smaller and more manageable components. However, due to the inherent uncertainty in real-world applications and the vagueness in human judgment, it is often difficult for decision makers to provide precise numerical comparisons. As a result, subjective assessments are more suitably expressed in terms of fuzzy values rather than exact ratios. To address this limitation, the Fuzzy Analytic Hierarchy Process (FAHP) has been widely adopted in the literature. Several models have been proposed to determine criteria weights and prioritize alternatives under uncertainty. Among these, the approaches developed by Witold Pedrycz and Frits van Laarhoven (1983) [25], James J. Buckley (1985) [26], and Da-Yong Chang (1996) [27] are among the most commonly used methods in decision-making problems. FAHP extends the classical AHP by incorporating fuzzy set theory, thereby enabling a more realistic representation of uncertainty in pairwise comparisons. The FAHP methodology generally consists of three main steps, as proposed by Buckley (1985) and Chang (1996), which are:

1. Define your goal and decompose the unstructured problem into a hierarchical structure with decision elements (criteria and alternatives).
2. Employ pairwise comparisons among decision elements to establish the fuzzy judgment matrix and a fuzzy weight vector.
3. Aggregate relative weights of decision elements to obtain an overall rating for alternatives and select the optimal one.

4. Software Reliability Allocation Model based on Hierarchical Framework

The ultimate evaluation of a system's attributes and reliability lies with the user. Therefore, prior to the development of an actual system, it is essential to consider users' views and opinions regarding the reliability of various functions. However, in practice, the perception of reliability often varies among users, software developers, and project managers, due to differences in expectations, experience, and domain knowledge. This divergence introduces uncertainty and complexity into the reliability assessment process.

To address this challenge, it becomes necessary to develop a systematic framework capable of integrating these diverse viewpoints in a coherent and consistent manner. Such an approach facilitates the assignment of appropriate reliability values to various components of a software system, including functions, programs, modules, and sub modules. By doing so, the proposed methodology ensures a more realistic and comprehensive evaluation of system reliability, aligning technical design with user expectations.

4.1. Hierarchy of software system

In this model, a hierarchical structure is constructed that integrates the perspectives of the user/decision makers, with those of software programmer and the software manager. The objective of this framework is to extract the necessary parameters for developing a reliability allocation model. In the analytical hierarchy process, a top-down

methodology is used to organize the problem into a hierarchical structure. Conversely, a bottom-up methodology mathematically integrates various steps of this hierarchy. In this introduced model, the hierarchy begins at the top, reflecting the perspective of the user, which is characterized as the overarching goal of software reliability or system reliability, represented by R in the first tier of the hierarchy in Figure 2.

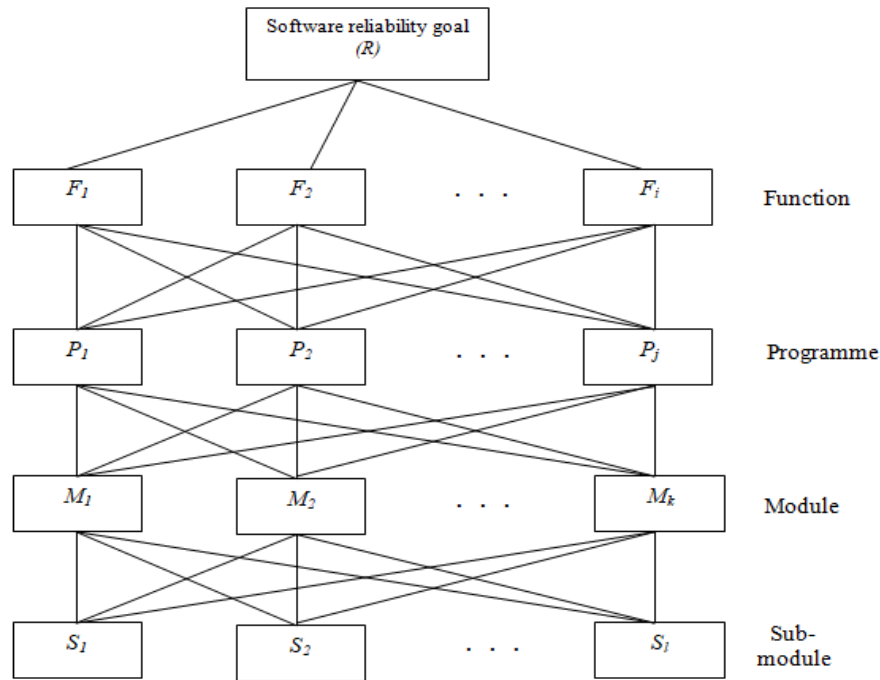


Fig. 2. The hierarchy of software reliability allocation model.

The user formulates their evaluation on the basis of software's functionality and attributes, as illustrate in the second phase of the hierarchy. The user anticipates that the software will execute a specific set of functions and yield the expected outcomes. This perspective reflects the user's view about the software. Let us consider that a typical user identifies such functions and labels them as $F_1, F_2, \dots, F_i$ for $i = 1, 2, \dots, n$.

The third phase of the hierarchy signifies the computer program developed to fulfil the functions outlined by the user. This phase reflects the perspective of the software engineer. (There is no reason to believe that each user-specified function would be programmed into only one programme, i.e., a one-to-one mapping from function to programmes could be inefficient and unrealistic). Consequently, every function may be associated with multiple programs, as illustrated in Figure 2. The software programs in the third phase are identified as $P_1, P_2, \dots, P_j$ for $j = 1, 2, \dots, n$.

The fourth phase of the hierarchy comprises the independent modules that constitute the programs. In the formulation, it is assumed that software engineer adheres strictly to the concept of structured programming, which has become an inevitable programming approach for medium and large systems, which in turn, are prime candidates for using a reliability allocation model. It is assumed that the software consists of k module labels of $M_1, M_2, \dots, M_k$ for $k = 1, 2, \dots, n$ which are denoted in the fourth phase of the hierarchy.

The fifth phase of the hierarchy comprises the independent submodules that constitute the modules. It is assumed that the software uses l submodule labels of $S_1, S_2, \dots, S_l$ for $l = 1, 2, \dots, n$ which are denoted in the fifth phase of the hierarchy.

4.1. Proposed methodology

In 1985, Buckley proposed the geometric mean method on FAHP, which has been utilized in various problems to evaluate the weights of different alternatives and criteria. Geometric mean method uses triangular fuzzy number to assess the perspectives of decision makers. However, trapezoidal fuzzy numbers provide a more flexible and generalized representation of uncertainty compared to triangular fuzzy numbers, as they can better capture the range and ambiguity inherent in human assessments. Therefore, the present study adopts trapezoidal fuzzy numbers to model user opinions more effectively.

Accordingly, the proposed methodology integrates trapezoidal fuzzy numbers with the FAHP framework. Fuzzy pairwise comparison matrices are constructed using linguistic evaluations, which are then transformed into corresponding trapezoidal fuzzy numbers. The linguistic variables employed for pairwise comparisons of each criterion are presented in Table 1.

Table 1. Linguistic variables for pairwise comparison of each criterion.

Linguistic scale	Trapezoidal scale	Trapezoidal fuzzy reciprocal scale
Equal	(1,1,1,1)	(1,1,1,1)
Equally important	(1,1,2,3)	(1/3,1/2,1,1)
Weakly important	(1,2,3,4)	(1/4,1/3,1/2,1)
Moderately more important	(2,3,4,5)	(1/5,1/4,1/3,1/2)
More important	(4,5,6,7)	(1/7,1/6,1/5,1/4)
Strongly more important	(6,7,8,9)	(1/9,1/8,1/7,1/6)

Now, the algorithm of proposed FAHP method is as follows:

Step 1: Construct the fuzzy pairwise comparison matrix $[a_{ij}]_{n \times n}$, in the form of trapezoidal fuzzy number by using Table 1 as follows:

$$\begin{bmatrix} (1,1,1,1) & (l_{12}, m_{12}, n_{12}, u_{12}) & \dots & (l_{1n}, m_{1n}, n_{1n}, u_{1n}) \\ (l_{21}, m_{21}, n_{21}, u_{21}) & (1,1,1,1) & \dots & (l_{2n}, m_{2n}, n_{2n}, u_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ (l_{n1}, m_{n1}, n_{n1}, u_{n1}) & (l_{n2}, m_{n2}, n_{n2}, u_{n2}) & \dots & (1,1,1,1) \end{bmatrix}$$

where $a_{ij} \times a_{ji} \approx 1$, $a_{ij} = (l_{ij}, m_{ij}, n_{ij}, u_{ij})$ and $(a_{ij})^{-1} = (\frac{1}{u_{ij}}, \frac{1}{n_{ij}}, \frac{1}{m_{ij}}, \frac{1}{l_{ij}})$.

Step 2: The fuzzy geometric mean value r_i , for each criterion i is calculated as

$$r_i = (a_{i1} \times a_{i2} \times \dots \times a_{in})^{\frac{1}{n}} \quad \forall (i = 1, 2, \dots, n).$$

Step 3: The fuzzy weight w_i for each criterion i is calculated as

$$w_i = r_i \times (r_1 + r_2 + \dots + r_n)^{-1} \quad \forall (i = 1, 2, \dots, n)$$

Here $r_i = (l_i, m_i, n_i, u_i)$ and $(r_i)^{-1} = (\frac{1}{u_i}, \frac{1}{n_i}, \frac{1}{m_i}, \frac{1}{l_i})$.

Step 4: Here fuzzy weights $w_i = (l_i, m_i, n_i, u_i)$ are defuzzified by taking average, which is as follows:

$$W_i = \frac{l_i + m_i + n_i + u_i}{4} \quad \forall (i = 1, 2, \dots, n)$$

After that we'll get the defuzzified weight vector as

$$W' = (W_1, W_2, \dots, W_n) \quad \forall (i = 1, 2, \dots, n).$$

Step 5: Normalize the defuzzified weight vectors as

$$N_i = \frac{W'_i}{\sum_{i=1}^n W'_i} \quad \forall (i = 1, 2, \dots, n)$$

After that we'll get the normalised weight vectors as

$$W = (N_1, N_2, \dots, N_n) \quad \forall (i = 1, 2, \dots, n).$$

4.2. Reliability allocation

Reliability is now allocated to all system components. For functions F_i , the allocated reliability [8] is calculated as

$$RF_i = R^{WF_i} \quad \forall (i = 1, 2, \dots, n)$$

Here RF_i is allocated reliability for functions $F_i \quad \forall i = 1, 2, \dots, n$, R is reliability goal of software system and WF_i is normalised weight vector of functions $F_i \quad \forall i = 1, 2, \dots, n$.

The reliability target/goal at the programme phase will be RF_i for those programmes, that are associated with function i . To execute function i , a matrix denoted as AP_j^i , with the superscript i , is provided to indicate that this matrix is intended for the execution of function F_i and subscript j is provided to indicate those programmes which are associated with function i . The fuzzy weights are determined using geometric mean method which is discussed in

subsection 4.2. After defuzzification and normalization, we'll get the normalised weight vectors of the programmes $P_j \forall j = 1, 2, \dots, n$ which belong to function i as

$$WP^i = (WP_j^i) \forall (i = 1, 2, \dots, n)$$

For programmes j which belong to function i , the allocated reliability is calculated as

$$RP_j^i = (RF_i)^{WP_j^i} \forall (i = 1, 2, \dots, n)$$

Here RF_i is allocated reliability for functions $F_i \forall i = 1, 2, \dots, n$ and WP_j^i is normalised weight vector of the programmes $P_j \forall j = 1, 2, \dots, n$ which belong to functions $F_i \forall i = 1, 2, \dots, n$.

The programme P_j will possess different reliability values corresponding to different functions. For convenience, the highest reliability value is chosen.

$$RP_j = \text{Maximum}(RP_j^i) \forall (j = 1, 2, \dots, n)$$

Here RP_j is allocated reliability for programmes $P_j \forall j = 1, 2, \dots, n$.

The reliability target/goal for the module phase will be RP_j for those modules that are associated with programme j . To execute programme j , a matrix denoted as AM_k^j , with the superscript j , is provided to indicate that this matrix is intended for the execution of programme P_j and subscript k is provided to indicate those modules which are associated with programme j . The fuzzy weights are determined using geometric mean method. After defuzzification and normalization, we'll get the normalised weight vectors of the modules $M_k \forall k = 1, 2, \dots, n$ which belong to programme j as

$$WM^j = (WM_k^j) \forall (j = 1, 2, \dots, n)$$

For modules k , which belong to programme j , the allocated reliability is calculated as

$$RM_k^j = (RP_j)^{WM_k^j} \forall (j = 1, 2, \dots, n)$$

Here RP_j is allocated reliability for programmes $P_j \forall j = 1, 2, \dots, n$ and WM_k^j is normalised weight vector of the modules $M_k \forall k = 1, 2, \dots, n$ which belong to programme j .

The module M_k will possess different reliability values corresponding to different programmes. For convenience, the highest reliability value is chosen.

$$RM_k = \text{Maximum}(RM_k^j) \forall (j, k = 1, 2, \dots, n)$$

Here RM_k is allocated reliability for modules $k = 1, 2, \dots, n$.

The reliability target/goal for the submodule phase will be RM_k for those submodules, that are associated with module k . To execute module k , a matrix denoted as AS_l^k , with the superscript k , is provided to indicate that this matrix is intended for the execution of module M_k and subscript l is provided to indicate those submodules which are associated with module k . The fuzzy weights are determined using geometric mean method. After defuzzification and normalization, we'll get the normalised weight vectors of the submodules $S_l \forall l = 1, 2, \dots, n$ which belong to module k as

$$WS^k = (WS_l^k) \forall (k = 1, 2, \dots, n)$$

For submodules l , which belong to module k , the allocated reliability is calculated as

$$RS_l^k = (RM_k)^{WS_l^k} \forall (k, l = 1, 2, \dots, n)$$

Here RM_k is allocated reliability for module $M_k \forall k = 1, 2, \dots, n$ and WS_l^k is normalised weight vectors of the submodules $S_l \forall l = 1, 2, \dots, n$ which belong to module k .

The submodule S_l will possess different reliability values corresponding to different modules. For convenience, the highest reliability value is chosen.

$$RS_l = \text{Maximum}(RS_l^k) \quad \forall (k, l = 1, 2, \dots, n)$$

Here RS_l is allocated reliability for submodules $l = 1, 2, \dots, n$.

5. Numerical Illustration

The hierarchical structure consisting of system, function, program, module, and submodule levels is widely applicable in real-world software engineering environments. In large-scale software systems such as enterprise applications, cloud-based platforms, and distributed systems, the overall system is decomposed into multiple functions, which are further implemented through programs, modules, and submodules to ensure modularity and efficient management. For example, in enterprise systems, functions such as authentication, data processing, and reporting are organized into programs and further divided into modules and submodules to perform specific tasks. Therefore, the adopted hierarchical structure reflects practical software design principles and provides a realistic basis for reliability allocation.

In this study, for the purpose of analysis and demonstration of the proposed model, it is assumed that the software system consists of a finite number of functions, programs, modules, and submodules. This assumption facilitates the systematic application of the methodology while preserving its applicability to real-world systems. The hierarchical structure of software reliability is depicted in Figure 3.

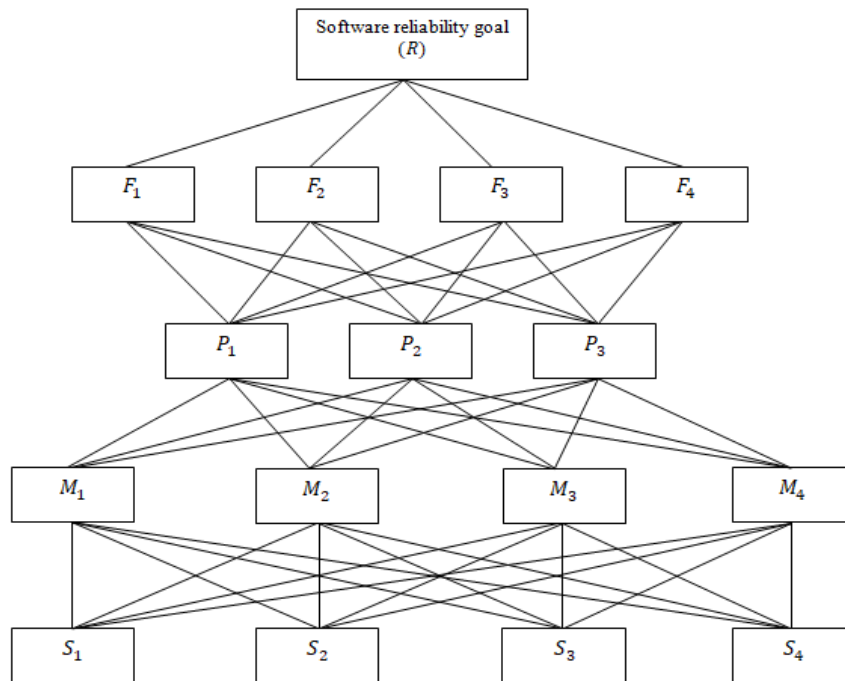


Fig. 3. The hierarchy of software reliability allocation model.

Consider a software system consisting of four functions, F_1, F_2, F_3 and F_4 intended for diverse applications. Additionally, it is presumed that the users desired reliability target or reliability goal (R) for the system is 0.95. Users are asked to perform pairwise comparisons of these functions using the trapezoidal fuzzy scale given in Table 1. This scale is used by decision maker when conducting the pairwise comparison. The resulting comparison matrix is presented in Table 2. To ensure the reliability and logical consistency of the judgments, the consistency of the constructed matrix was evaluated after defuzzification. The obtained consistency ratio (CR) is 0.0185, which is well below the acceptable threshold value of 0.1. This indicates that the pairwise comparisons provided by the experts are consistent and the matrix is suitable for further analysis.

Table 2. Pairwise comparison matrix for functions F_i ($i = 1, 2, 3, 4$).

Functions	F_1	F_2	F_3	F_4
F_1	(1,1,1,1)	(1,1,2,3)	(1,2,3,4)	(2,3,4,5)
F_2	(1/3,1/2,1,1)	(1,1,1,1)	(1,1,2,3)	(1,2,3,4)
F_3	(1/4,1/3,1/2,1)	(1/3,1/2,1,1)	(1,1,1,1)	(1,1,2,3)
F_4	(1/5,1/4,1/3,1/2)	(1/4,1/3,1/2,1)	(1/3,1/2,1,1)	(1,1,1,1)

Based on method discussed in subsection 4.2, geometric mean value r_{F_i} for each function F_i ($i = 1,2,3,4$) can be calculated as

$$\begin{aligned} r_{F_1} &= (1.1892, 1.5650, 2.2133, 2.7831) \\ r_{F_2} &= (0.7598, 1, 1.5650, 1.8612) \\ r_{F_3} &= (0.5372, 0.6389, 1, 1.3160) \\ r_{F_4} &= (0.3593, 0.4518, 0.6389, 0.8408) \end{aligned}$$

The fuzzy weights w_{F_i} for each function F_i ($i = 1,2,3,4$) can be calculated as

$$\begin{aligned} w_{F_1} &= (0.4179, 0.4280, 0.4085, 0.4092) \\ w_{F_2} &= (0.2670, 0.2735, 0.2888, 0.2736) \\ w_{F_3} &= (0.1887, 0.1747, 0.1845, 0.1934) \\ w_{F_4} &= (0.1262, 0.1235, 0.1179, 0.1236) \end{aligned}$$

Via defuzzification, the defuzzified weight vectors $W'F_i$ of functions F_i ($i = 1,2,3,4$) are

$$W'F_i = (0.4159, 0.2757, 0.1853, 0.1228)$$

Via normalization, the normalised weight vectors WF_i of functions F_i ($i = 1,2,3,4$) are

$$WF_i = (0.4160, 0.2757, 0.1853, 0.1228)$$

For functions F_i ($i = 1,2,3,4$), the allocated reliability RF_i is calculated as

$$RF_1 = 0.9788, RF_2 = 0.9859, RF_3 = 0.9905, RF_4 = 0.9737$$

For function F_1 , programme P_1, P_2 and P_3 are taken as shown in Figure 3. For accomplishing function F_1 , the comparison matrix AP_j^1 ($j = 1,2,3$) is depicted by Table 3 and the consistency ratio (CR) is 0.08, confirming acceptable consistency.

Table 3. Pairwise comparison matrix AP_j^1 .

Programmes	P_1	P_2	P_3
P_1	(1,1,1,1)	(1,1,2,3)	(1,2,3,4)
P_2	(1/3,1/2,1,1)	(1,1,1,1)	(1,1,2,3)
P_3	(1/4,1/3,1/2,1)	(1/3,1/2,1,1)	(1,1,1,1)

The normalised weight vector (WP^1) of programmes P_j ($j = 1,2,3$) which belong to function F_1 are

$$WP^1 = (WP_1^1, WP_2^1, WP_3^1) = (0.4766, 0.3151, 0.2080)$$

Reliability allocations for programmes P_j ($j = 1,2,3$) which belong to function F_1 are

$$RP_1^1 = 0.9855, RP_2^1 = 0.9904, RP_3^1 = 0.9936$$

For accomplishing function F_2 , the comparison matrix AP_j^2 ($j = 1,2,3$) is depicted by Table 4 and the consistency ratio (CR) is 0.03, confirming acceptable consistency.

Table 4. Pairwise comparison matrix AP_j^2 .

Programmes	P_1	P_2	P_3
P_1	(1,1,1,1)	(1,2,3,4)	(2,3,4,5)
P_2	(1/4,1/3,1/2,1)	(1,1,1,1)	(1,1,2,3)
P_3	(1/5,1/4,1/3,1/2)	(1,2,3,4)	(1,1,1,1)

The normalised weight vector WP^2 of programmes P_j ($j = 1,2,3$) which belong to function F_2 are

$$WP^2 = (WP_1^2, WP_2^2, WP_3^2) = (0.5690, 0.2617, 0.1692)$$

Reliability allocations for programmes P_j ($j = 1,2,3$) which belong to function F_2 are

$$RP_1^2 = 0.9885, RP_2^2 = 0.9947, RP_3^2 = 0.9965$$

For accomplishing function F_3 , the comparison matrix AP_j^3 ($j = 1,2,3$) is depicted by Table 5 and the CR is 0.06, confirming acceptable consistency.

Table 5. Pairwise comparison matrix AP_j^3 .

Programmes	P_1	P_2	P_3
P_1	(1,1,1,1)	(2,3,4,5)	(4,5,6,7)
P_2	(1/5,1/4,1/3,1/2)	(1,1,1,1)	(1,2,3,4)
P_3	(1/7,1/6,1/5,1/4)	(1/4,1/3,1/2,1)	(1,1,1,1)

The normalised weight vector WP^3 of programmes P_j ($j = 1,2,3$) which belong to function F_3 are

$$WP^3 = (WP_1^3, WP_2^3, WP_3^3) = (0.6490, 0.2650, 0.1169)$$

Reliability allocations for programmes P_j ($j = 1,2,3$) which belong to function F_3 are

$$RP_1^3 = 0.9934, RP_2^3 = 0.9973, RP_3^3 = 0.9988$$

For accomplishing function F_4 , the comparison matrix AP_j^4 ($j = 1,2,3$) is depicted by Table 6 and the CR is 0.01, confirming acceptable consistency.

Table 6. Pairwise comparison matrix AP_j^4 .

Programmes	P_1	P_2	P_3
P_1	(1,1,1,1)	(1,1,2,3)	(2,3,4,5)
P_2	(1/3,1/2,1,1)	(1,1,1,1)	(1,2,3,4)
P_3	(1/5,1/4,1/3,1/2)	(1/4,1/3,1/2,1)	(1,1,1,1)

The normalised weight vector WP^4 of programmes P_j ($j = 1,2,3$) which belong to function F_4 are

$$WP^4 = (WP_1^4, WP_2^4, WP_3^4) = (0.5134, 0.3336, 0.1529)$$

Reliability allocations for programmes P_j ($j = 1,2,3$) which belong to function F_4 are

$$RP_1^4 = 0.9948, RP_2^4 = 0.9966, RP_3^4 = 0.9984$$

Reliability allocations (RP_j) for programmes P_j ($j = 1,2,3$) is calculated as

$$\begin{aligned} RP_1 &= \text{Maximum}(RP_1^1, RP_1^2, RP_1^3, RP_1^4) = 0.9985 \\ RP_2 &= \text{Maximum}(RP_2^1, RP_2^2, RP_2^3, RP_2^4) = 0.9973 \\ RP_3 &= \text{Maximum}(RP_3^1, RP_3^2, RP_3^3, RP_3^4) = 0.9988 \end{aligned}$$

For accomplishing programme P_1 , module M_1, M_2, M_3 and M_4 are taken as shown in Figure 3. For accomplishing programme P_1 , the pairwise comparison matrix AM_k^1 ($k = 1,2,3,4$) is depicted by Table 7 and the CR is 0.018, confirming acceptable consistency.

Table 7. Pairwise comparison matrix AM_k^1 .

Modules	M_1	M_2	M_3	M_4
M_1	(1,1,1,1)	(1,1,2,3)	(1,2,3,4)	(2,3,4,5)
M_2	(1/3,1/2,1,1)	(1,1,1,1)	(1,1,2,3)	(1,2,3,4)
M_3	(1/4,1/3,1/2,1)	(1/3,1/2,1,1)	(1,1,1,1)	(1,1,2,3)
M_4	(1/5,1/4,1/3,1/2)	(1/4,1/3,1/2,1)	(1/3,1/2,1,1)	(1,1,1,1)

The normalised weight vector WM^1 of modules M_k ($k = 1,2,3,4$) which belong to programme P_1 are

$$WM^1 = (WM_1^1, WM_2^1, WM_3^1, WM_4^1) = (0.4160, 0.2757, 0.1863, 0.1228)$$

Reliability allocations for modules M_k ($k = 1,2,3,4$) which belong to programme P_1 are

$$RM_1^1 = 0.9958, RM_2^1 = 0.9972, RM_3^1 = 0.9981, RM_4^1 = 0.9987$$

For accomplishing programme P_2 , the pairwise comparison matrix AM_k^2 ($k = 1, 2, 3, 4$) is depicted by Table 8 and the CR is 0.03, confirming acceptable consistency.

Table 8. Pairwise comparison matrix AM_k^2 .

Modules	M_1	M_2	M_3	M_4
M_1	(1,1,1,1)	(1,2,3,4)	(2,3,4,5)	(4,5,6,7)
M_2	(1/4,1/3,1/2,1)	(1,1,1,1)	(1,1,2,3)	(2,3,4,5)
M_3	(1/5,1/4,1/3,1/2)	(1/3,1/2,1,1)	(1,1,1,1)	(1,1,2,3)
M_4	(1/7,1/6,1/5,1/4)	(1/5,1/4,1/3,1/2)	(1/3,1/2,1,1)	(1,1,1,1)

The normalised weight vectors WM^2 of modules M_k ($k = 1, 2, 3, 4$) which belong to programme P_2 are

$$WM^2 = (WM_1^2, WM_2^2, WM_3^2, WM_4^2) = (0.5089, 0.2266, 0.1353, 0.1290)$$

Reliability allocations for modules M_k ($k = 1, 2, 3, 4$) which belong to programme P_2 are

$$RM_1^2 = 0.9948, RM_2^2 = 0.9977, RM_3^2 = 0.9986, RM_4^2 = 0.9987$$

For accomplishing programme P_3 , the pairwise comparison matrix AM_k^3 ($k = 1, 2, 3, 4$) is depicted by Table 9 and the CR is 0.04, confirming acceptable consistency.

Table 9. Pairwise comparison matrix AM_k^3 .

Modules	M_1	M_2	M_3	M_4
M_1	(1,1,1,1)	(2,3,4,5)	(1,2,3,4)	(2,3,4,5)
M_2	(1/5,1/4,1/3,1/2)	(1,1,1,1)	(1,1,2,3)	(1,2,3,4)
M_3	(1/4,1/3,1/2,1)	(1/3,1/2,1,1)	(1,1,1,1)	(1,1,2,3)
M_4	(1/5,1/4,1/3,1/2)	(1/4,1/3,1/2,1)	(1/3,1/2,1,1)	(1,1,1,1)

The normalised weight vector WM^3 of modules M_k ($k = 1, 2, 3, 4$) which belong to programme P_3 are

$$WM^3 = (WM_1^3, WM_2^3, WM_3^3, WM_4^3) = (0.4695, 0.2291, 0.1781, 0.1230)$$

Reliability allocations for modules M_k ($k = 1, 2, 3, 4$) which belong to programme P_3 are

$$RM_1^3 = 0.9952, RM_2^3 = 0.9977, RM_3^3 = 0.9986, RM_4^3 = 0.9987$$

Reliability allocation (RM_k) for modules M_k ($k = 1, 2, 3, 4$) is calculated as

$$RM_1 = \text{Maximum}(RM_1^1, RM_1^2, RM_1^3) = 0.9958$$

$$RM_2 = \text{Maximum}(RM_2^1, RM_2^2, RM_2^3) = 0.9977$$

$$RM_3 = \text{Maximum}(RM_3^1, RM_3^2, RM_3^3) = 0.9986$$

$$RM_4 = \text{Maximum}(RM_4^1, RM_4^2, RM_4^3) = 0.9987$$

For accomplishing module M_1 , submodule S_1, S_2, S_3 and S_4 are taken as shown in Figure 3. For accomplishing module M_1 , the pairwise comparison matrix AS_l^1 ($l = 1, 2, 3, 4$) is depicted by Table 10 and the CR is 0.025, confirming acceptable consistency.

Table 10. Pairwise comparison matrix AS_l^1 .

Submodules	S_1	S_2	S_3	S_4
S_1	(1,1,1,1)	(1,1,2,3)	(2,3,4,5)	(4,5,6,7)
S_2	(1/3,1/2,1,1)	(1,1,1,1)	(1,2,3,4)	(2,3,4,5)
S_3	(1/5,1/4,1/3,1/2)	(1/4,1/3,1/2,1)	(1,1,1,1)	(1,2,3,4)
S_4	(1/7,1/6,1/5,1/4)	(1/5,1/4,1/3,1/2)	(1/4,1/3,1/2,1)	(1,1,1,1)

The normalised weight vector WS^1 of submodules S_l ($l = 1, 2, 3, 4$) which belong to module M_1 are

$$WS^1 = (WS_1^1, WS_2^1, WS_3^1, WS_4^1) = (0.4678, 0.2978, 0.1510, 0.0812)$$

Reliability allocations for submodule $S_l (l = 1,2,3,4)$ which belong to module M_1 are

$$RS_1^1 = 0.9953, RS_2^1 = 0.9970, RS_3^1 = 0.9984, RS_4^1 = 0.9991$$

For accomplishing module M_2 , the pairwise comparison matrix $AS_l^2 (l = 1,2,3,4)$ is depicted by Table 11 and the CR is 0.018, confirming acceptable consistency.

Table 11. Pairwise comparison matrix AS_l^2 .

Submodules	S_1	S_2	S_3	S_4
S_1	(1,1,1,1)	(1,1,2,3)	(1,2,3,4)	(2,3,4,5)
S_2	(1/3,1/2,1,1)	(1,1,1,1)	(1,1,2,3)	(1,2,3,4)
S_3	(1/4,1/3,1/2,1)	(1/3,1/2,1,1)	(1,1,1,1)	(1,1,2,3)
S_4	(1/5,1/4,1/3,1/2)	(1/4,1/3,1/2,1)	(1/3,1/2,1,1)	(1,1,1,1)

The normalised weight vector WS^2 of submodules $S_l (l = 1,2,3,4)$ which belong to module M_2 are

$$WS^2 = (WS_1^2, WS_2^2, WS_3^2, WS_4^2) = (0.4160, 0.2757, 0.1863, 0.1228)$$

Reliability allocations for submodules $S_l (l = 1,2,3,4)$ which belong to module M_2 are

$$RS_1^2 = 0.9958, RS_2^2 = 0.9972, RS_3^2 = 0.9981, RS_4^2 = 0.9987$$

For accomplishing module M_3 , the pairwise comparison matrix $AS_l^3 (l = 1,2,3,4)$ is depicted by Table 12 and the CR is 0.022, confirming acceptable consistency.

Table 12. Pairwise comparison matrix AS_l^3 .

Submodules	S_1	S_2	S_3	S_4
S_1	(1,1,1,1)	(1,1,2,3)	(1,2,3,4)	(2,3,4,5)
S_2	(1/3,1/2,1,1)	(1,1,1,1)	(1,1,2,3)	(1,2,3,4)
S_3	(1/4,1/3,1/2,1)	(1/3,1/2,1,1)	(1,1,1,1)	(1,1,2,3)
S_4	(1/5,1/4,1/3,1/2)	(1/4,1/3,1/2,1)	(1/3,1/2,1,1)	(1,1,1,1)

The normalised weight vector WS^3 of submodules $S_l (l = 1,2,3,4)$ which belong to module M_3 are

$$WS^3 = (WS_1^3, WS_2^3, WS_3^3, WS_4^3) = (0.4160, 0.2257, 0.1853, 0.1228)$$

Reliability allocations for submodules $S_l (l = 1,2,3,4)$ which belong to module M_3 are

$$RS_1^3 = 0.9788, RS_2^3 = 0.9859, RS_3^3 = 0.9905, RS_4^3 = 0.9937$$

For accomplishing module M_4 , the pairwise comparison matrix $AS_l^4 (l = 1,2,3,4)$ is depicted by Table 13 and the CR is 0.033, confirming acceptable consistency.

Table 13. Pairwise comparison matrix AS_l^4 .

Submodules	S_1	S_2	S_3	S_4
S_1	(1,1,1,1)	(2,3,4,5)	(4,5,6,7)	(6,7,8,9)
S_2	(1/5,1/4,1/3,1/2)	(1,1,1,1)	(2,3,4,5)	(4,5,6,7)
S_3	(1/7,1/6,1/5,1/4)	(1/5,1/4,1/3,1/2)	(1,1,1,1)	(2,3,4,5)
S_4	(1/9,1/8,1/7,1/6)	(1/7,1/6,1/5,1/4)	(1/5,1/4,1/3,1/2)	(1,1,1,1)

The normalised weight vector WS^4 of submodules $S_l (l = 1,2,3,4)$ which belong to module M_4 are

$$WS^4 = (WS_1^4, WS_2^4, WS_3^4, WS_4^4) = (0.5780, 0.2597, 0.1119, 0.0502)$$

Reliability allocations for submodules $S_l (l = 1,2,3,4)$ which belong to module M_4 are

$$RS_1^4 = 0.9942, RS_2^4 = 0.9973, RS_3^4 = 0.9988, RS_4^4 = 0.9994$$

Reliability allocation (RS_l) for submodules $S_l (l = 1,2,3,4)$ is calculated as

$$RS_1 = \text{Maximum } (RS_1^1, RS_1^2, RS_1^3, RS_1^4) = 0.9958$$

$$\begin{aligned}
 RS_2 &= \text{Maximum}(RS_2^1, RS_2^2, RS_2^3, RS_2^4) = 0.9973 \\
 RS_3 &= \text{Maximum}(RS_3^1, RS_3^2, RS_3^3, RS_3^4) = 0.9988 \\
 RS_4 &= \text{Maximum}(RS_4^1, RS_4^2, RS_4^3, RS_4^4) = 0.9994
 \end{aligned}$$

Therefore using geometric mean method in reliability allocation, reliability goals are calculated for every software components prior to the design of the actual system. In this process we have achieved maximum reliability for our system. To ensure that these objectives are realistic and meaningful, it is essential to take into account the typical user requirements and the specifications regarding the reliability of software system and architecture of the software. This approach reduces the communication gap among users, programmers and software managers concerning the allocation of software reliability.

6. Comparison

In complex systems, human judgments are generally expressed in linguistic and vague terms rather than precise numerical values. Such uncertainty cannot be effectively handled using crisp methods. Therefore, fuzzy set theory provides a better framework to represent these imprecise and subjective evaluations. Although AHP is a widely used decision-making tool, it has several inherent limitations:

1. It is mainly applicable to crisp (non-fuzzy) decision environments.
2. It uses a fixed and sometimes unbalanced scale for pairwise comparisons.
3. It does not consider uncertainty in human judgments.
4. The ranking results may lack precision.
5. It is highly influenced by subjective preferences of decision-makers.

To overcome the above limitations, there is a need for a method that can incorporate uncertainty and vagueness present in human thinking. This leads to the adoption of the fuzzy analytic hierarchy process (FAHP).

To illustrate the effectiveness of the proposed approach, a numerical example is considered. Let, for the software system considered in this study, each submodule four criteria, namely T_1 , T_2 , T_3 and T_4 , are taken into account for reliability allocation. The fuzzy pairwise comparison matrix for these criteria is constructed using the proposed FAHP approach and is presented in Table 14.

Table 14. Pairwise comparison matrix for criteria.

Functions	T_1	T_2	T_3	T_4
T_1	(1,1,1,1)	(1,1,2,3)	(1,2,3,4)	(2,3,4,5)
T_2	(1/3,1/2,1,1)	(1,1,1,1)	(1,1,2,3)	(1,2,3,4)
T_3	(1/4,1/3,1/2,1)	(1/3,1/2,1,1)	(1,1,1,1)	(1,1,2,3)
T_4	(1/5,1/4,1/3,1/2)	(1/4,1/3,1/2,1)	(1/3,1/2,1,1)	(1,1,1,1)

To examine the consistency of the judgments, the matrix is first defuzzified and the corresponding consistency ratio is 0.0185, which is found to be within the acceptable limit. Using the proposed FAHP-based geometric mean method, the weights of the criteria are obtained as (0.4160, 0.2757, 0.1853, 0.1228). For the purpose of comparison, the same matrix is analyzed using the classical AHP method, yielding the weights (0.4131, 0.2770, 0.1872, 0.1225). It is observed that both methods produce nearly identical results and the same ranking of criteria, thereby validating the consistency and robustness of the proposed FAHP approach.

Furthermore, while AHP relies on precise numerical judgments, the proposed FAHP approach effectively incorporates uncertainty and vagueness inherent in human decision-making through trapezoidal fuzzy numbers. Hence, FAHP provides a more realistic and flexible framework, making it more suitable for reliability allocation in complex software systems.

7. Conclusion

This study presents a novel model for addressing the reliability allocation problem in software systems with the objective of achieving maximum reliability during the software development process. The proposed framework effectively integrates users' perspectives on software functionalities with the viewpoints of software engineers and programmers re-garding programs, modules, and submodules. A hierarchical structure is systematically developed in a fuzzy environment to decompose the problem from higher to lower levels. The FAHP, based on the geometric mean method with trapezoidal fuzzy numbers, is employed to derive the necessary parameters for reliability allocation. Furthermore, a comparative analysis between the classical AHP and the proposed FAHP approach has been carried out. The results indicate that both methods yield consistent rankings; however, FAHP provides a more realistic and flexible

representation by incorporating uncertainty and vagueness in human judgments. This demonstrates the robustness and effectiveness of the proposed model.

Future work may focus on extending the proposed model by incorporating other advanced fuzzy environments, such as interval-valued or q-rung orthopair fuzzy sets, to further enhance its capability in handling higher degrees of uncertainty. In addition, the model can be applied to large-scale real-world software systems and compared with optimization-based and reliability growth models to further validate its practical effectiveness.

All the Declarations and Statements

Author Contributions Statement

Ankita – Responsible for conceptualization, methodology, implementation, result analysis and drafting the manuscript, conducted the implementation of experiment, prepared the result, contributed to the literature review, and formatted and drafted the whole manuscript.

Deepak Kumar – Supervised the research work, reviewed the manuscript thoroughly, and gave constructive guidance to improve the findings easier to follow and increase the quality of research.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding Declaration

No outside funding was received for conducting this research.

Data Availability Statement

Data availability is not applicable.

Ethical Declarations

The research does not include any human or animal subjects, nor does it use personal data requiring ethical approval. Therefore, ethical approval was not required for this research.

Acknowledgments

We sincerely thank the experts for their professional evaluation and valuable recommendations, which have contributed to improving the quality of the experiment and the reliability of its results.

Declaration of Generative AI in Scholarly Writing

The authors made use of AI-assisted tools during the writing process to improve clarity, structure, and language. The authors ensured that all content was critically reviewed and approved for accuracy and originality.

Abbreviations

The following abbreviations are used in this manuscript:

FAHP - Fuzzy analytic hierarchy process

AHP - Analytic hierarchy process

References

- [1] Zahedi F., and Ashrafi N. (1991). Software reliability allocation based on structure, utility, price, and cost. *IEEE Transactions on Software Engineering*, 17, 345–356.
- [2] Poore J.H., Mills H.D., and Mutchler D. (1993). Planning and certifying software system reliability. *IEEE Software*, 10, 88–99.
- [3] Leung Y.W. (2004). Optimal reliability allocation for modular software systems designed for multiple customers. *IEICE Transactions on Information and Systems*, 12, 1655–1662.
- [4] Mirchandani C. (2018). Adaptive software reliability growth. *Procedia Computer Science*, 140, 122–132.
- [5] Pradhan S.K., Kumar A. and Kumar V. (2021). An optimal resource allocation model considering two-phase software reliability growth model with testing effort and imperfect debugging. *Reliability: Theory & Applications*, 16.
- [6] Luo H., Xu L. (2017). A novel software reliability growth model based on generalised imperfect debugging NHPP framework. *IEEE*.

- [7] Fang, C. C., Ma, L., & Kuo, W. (2025). Integrating human learning factors and bayesian analysis into software reliability growth models for optimal release strategies. *IEEE Access*.
- [8] Xie, D. (2025). Towards Improving Software Reliability through Modern Natural Language Processing Techniques (Doctoral dissertation, Purdue University Graduate School).
- [9] Samal, U., & Kumar, A. (2025). Empowering software reliability: Leveraging efficient fault detection and removal efficiency. *Quality Engineering*, 37(1), 118-129.
- [10] Hussain, A., Oraibi, Y., Mashikhin, Z., Jameel, A., Tashtoush, M., & Az-Zo'Bi, E. A. (2025). New software reliability growth model: Piratical swarm optimization-based parameter estimation in environments with uncertainty and dependent failures. *Statistics, Optimization & Information Computing*, 13(1), 209-221.
- [11] Dhavakumar, P., & Vengadeswaran, S. (2026). Software defect prediction using graph sample and aggregate-attention network optimized with nomadic people optimizer for enhancing the software reliability. *Computer Standards & Interfaces*, 95, 104033.
- [12] Cai, K.Y., Wen, C.Y., and Zhang, M.L. (1991). A critical review of software reliability modeling. *Reliability Engineering and System Safety*, 32, 357-371.
- [13] Cai, K.Y., Wen, C.Y., and Zhang, M.L. (1993). A novel approach to software reliability modelling. *Microelectronic and Reliability*, 33, 2256-2267.
- [14] Aggarwal K. K. and Singh Y. (1995). Software reliability apportionment using the analytic hierarchy process. *ACM SIGSOFT Software Engineering Notes*, 20, 56-61.
- [15] Chatterjee S., Singh J.B. and Roy A. (2013). A structure-based software reliability allocation using fuzzy analytic hierarchy process. *International Journal of Systems Science*.
- [16] Upadhyay, A., Yadav, D. K., & Yogi, K. K. (2022). FAHP based software reliability allocation concerning operational profile. *International Journal of System Assurance Engineering and Management*, 13(5), 2583-2593.
- [17] Babu, S., & Singh, R. (2024). Enhancing software reliability prediction using fuzzy AHP-based mathematical model and ANN integration. *International Journal of Information Technology*, 16, 4451-4464.
- [18] Yuen, K. K. F. (2024). Fuzzy cognitive network process for software reliability and quality measurement: comparisons with fuzzy analytic hierarchy process. *Journal of reliable intelligent environments*, 10(3), 319-336.
- [19] Aliyeva, K. R. (2025). Application of fuzzy AHP-TOPSA^oS hybrid method in facility location selection for software systems. *Ultimatics: Jurnal Teknik Informatika*, 17, 34-41.
- [20] Khan, R. A., Al Hashimi, H. A., Alwageed, H. S., Keshta, I., Almagrabi, A. O., & Ayouni, S. (2025). Securing Software Development Through People Maturity: A Fuzzy-AHP Decision-Making Framework. *Journal of Software: Evolution and Process*, 37(9), e70045.
- [21] Khan, R. A., Keshta, I., Al Hashimi, H. A., Almagrabi, A. O., Alwageed, H. S., & Alzahrani, M. (2025). A fuzzy-AHP decision-making framework for optimizing software maintenance and deployment in information security systems. *Journal of Software: Evolution and Process*, 37, e2758.
- [22] Zadeh L.A. (1965). Fuzzy sets. *Information Control*, 8, 338-353.
- [23] Dubois D. and Prade H., (1978). Operations on fuzzy numbers. *International Journal of Systems Science*, 9, 613-626. Lee H. (1993). A structure methodology for software development effort prediction using the analytic hierarchy process. *Journal of System and Software*, 21, 179-186.
- [24] Saaty T.L. (1980). *The Analytic Hierarchy Process: Planning, Priority Setting, Resources Allocation*, London: McGraw-Hill.
- [25] Van Laarhoven P.J.M., and Pedrycz W. (1983). A fuzzy extension of Saaty's priority theory. *Fuzzy Sets and Systems*, 11, 229-241.
- [26] Buckley J. J. (1985). Fuzzy hierarchical analysis. *Fuzzy Sets and Systems*, 17, 233-247.
- [27] Chang D.Y. (1996). Applications of extent analysis method on fuzzy AHP. *European Journal of Operational Research*, 95, 649-655.

Authors' Profiles



Ankita was born in Karnaprayag, Uttarakhand, India. She received her Bachelor's degree from Hemvati Nandan Bahuguna Garhwal University, Uttarakhand in 2021, and her Master's degree in Mathematics from Sri Dev Suman Uttarakhand University, Uttarakhand, India in 2023. Currently, she is pursuing her Ph.D. in Mathematics from Kumaun University, Nainital, India. Her major field of research is fuzzy set theory, fuzzy reliability theory and decision-making. She is actively engaged in research in fuzzy reliability modeling and its applications in reliability analysis of complex systems.



Deepak Kumar is an associate professor in the Department of Mathematics, Kumaun University, Nainital, Uttarakhand, India. He received his Bachelor's, Master's and Ph.D. degree from G.B. Pant University of Agriculture and Technology, Uttarakhand, India. He has more than fifteen years of teaching and research experience and has published over 80 research papers in reputed SCI/Scopus-indexed journals, along with book chapters and conference publications. He has successfully supervised Ph.D. scholars and is actively involved as a reviewer for various reputed international journals. His research interests include fuzzy set theory, reliability theory, and their applications in reliability analysis of complex systems. Dr. Kumar has delivered invited lectures in his area of expertise and has contributed to organizing academic events such as conferences, seminars, and workshops. He has received numerous awards and recognitions for his contributions to research and academia.

How to cite this paper:

Ankita, Deepak Kumar, "Software Reliability Allocation using FAHP and Trapezoidal Fuzzy Set Theory", International Journal of Mathematical Sciences and Computing(IJMSC), Vol.12, No.3, pp. 24-38, 2026. DOI: 10.5815/ijmsc.2026.03.02