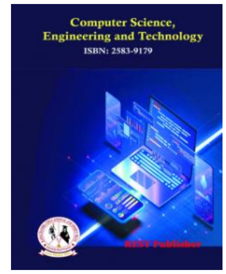
Computer Science, Engineering and Technology

Vol: 3(1), March 2025

REST Publisher; ISSN: 2583-9179 (Online)

Website: https://restpublisher.com/journals/cset/

DOI: https://doi.org/10.46632/cset/3/1/8



Evaluating the Interoperability and Relative Importance of Software Development Life Cycle Components Using Grey Relational Analysis

***Umesh K. Joshi, Vidya A. Hingmire**

Sandip Foundation's Sandip Institute of Technology and Research Centre, Nashik, Maharashtra, India.

* Corresponding Author Email: umeshkj2001@gmail.com

Abstract: *The software development life cycle (SDLC) encompasses diverse aspects, including design, requirements, methodologies, programming languages, tools, frameworks, and user interface design. Effective interoperability and integration among these components are crucial for streamlining workflows, mitigating compatibility issues, and delivering robust software solutions. This study employs the Grey Relational Analysis (GRA) method to evaluate the interoperability and relative importance of these SDLC elements in software development endeavors. The analysis reveals that fostering seamless collaboration among design, requirements, methodologies, languages, tools, and frameworks is essential for optimizing development processes and enhancing overall efficiency and effectiveness. The normalized data highlight the varying degrees of influence each component exerts across different SDLC phases, underscoring the need to recognize and leverage these interdependencies. The deviation sequence, obtained through the GRA method, unveils the relative importance or effectiveness of various elements, providing insights for informed decision-making and identifying areas for potential improvement or optimization. Furthermore, the Grey Relation Coefficient quantifies the correlations between different facets of software development, enabling stakeholders to prioritize areas for improvement and focus attention on elements with the most substantial influence. Notably, the final Grey Relational Grade (GRG) Rank reveals that testing frameworks secured the top rank, suggesting their significant impact on overall development quality, followed by programming languages and user interface design. These rankings offer valuable insights for prioritizing efforts and allocating resources, ultimately enhancing development processes and achieving better project outcomes. The study underscores the importance of a holistic and integrated approach to software development, recognizing the interdependencies among various components and leveraging the insights derived from the GRA method. By embracing this approach, software teams can make informed decisions, streamline workflows, and deliver high-quality, user-centric solutions that meet industry standards and drive success in an increasingly competitive digital landscape.*

Key words: *Software Development Life Cycle (SDLC), Grey Relational Analysis (GRA), Interoperability, Software Development, Components, Testing Frameworks, Programming Languages, User Interface Design, Development Methodologies, Project Management and Decision-making.*

1. INTRODUCTION

SDLC stands for Software Development Life Cycle, a term used to describe the stages through which software evolves. Some argue that SDLC refers specifically to the life cycle of software, while others believe it includes the development of systems. This article focuses on software development, although the same principles can be applied to computer systems. Much of the innovation and thought leadership in model and concept design originates from software development, with computer development drawing heavily from these advancements. It's important to understand that a model serves a different purpose. While the former explains what should be done, the latter outlines how to do it, making the method both descriptive and systematic. Thus, in this article, we examine SDLC models and their relevance to specific types of software projects. This approach considers the context in which an SDLC model is used. For instance, the waterfall model may be ideal for developing an enterprise relational database, but it might not be the best choice for creating an internet-based application [1]. In software development, both topical and episodic knowledge are essential. Topical knowledge encompasses understanding the specific domain relevant to the project, which is crucial for software developers. However, developers might face challenges if they only possess topical knowledge and lack practical experience. This experience, or episodic knowledge, often involves familiarity with real-world application and nuances of the domain. For instance, developers may design software that appears well-structured but fails to perform effectively due to inappropriate implementation choices. At the coding level, a deficiency in episodic knowledge of the programming language can lead to overly complex and inefficient code. This highlights the importance of both theoretical understanding and practical experience in creating effective software solutions. The quality of software design derived from a particular method depends on the designer's episodic knowledge, which can vary widely. This knowledge might be acquired through a book, a crash course, or other learning methods focused on the topic. Some instructions may necessitate more episodic knowledge than others, and it is essential to assess the required amount of this knowledge when wanting to evaluate the quality of the method or the software design [2]. High-performance computers are extensively used for developing application software, especially for scientific and engineering purposes. This type of software is often discussed in the software engineering literature; though specific reports may not always be available. It enables the simulation of physical phenomena, processing of large datasets, and execution of complex calculations—tasks that traditional software development methods cannot adequately address. The unique characteristics of this software make its study both interesting and valuable. A third distinctive feature is that developers in this field are "process" oriented and tend to avoid improvisational methods. This aversion, while posing risks in other software programs where improvisation can be successful, stems from the formal nature of these applications and a general lack of software engineering training. Additionally, many projects in this domain have long life cycles, often spanning decades, during which software evolves alongside advancements in scientific knowledge. As a result, developers believe they cannot afford flexibility and adhere strictly to rigorous software development processes [3]. Before developers promote and deploy a new technology on a large scale, it must be validated within a specific context to ensure its effectiveness. Most companies, despite having similar needs, require compelling evidence of a new method's success before adopting it. This is particularly crucial for large companies, which need to coordinate new technologies and processes with their existing systems due to their complexity. To evaluate the suitability of agile practices and other development methods, many organizations, including members of the Software Experience Center (SEC), have initiated a series of operations. These evaluations help determine whether the practices meet the company's needs and can be effectively integrated [4]. The rapid evolution of software development impacts all domains, necessitating high-accuracy parameters for successful development. Creating effective models is challenging, and the costs associated with software development and training continue to rise. This unpredictability raises concerns within the software engineering community. One of their key objectives is to develop accurate models for predicting the software development lifecycle and costs. Over the past two decades, pioneering research efforts have led to the evolution of various software evaluation models. Among these, classical multiple regression techniques are commonly used. However, as discussed in this paper, these classical models may not always be the most effective when applied to software engineering data [5]. Product managers typically work on multi-year product plans, coordinating with marketing and engineering for specific product releases. Despite the desire to include numerous features, constraints often exist. CEO Bill Gates, along with senior managers and other executives, closely monitored key projects by participating in quarterly project reviews. Additionally, project managers submitted monthly status reports, and administrators reviewed the progress in relation to the divisions' three-year production plans. Most projects enjoyed considerable freedom, including access to as many developers and resources as needed for hiring testers. This financial independence was crucial, particularly for projects like Internet Explorer. Ben Slivka, who managed the first three IE projects, mentioned that there were no budgetary constraints for these projects [6]. Over the past two decades, the management, development, and maintenance of software have transitioned from being concentrated in one geographic location to being distributed globally. This shift is often referred to as "global," "distributed," or "multi-platform" software development. Several business factors have driven this trend. Initially, the global demand for software products and services surged in the late 1980s, prompting IT companies to expand into new markets and diversify their product lines. Simultaneously, many companies began to focus on their core competencies, choosing to outsource or "offshore" other activities to specialists. Offshoring offered significant advantages, such as access to a large pool of skilled labor, the ability to develop around the clock, and most importantly, substantial cost savings due to lower labor costs in developing countries [7]. In the Software Development segment in Europe, America, Asia, and Australia, more than 100 development centers worldwide handle thousands of distributed software projects. These efforts are managed by a central R&D team consisting of engineers. This setup combines the classic matrix structure with strong operational and project organizations, focusing on long-term skills and technical evolution. Telecommunications providers have operated in a global environment for years. Initially, the primary drivers were local customization and after-sales service, which demonstrated job creation for new local customers and justified contracts. Internationally, the focus on distributed software development has led to an increase in resources, new markets, and products through acquisitions and mergers, enhancing the existing team of engineers and fostering creativity [8]. This transformation affects marketing and distribution, not only in terms of product concepts but also in designing, building, testing, and delivering products to customers. It has a significant impact globally. The processes of distributed software development are adopted by numerous companies to enhance profits, productivity, cost-efficiency, and quality. Consequently, many software development platforms now embrace multicultural, globally distributed teams of engineers, managers, and administrators, who face substantial challenges related to technology, society, and culture [9]. Nabulus focuses on agile software development and emphasizes the importance of personal and situational interpretation. The challenge lies in dismantling formal mechanisms, which can perpetuate misconceptions and create a fertile ground for advocates to propagate these errors in a non-productive cycle. When dealing with ambiguous topics, it is crucial to understand the boundaries and intricacies within those limits personally. The second issue stems from poor dissemination and inadequate synthesis of fragmented research findings. Over the years, our understanding of the various features of agile software development has not significantly expanded. Despite ongoing research efforts, extracting the nuanced details of software development remains a complex task. Unfortunately, the scattered results of these efforts are not well-publicized or readily accessible to readers. Consequently, the global community remains unaware of the collective knowledge about software development [10]. It is a collection of software development methodologies that emphasize self-organization and continuous improvement among cross-functional teams. Formed through collaboration, these methods aim to deliver high-quality software that meets customer needs, allowing for rapid delivery and adaptability to changing requirements. In this paper, we identify dynamic factors that significantly impact the described development approach, enhancing the software development process to meet evolving business environments. We briefly compare agile development methods with traditional systems development methods and discuss the current status of agile practices. We propose that agile software development is emerging as a valuable alternative to traditional approaches, driven by the need to continuously deliver valuable software and satisfy customer requirements [11]. In the early stages of software development, extensive communication is crucial. There are two main types of communication required for software projects. Firstly, formal communication is essential, necessitating a clear and understandable interface for official interactions. This includes updating project statuses, escalating issues, and identifying responsibility for work products. An unclear or poorly defined interface can lead to wasted time and unresolved problems in these critical tasks. Poor documentation often leads to ineffective joint development among developers. This resistance to documentation is a widely acknowledged issue, yet its importance cannot be overstated. In global software development (GSD), there exists a diverse array of challenges beyond just creating documentation. It's not only about initially documenting artifacts, but also about consistently updating and revising them. This ongoing process is crucial for eliminating assumptions and ambiguity and ensuring maintainability. Documentation must accurately reflect the needs and practices of various groups involved and adapt to how they operate [12]. Secondly, numerous commercial software firms safeguard the "source code" of their software, which comprises a sequence of instructions interpreted by the computer to accomplish the program's objectives. Programmers compose computer software in the form of source code, outlining the purpose of each segment and providing a concise written explanation of the design. They document this source code for their project. To convert a program into a format executable by a computer, a source code compiler translates it into machine code using a software tool known as a compiler [13]. Trade publications highlight a notable shift from conventional software to Software as a Service (SaaS). While some view this transition favorably, acknowledging its differences from traditional software, significant variances have been identified. Business magazines emphasize the absence of large upfront investments in SaaS, impacting both buyers' and sellers' cash flows. Instead, SaaS operates on smaller, consistent cash flows over extended periods. Additionally, data security concerns arise as users' data resides on vendors' servers, posing potential risks. The University of Florida outlines cost savings, enhanced bargaining power, and the flexibility to switch providers as key benefits of adopting SaaS [14].

2. MATERIALS AND METHOD

Development Methodologies: The development process within a project life cycle is multifaceted, encompassing various phases. Some of these phases are optional, and the order of the phases depends on the specific project being executed.
Programming Languages: A programming language is a tool that programmers use to communicate with computers. It allows for various methods of creating machine code string values and contains a set of rules or syntax. In some cases, programming languages also incorporate graphical elements.
Development Tools: Software development tools are applications designed to run on personal computers, aiding programmers or system developers in creating, modifying, or testing application programs. These tools often include text editors and assemblers/compilers.
Testing Frameworks: Test automation frameworks are systems of rules and a collection of related tools designed to create events and streamline testing processes. They are engineered to enhance operational efficiency.
User Interface Design: User interface (UI) design involves crafting the look and feel of interfaces in digital devices, focusing on style and usability. This process is undertaken by designers with the goal of making these interfaces easy and enjoyable for users. UI design encompasses graphical user interfaces as well as other formats, such as voice-controlled interfaces.
Design: Design is a visual solution to problems, involving textual and/or graphical components. It seeks to attract attention through aesthetically pleasing and engaging elements.
Requirements: A requirement is a characteristic or statement describing a product or process. It can be functional, non-functional, or design-related, and may also identify any constraints. Throughout the project's lifecycle, various stakeholders, including users, system engineers, and program managers, generate different types of requirements.
Project: A plan is a strategy devised to achieve a specific set of goals within a defined time frame. It comprises a series of tasks, which are carried out by a group of designated individuals. This group is led by the project manager, who is responsible for planning, scheduling, monitoring progress, and ensuring successful completion.
Process: A process embodies a distinct outcome. It involves a sequence of steps aimed at initiating peace operations promptly. I fully support this approach. They've been gradually expanding over the years, so it's wise to persist. The method involves eliminating options systematically.
GRA Method: A decision-making group with expertise across multiple criteria will be responsible for reaching a conclusion. In various fields such as management, factors like problems, economics, and engineering carry significant weight. Typically, these criteria are evaluated and alternative approaches are considered based on their respective importance, quantified through real numbers. However, owing to the complexity of real-world systems, decision-makers often face challenges in arriving at the optimal choice due to the uncertainties inherent in day-to-day operations. Addressing these uncertainties and ambiguities is crucial, and one approach that proves valuable is intuitive fuzzy set theory, an extension of fuzzy set theory that incorporates membership and non-membership grades. While fuzzy set theory has been extended in diverse ways by different scholars, the imprecise nature of Intuitive Fuzzy Set (IFS) theory presents a significant challenge for modeling purposes. Nevertheless, IFS theory garners attention from many researchers because of its relevance in dealing with uncertainty, and its definition of various aggregation operators facilitates the synthesis of information from multiple sources [15]. The optimization of multiple response characteristics, rather than single response properties, is a complex endeavor. In a study conducted by Dharma lingam et al., they utilized the Taguchi method combined with Gray Relational Analysis (GRA) to investigate the dry sliding wear behavior of an AlSi10Mg-Al2O3-MoS2 hybrid composite. They focused on improving multiple response characteristics related to wear behavior. The study identified the weight percentage of MoS2, as well as the sliding velocity (SV) and applied load (AL) parameters, as crucial factors influencing the Specific Wear Ratio (SWR) and coefficient of friction (COF) under optimal conditions. The addition of red mud to aluminum alloys was also analyzed for its impact on SWR and COF. Through the Taguchi method coupled with GRA, the researchers optimized the SWR and COF, with notable influences observed from parameters such as red mud content and sliding speed. Additionally, the study highlighted the significance of hardness and applied load in determining wear behavior [16]. Various methods for multi-criteria decision-making have been established to identify the most suitable solution based on specific criteria. However, these processes often encounter a challenge as the criteria frequently conflict with each other, making it difficult to find a satisfactory solution that satisfies all criteria simultaneously. Despite the collective aim of researchers to devise simple, reliable, and accurate solutions, the inherent complexity and uncertainty of real-life scenarios render many existing processes inadequate for addressing all problems. While numerous processes have been introduced by researchers, a significant drawback is the lack of consideration for method compatibility. Most

methods fail to adequately handle both non-cardinal and cardinal data. Hence, the primary objective of this study is to address this issue by proposing a coherent system that simplifies the decision-making process and adapts to different situations. The proposed method combines superiority and inferiority ranking (SIR) with the Grey Relational Analysis (GRA) technique. This novel approach, introduced for the first time by the authors, marks a significant advancement in multi-criteria decision-making methodology, offering a more comprehensive and effective solution framework [17]. This issue impacts the fundamental principles of supply chain management, which involve the coordination and synchronization of various elements such as units, materials, information, and capital flow throughout the supply chain. Evaluating the performance of the supply chain poses a significant challenge due to its vast scale and the independence of units across the chain. This limitation in performance appraisal is a crucial consideration in strategic decision-making, as it necessitates a comprehensive approach that encompasses the entire spectrum of supply chain activities, including marketing, distribution, planning, production, and purchasing. Traditionally, these units within the supply chain have operated independently, but the increasing significance of procurement processes underscores the heightened sensitivity of outcomes to strategic decisions and procurement activities. Supplier selection, in particular, emerges as a critical aspect of effective supply chain management, with its implications extending to profitability and long-term success. Consequently, many companies are increasingly focusing on enhancing the capabilities of their suppliers to remain competitive amidst factors such as shortened product life cycles, evolving customer demands, and rapid shifts in consumer preferences. This emphasis on supplier improvement underscores the critical importance of assessment and selection processes in supply chain management [18]. The process of selecting materials is a balancing act, weighing both cost and environmental concerns alongside material properties. There's a significant trade-off involved, considering various factors and vulnerabilities. This process involves exchanges and integrals, with a focus on solving problems by considering the entire product life cycle, particularly the end-of-life stage. This approach involves implementing strategies for product disposal while also assessing environmental impacts. This paper proposes a comprehensive approach to material selection, integrating considerations such as problem characteristics, end-of-life strategies, and cost. It emphasizes the importance of selecting materials that align with disposal strategies. The study aims to explore the relationship between these factors using Correlation Analysis (GRA), a method particularly suited for analyzing complex systems and relationship similarities [19]. GRA, or Grey Relational Analysis, represents a method of non-linear data interpolation. It addresses communication challenges by leveraging various input variables such as cultivated land area, fertilizer usage in agriculture, machine power, agricultural reservoirs, and disaster-affected areas to project food production. While numerous factors influence food production, GRA focuses on grain yield and examines the intricate relationship between these factors. It identifies significant influencing factors through normalization and correlation coefficients, retaining only the most relevant predictors to enhance model accuracy. This approach ensures fair representation by eliminating factors with minimal direct impact on outcomes [20]. On the flip side, achieving a high material removal rate (MRR) in production is highly sought after in the industry because it allows for the rapid removal of mass while maintaining product quality. This is particularly crucial in processes like turning, where factors such as feed rate, cutting depth, and speed influence MRR. Increasing these variables can indeed enhance MRR, but it also leads to higher power demands, often exceeding the capabilities of existing machine tools. Therefore, selecting the optimal parameters in the manufacturing process is vital for ensuring both product quality and high MRR, ultimately enhancing the efficiency and effectiveness of manufacturing, which in turn plays a significant role in the economy [21]. A highly effective method for evaluating the precision of a drilled hole is crucial, as it directly impacts several key aspects. For instance, even minor deviations in the drilling process can lead to decreased durability and reliability of the final product due to increased hardness, ultimately affecting production costs. Thus, prioritizing hardness control becomes paramount for enhancing machine quality. To achieve this, continuous exploration and refinement of various drilling parameters are essential. Andrey Belyakov explores this extensively, delving into the intricate structure and mechanical behavior of metals and their alloys. Additionally, Om Prakash Singh and colleagues delve into optimizing shoulder grinding operations using the GRA method, particularly focusing on parameters for Al 6063. Their research underscores the significance of factors like cutting and feed rates in achieving optimal results, as evidenced by their findings [22]. From the available literature on PCA-GRA, it's commonly acknowledged that most studies focus on process parameters, particularly those pertaining to continuous manufacturing processes and controllable data. However, there's a notable dearth of literature concerning design parameters. The bellows, as a mechanical component, presents a distinctive case due to its expansion and contraction functionalities, which are inherent to its design. These characteristics introduce unique and uncontrollable variables during its operation [23]. Modern heavy industries are seeking innovative solutions to lighten the load while ensuring durability and eco-friendliness. In this quest, robust and lightweight load cells have become paramount. Today, there's a burgeoning exploration of novel techniques to achieve this goal, with a focus on advanced methodologies like optimization through numerical simulations. These cutting-edge approaches, already implemented in some production lines, leverage tools like finite element analysis and computer-based numerical simulations. By doing so, they substantially enhance efficiency and effectiveness. Such methods allow for a precise measurement of load variations under diverse environmental conditions, thereby improving the accuracy of load cells, often measured as a percentage of their total capacity [24]. he evaluation and enhancement of novel Multiple Attribute Decision Making (MADM) approaches for route selection involve extensive research and development efforts. Initially, a comprehensive review of existing studies is conducted to establish an evaluation framework for the criteria. This review yields an initial list of indicators, which is subsequently refined through consultations with road construction experts and Urban Development Panels in multiple meetings. Key indicators are identified through these discussions and are assigned varying levels of importance based on expert input and frequency of mention. Following this, a total of 35 indicators are carefully chosen for the Optimum Variation Method. Each indicator is then weighted using a combination of expert opinions and dynamic fuzzy Grey Relational Analysis (GRA). Through this methodological approach, the optimal trajectory variation is determined from four potential candidates, ensuring a systematic and informed decision-making process in route selection [25]. The aim of this study is to explore the multivariate normality and equality of variance matrices. Unlike previous analyses, this study focuses on empirical investigation rather than speculation, aiming to validate the accuracy of methods used for analyzing data covariance structures. Specifically, it seeks to improve model selection in PROC MIXED. While feasibility and information criteria are options, this article primarily focuses on defining the mean structure and assessing assumptions on parameters. It evaluates both patterned and non-patterned covariance structures using Restricted/Residual Maximum Likelihood (REML) estimation [26]. In other types of studies, such as those involving emitters connected to Nano antennas or forecasting the quantum yield or nonlinear characteristics of opt mechanical systems, understanding beyond the spectral range is crucial. This entails quantifying the coupling methods and quasi-normal modes (QNMs), along with employing a hybrid strategy that combines actual computations with frequency-based analysis. By augmenting QNM investigation with additional computational procedures, we can effectively address the necessity of considering various methodologies. The underlying principle is to establish a standard procedure, although numerical techniques and QNMs may sometimes affect the outcome within the spectral domain. Accurate interpolation from given data points becomes pivotal in such scenarios [27]. For widespread adoption of IoT across India, addressing implementation challenges is imperative. A comprehensive examination of these hurdles is essential to pinpoint the most critical issues. Prioritization through ranking is necessary to determine the most significant challenges. Consequently, decision-making becomes crucial, akin to methods like GRA and AHP. Resolving this decision-making challenge involves employing key strategies. IoT encompasses various devices ranging from smartphones, tablets, and bicycles to cars, trains, refrigerators, washing machines, and other electronic devices equipped with sensors. The full-fledged implementation of IoT will have a profound impact on both individuals and businesses [28]. This research field has seen several notable contributions, such as addressing CO2 emissions and identifying the factors influencing them. Indigeneity concerns, particularly regarding CO2 emissions in Pakistan, have been examined. Additionally, efforts have been made to analyze the average impact of emissions, utilizing innovative mathematical models like gray correlation analysis (GRA) and conservative approaches like Maxim in. The study employed the criterion method, focusing on various sectors such as transport and industry, as well as energy consumption in CO2-emitting sectors, considering their implications for economic and population growth in Pakistan. Through our analysis, we aim to uncover new insights into the relationships between these factors. This study not only offers domestic insights but also aligns with existing research. By conducting tests to ascertain which factors contribute more significantly to CO2 emissions, we anticipate achieving more accurate results with the aid of novel modeling techniques [29].

3. ANALYSIS AND DISCUSSION

TABLE 1. Software development

	Design	Requirements	Project	Process
Development Methodologies	22.10	141.21	85.24	88.79
Programming Languages	20.22	154.23	65.29	11.22
Development Tools	33.13	121.16	45.27	65.42
Testing Frameworks	40.44	186.55	26.43	90.99
User Interface Design	66.90	117.56	71.91	50.16

Interoperability in software development, as depicted in Table 1, necessitates seamless interaction among various components. It involves ensuring that design aligns with requirements, development methodologies complement programming languages, and tools integrate with testing frameworks. Effective interoperability facilitates smooth project management, enabling efficient allocation of resources and adherence to timelines. For instance, integrating user interface design with development tools enhances collaboration and ensures user-centric product development. By fostering interoperability across these domains, software teams can streamline workflows, mitigate compatibility issues, and deliver robust, integrated solutions to meet user needs effectively.

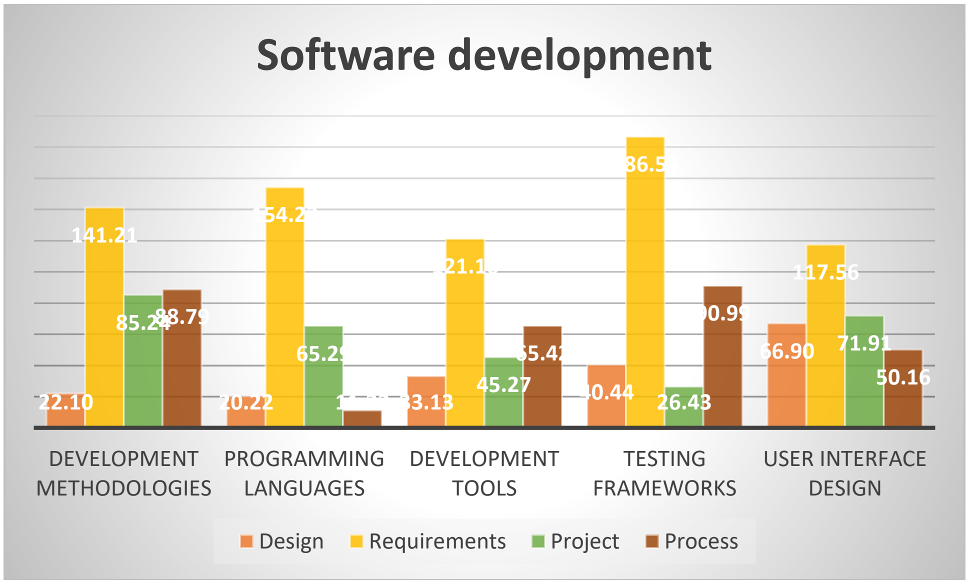


FIGURE 1. Software development

Interoperability within software development, as illustrated in Figure 1, emphasizes the seamless collaboration among design, requirements, project management, and process execution. It involves integrating methodologies, languages, tools, and frameworks to optimize development workflows and ensure cohesive product outcomes, ultimately enhancing efficiency and effectiveness in software engineering endeavors.

TABLE 2. Normalized Data

	Design	Requirements	Project	Process
Development Methodologies	0.0403	0.3428	0.0000	0.0276
Programming Languages	0.0000	0.5315	0.3392	1.0000
Development Tools	0.2766	0.0522	0.6796	0.3205
Testing Frameworks	0.4332	1.0000	1.0000	0.0000
User Interface Design	1.0000	0.0000	0.2267	0.5118

Interoperability, as depicted in Table 2, underscores the balanced integration of diverse elements in software development. Here, normalized data highlight the varying degrees of influence each component has on different phases. For instance, user interface design holds significant weight in the design phase but less in project management. Conversely, testing frameworks are critical across requirements and project phases. Effective interoperability necessitates recognizing these nuances, ensuring seamless interaction between components. By understanding and leveraging these interdependencies, software teams can optimize processes, enhance collaboration, and deliver cohesive, high-quality products that meet user expectations and industry standards efficiently.

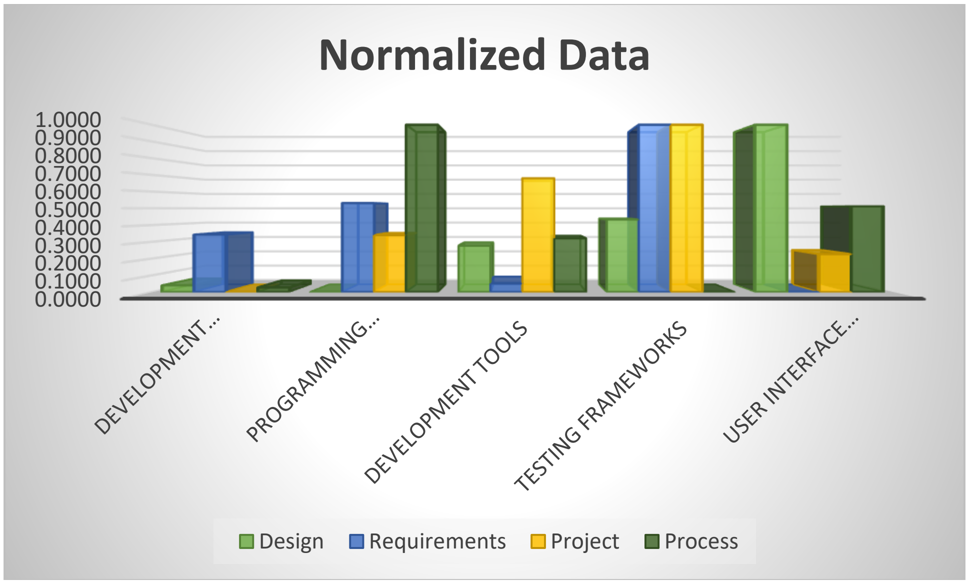


FIGURE 2. Normalized data

Figure 2, data normalized via the GRA method showcases interrelations among various aspects of software development. Each column denotes a distinct facet such as design, requirements, project, and process. Through normalized values, the table elucidates the relative importance or impact of different methodologies, languages, tools, frameworks, and user interface design within the development spectrum.

TABLE 3. Deviation sequence

	Design	Requirements	Project	Process
Development Methodologies	0.9597	0.6572	1.0000	0.9724
Programming Languages	1.0000	0.4685	0.6608	0.0000
Development Tools	0.7234	0.9478	0.3204	0.6795
Testing Frameworks	0.5668	0.0000	0.0000	1.0000
User Interface Design	0.0000	1.0000	0.7733	0.4882

Table 3, utilizing the GRA method, presents the deviation sequence for various aspects of software development: design, requirements, project, and process. Each entry reflects the degree of deviation from the ideal state, offering insights into the relative importance or effectiveness of different elements. For instance, in development methodologies, a higher value indicates greater deviation, suggesting areas for potential improvement or optimization. Similarly, the sequence unveils nuances in programming languages, development tools, testing frameworks, and user interface design, facilitating informed decision-making and enhancing overall development efficiency through the understanding and leveraging of these interrelations.

TABLE 4. Grey Relation Coefficient

	Design	Requirements	Project	Process
Development Methodologies	0.3425	0.4321	0.3333	0.3396
Programming Languages	0.3333	0.5163	0.4307	1.0000
Development Tools	0.4087	0.3453	0.6095	0.4239
Testing Frameworks	0.4687	1.0000	1.0000	0.3333
User Interface Design	1.0000	0.3333	0.3927	0.5060

Table 4 employs the Grey Relation Coefficient (GRA) method to quantify the relationship between various facets of software development: design, requirements, project, and process. Each cell represents the degree of correlation between the corresponding rows and columns. For example, in development methodologies, a coefficient closer to 1 indicates a stronger correlation, suggesting a significant impact on other aspects. Conversely, lower coefficients suggest relatively weaker correlations. By interpreting these coefficients, stakeholders can prioritize areas for improvement or focus attention on elements with the most substantial influence, optimizing the overall development process and enhancing project outcomes through informed decision-making.

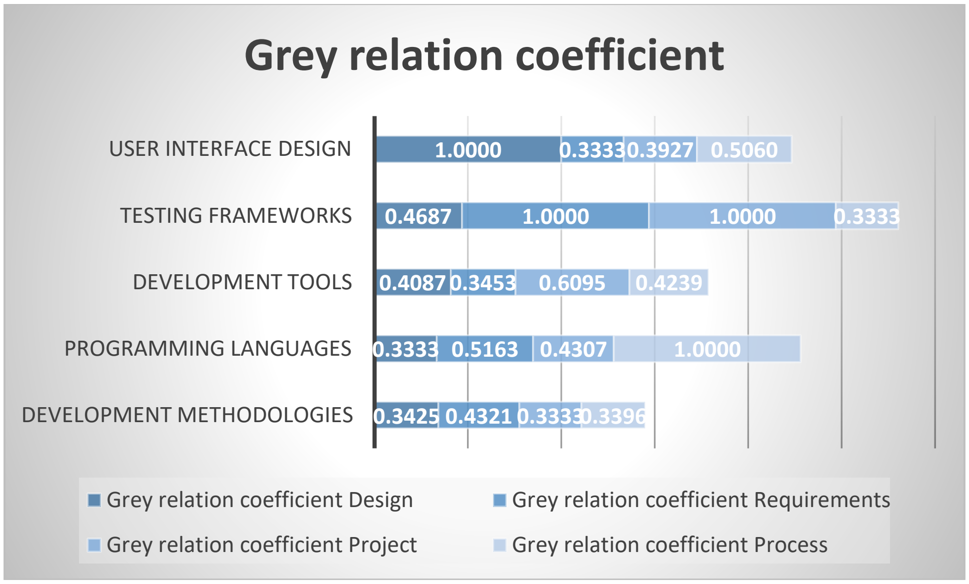


FIGURE 3. Grey relation coefficient

In Figure 3, the Grey Relation Coefficient (GRA) method illustrates correlations among software development facets: design, requirements, project, and process. Values in each cell denote the degree of relationship between respective rows and columns. These coefficients aid in identifying influential factors and optimizing development strategies, ultimately enhancing project efficiency and outcomes.

TABLE 5. Result of final GRG Rank

	GRG	Rank
Development Methodologies	0.3619	5
Programming Languages	0.5701	2
Development Tools	0.4469	4
Testing Frameworks	0.7005	1
User Interface Design	0.5580	3

Table 5 displays the results of the final Grey Relational Grade (GRG) Rank, computed using the GRA method, for different aspects of software development. The GRG values indicate the overall performance or effectiveness of each facet relative to others, with higher values representing better performance. Corresponding ranks highlight the relative standing of each aspect within the development context. For instance, testing frameworks achieved the highest GRG value, securing the top rank, suggesting its significant impact on overall development quality. These rankings offer valuable insights for prioritizing efforts and allocating resources, thereby enhancing development processes and achieving better project outcomes.

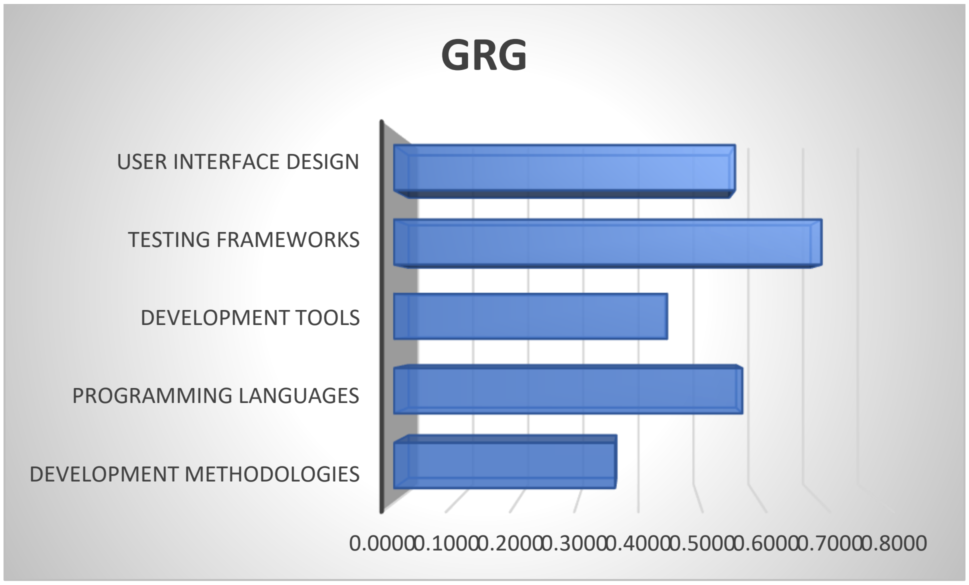


FIGURE 4. GRG

Figure 4 illustrates the Grey Relational Grade (GRG) results obtained through the GRA method for various software development aspects. GRG values represent the overall effectiveness, with higher scores indicating better performance. Corresponding ranks demonstrate the relative standings of each facet, aiding in prioritizing efforts and optimizing development strategies for enhanced project outcomes.

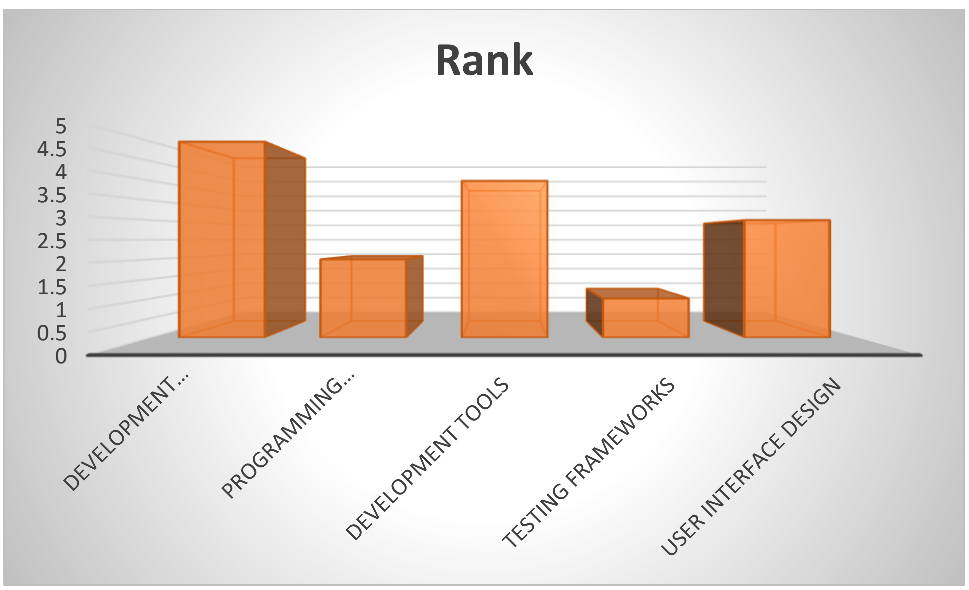


FIGURE 5. Shown the Rank

Figure 5 presents the rankings derived from the Grey Relational Grade (GRG) method, reflecting the relative performance of different software development aspects. Each facet is assigned a rank based on its effectiveness, with lower ranks indicating better performance. For instance, testing frameworks achieved the top rank, signifying its significant impact on overall development quality. These rankings offer actionable insights for stakeholders to prioritize areas for improvement or investment, thereby optimizing development processes and enhancing project outcomes. Through understanding these rankings, teams can focus efforts on aspects that contribute most to success, fostering efficiency and innovation in software development endeavors.

4. CONCLUSION

The software development life cycle (SDLC) encompasses various interrelated aspects, including design, requirements, project management, and process execution. This study employed the Grey Relational Analysis (GRA) method to evaluate the interoperability and relative importance of these components in software development endeavors. The analysis revealed that effective interoperability is crucial for streamlining workflows, mitigating compatibility issues, and delivering robust, integrated solutions that meet user needs. By fostering seamless collaboration among design, requirements, methodologies, languages, tools, frameworks, and user interface design, software teams can optimize development processes and enhance overall efficiency and effectiveness. The normalized data highlighted the varying degrees of influence each component exerts on different phases of the SDLC. For instance, user interface design holds significant weight in the design phase but less in project management, while testing frameworks are critical across requirements and project phases. Understanding and leveraging these interdependencies is essential for optimizing processes and enhancing collaboration. The deviation sequence, obtained through the GRA method, unveiled the relative importance or effectiveness of different elements within the development spectrum. Higher deviation values indicated areas for potential improvement or optimization, providing insights for informed decision-making. Furthermore, the Grey Relation Coefficient quantified the correlations between various facets of software development. By interpreting these coefficients, stakeholders can prioritize areas for improvement or focus attention on elements with the most substantial influence, optimizing the overall development process and enhancing project outcomes. The final Grey Relational Grade (GRG) Rank revealed the relative performance and effectiveness of each aspect. Testing frameworks secured the top rank, suggesting its significant impact on overall development quality, followed by programming languages and user interface design. These rankings offer valuable insights for prioritizing efforts and allocating resources, ultimately enhancing development processes and achieving better project outcomes. The study's findings underscore the importance of a holistic and integrated approach to software development. By recognizing the interdependencies among design, requirements, methodologies, tools, and frameworks, and leveraging the insights from the GRA method, software teams can make informed decisions, streamline workflows, and deliver high-quality products that meet user expectations and industry standards efficiently. Moreover, the GRA method's ability to handle uncertain and ambiguous data makes it a valuable tool for software development, where complex and dynamic environments are common. By incorporating this method, organizations can effectively navigate the challenges of software engineering and stay ahead in an ever-evolving technological landscape. This study demonstrates the efficacy of the GRA method in evaluating the interoperability and relative importance of various aspects of software development. By embracing a holistic approach and leveraging the insights derived from this analysis, software teams can foster collaboration, optimize processes, and deliver innovative, user-centric solutions that drive success in an increasingly competitive digital landscape.

REFERENCES

[1]. Ruparelia, Nayan B. "Software development lifecycle models." *ACM SIGSOFT Software Engineering Notes* 35, no. 3 (2010): 8-13.

[2]. Robillard, Pierre N. "The role of knowledge in software development." *Communications of the ACM* 42, no. 1 (1999): 87-92.

[3]. Carver, Jeffrey C., Richard P. Kendall, Susan E. Squires, and Douglass E. Post. "Software development environments for scientific and engineering software: A series of case studies." In *29th International Conference on Software Engineering (ICSE'07)*, pp. 550-559. Ieee, 2007.

[4]. Lindvall, Mikael, Dirk Muthig, Aldo Dagnino, Christina Wallin, Michael Stupperich, David Kiefer, John May, and Tuomo Kahkonen. "Agile software development in large organizations." *Computer* 37, no. 12 (2004): 26-34.

[5]. Boehm, Barry, Chris Abts, and Sunita Chulani. "Software development cost estimation approaches—A survey." *Annals of software engineering* 10, no. 1 (2000): 177-205.

[6]. Poo-Caamaño, Germán, Eric Knauss, Leif Singer, and Daniel M. German. "Herding cats in a FOSS ecosystem: a tale of communication and coordination for release management." *Journal of Internet Services and Applications* 8 (2017): 1-24.

[7]. Sengupta, Bikram, Satish Chandra, and Vibha Sinha. "A research agenda for distributed software development." In *Proceedings of the 28th international conference on Software engineering*, pp. 731-740. 2006.

[8]. Shukla, Yogeshwer, Anjali Yadav, and Annu Arora. "Carcinogenic and cocarcinogenic potential of cypermethrin on mouse skin." *Cancer Letters* 182, no. 1 (2002): 33-41.
[9]. Prikladnicki, Rafael, Jorge Luis Nicolas Audy, and Roberto Evaristo. "Global software development in practice lessons learned." *Software Process: Improvement and Practice* 8, no. 4 (2003): 267-281.
[10]. Dingsøyr, Torgeir, Tore Dybå, and Nils Brede Moe, eds. "Agile software development: current research and future directions." (2010).
[11]. Moniruzzaman, A. B. M., and Dr Syed Akhter Hossain. "Comparative Study on Agile software development methodologies." *arXiv preprint arXiv:1307.3356* (2013).
[12]. Khan, Arif Ali, Mohammad Shameem, Rakesh Ranjan Kumar, Shahid Hussain, and Xuefeng Yan. "Fuzzy AHP based prioritization and taxonomy of software process improvement success factors in global software development." *Applied Soft Computing* 83 (2019): 105648.
[13]. Von Krogh, Georg, and Eric Von Hippel. "Special issue on open source software development." *Research Policy* 32, no. 7 (2003): 1149-1157.
[14]. Choudhary, Vidyanand. "Software as a service: Implications for investment in software development." In *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, pp. 209a-209a. IEEE, 2007.
[15]. Ashraf, Shahzaib, Saleem Abdullah, Muhammad Qiyas, and Aziz Khan. "The application of gra method base on Choquet integral using spherical fuzzy information in decision making problems." *Journal of New Theory* 28 (2019): 84-97.
[16]. Kumar, B. P., and A. K. Birru. "Tribological behavior of aluminium metal matrix composite with addition of bamboo leaf ash by GRA-Taguchi method." *Tribology in Industry* 40, no. 2 (2018): 311.
[17]. Alvandi, Mohsen, Majid Elahi, Mostafa Memarzade, and Atefeh Hesaraki. "Developing a new MADM method (SIGRA) by integrating SIR and GRA methods." *AJSR. American Journal of Scientific Research* 36 (2011): 21-35.
[18]. Markabi, Mohsen Sayyah, and Mohammad Sabbagh. "A hybrid method of GRA and DEA for evaluating and selecting efficient suppliers plus a novel ranking method for grey numbers." *Journal of Industrial Engineering and Management* 7, no. 5 (2014): 1197-1221.
[19]. Jayakrishna, Kandasamy, Sekar Vinodh, Vijayaselvan Sakthi Sanghvi, and Chinadurai Deepika. "Application of GRA for sustainable material selection and evaluation using LCA." *Journal of The Institution of Engineers (India): Series C* 97 (2016): 309-322.
[20]. Hu, Diantao, Cong Zhang, Wenqi Cao, Xintao Lv, and Songwu Xie. "Grain yield predict based on GRA-AdaBoost-SVR model." *Journal on Big Data* 3, no. 2 (2021): 65.
[21]. Mufarrih, A., H. Istiqlaliyah, and M. M. Ilha. "Optimization of Roundness, MRR and Surface Roughness on Turning Process using Taguchi-GRA." In *Journal of Physics: Conference Series*, vol. 1179, no. 1, p. 012099. IOP Publishing, 2019.
[22]. Sharma, Khushboo, Gaurav Kumar, and Mukesh Kumar. "Application of taguchi method coupled with GRA for optimization of drilling process parameters." In *IOP Conference Series: Materials Science and Engineering*, vol. 1149, no. 1, p. 012032. IOP Publishing, 2021.
[23]. Pagar, N. D., and S. H. Gawande. "Multi-response design optimisation of convolution stresses of metal bellows using integrated PCA-GRA approach." *Australian Journal of Mechanical Engineering* 20, no. 2 (2022): 582-602.
[24]. Kolhapure, Rakesh, Manjunath Rawal, Suyog Rayjadhav, Bahubali Kabnure, Girish Mekalke, Rajkumar Patil, and Vishal Balwan. "Shape optimization of washer load cell using GRA method." *Materials Today: Proceedings* 27 (2020): 238-243.
[25]. Dahooie, J., A. Vanaki, N. Mohammadi, and H. Firoozfar. "Selection of optimal variant route based on dynamic fuzzy GRA." *Decision Science Letters* 7, no. 4 (2018): 523-534.
[26]. Vallejo Seco, Guillermo, Jaime Arnau Gras, and Manuel Ato García. "Comparative robustness of recent methods for analyzing multivariate repeated measures designs." *Educational and Psychological Measurement* 67, no. 3 (2007): 410-432.
[27]. Wu, Tong, Denis Arrivault, Marc Duruflé, Alexandre Gras, Felix Binkowski, Sven Burger, Wei Yan, and Philippe Lalanne. "Efficient hybrid method for the modal analysis of optical microcavities and nanoresonators." *JOSA A* 38, no. 8 (2021): 1224-1231.
[28]. Teli, Tawseef Ahmed, and Faheem Syeed Masoodi. "Iot implementation in India: A complex decision process using gra/ahp." In *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 773-778. IEEE, 2023.
[29]. Rehman, Erum, Muhammad Ikram, Ma Tie Feng, and Shazia Rehman. "Sectoral-based CO 2 emissions of Pakistan: a novel Grey Relation Analysis (GRA) approach." *Environmental Science and Pollution Research* 27 (2020): 29118-29129.

