IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 29 May 2025, accepted 8 July 2025, date of publication 16 July 2025, date of current version 19 August 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3589942

RESEARCH ARTICLE

RECTV: Enhancing Agile Decision-Making in Startups Through Value-Driven Requirements Prioritization

FRÉDÉRIC PATTYN [1], (Member, IEEE), RAFAQAT KAZMI [2], PETER GOETZ [3], AFNAN ALDHAHRI [4], AND AQSA [5]

[1]Business Informatics and Operations Management, Ghent University, 9000 Ghent, Belgium
[2]Department of Software Engineering, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan
[3]New Data Vision Ltd., 2000 Antwerp, Belgium
[4]Department of Software Engineering, College of Computing, Umm Al-Qura University, Makkah 24382, Saudi Arabia
[5]Department of Computer Science, Wuhan University, Wuhan 430079, China

Corresponding author: Rafaqat Kazmi (rafaqat.kazmi@iub.edu.pk)

This work was supported by Umm Al-Qura University, Saudi Arabia, under Grant 25UQU4310136GSSR05.

ABSTRACT Effective requirements prioritization is a success factor for software development. Models of prioritization that are traditionally used like MoSCoW, RICE, AHP and the Kano Model are deficient in responding to what is essential for startups in terms of speed and delivery of value, efficiency in cost, and mitigation of risk. In this paper, we introduce RECTV, a new startup requirements prioritization framework. Essential decision making components including Cost, Risk, Technical Feasibility, Confidence Level (C), and Time to Value (TtV) are integrated as key pieces of the RECTV tool to help startups prioritize based on informed, data based decision making. In contrast to traditional models, RECTV validates the outcomes of the features while delivering business value in a way where startups can adapt to market feedback and uncertainty rapidly. This work positions RECTV as a practical alternative to traditional models, offering startups a structured and responsive approach to prioritize features under real-world constraints. By conducting a comparative analysis with other models, the framework is evaluated and proved to be superior at tackling the startup glitches such as low funds, unspecified features, and the necessity of fast market adjustment. For startups, RECTV helps them to make better use of resources and develop features that avoid unnecessary risks.

INDEX TERMS Requirements prioritization, startups, risk assessment, time to value, confidence level, RECTV framework, software product management.

I. INTRODUCTION

Software requirements engineering plays a critical role in the software development life cycle (SDLC). Requirements engineering aims to turn the ambiguous desires of future software users into concise and formal specifications. However, it is not solely this that determines its significance in the software lifecycle. Requirements engineering cuts software errors early in the software development process. Gathering requirements is a challenging process. It involves several rounds of meetings between the software user and the software engineer. Moreover, prioritizing the most crucial requirements among multiple needs can be challenging [1]. For a business to achieve success, it is important to select the main features and functionalities to start with, since this helps reduce time-to-market and boost ROI. The standard Requirement Prioritization (RP) methods include MoSCoW, RICE, AHP and Kano which make the process of decision-making easier [2]. They do not help startups because they fail to account for how the market can change and for not having much money available while developing things quickly [3]. Since they depend on feedback from customers, startups must

The associate editor coordinating the review of this manuscript and approving it for publication was Francisco J. Garcia-Penalvo.

keep improving their plan to ensure financial security and manage risks.

The models used to prioritize business activities are not always effective for startups. RICE and MoSCoW are not suitable for early-stage startups that do not have much money because they are not designed to consider costs and possible risks [4]. Startup companies cannot judge whether or not to invest in features, because these models do not show their confidence levels yet. The main problem is that TtV is not used to help guide decisions [5]. Due to the rapid pace of startups, they must quickly give users something useful to help them succeed. Instead of devoting resources to long and risky projects, the current approach should ensure that features are prioritized until they give business value to the company [6]. Most available models are inflexible and cannot answer to the needs of startups which blocks agile development. It should bring together effectiveness, safety and the timely creation of value by linking them in one framework.

The main goal of this research is to create RECTV which represents a new requirements prioritization framework designed for startup organizations. The framework provides startups with a data-driven method to manage cost evaluation together with risk evaluation and technical viability assessment for effective strategic decision-making [1]. Two essential decision-making elements which RECTV models introduce consist of Confidence Level (C) and Time to Value (TtV). The Confidence Level (C) measurement system demonstrates the reliability of predicted feature outcomes therefore companies give priority to proven features rather than theoretical ones. Startups should focus on features with quick measurable business value returns through the Time to Value (TtV) assessment. Through these two factors RECTV helps startups make better resource management decisions while operating in uncertain business environments.

This study presents two significant outcomes. The study presents RECTV as a startup-focused requirements prioritization model which integrates cost estimation and risk assessment with feasibility testing and time-to-value measurements and confidence level evaluations. The structured scoring system in this model provides startups with better decision-making abilities that ensure prioritization accuracy when markets rapidly change. This research evaluates RECTV against standard prioritization frameworks by testing it against MoSCoW, RICE, AHP and Kano. The evaluation shows that RECTV provides startup environments superior capabilities to handle risk management alongside cost reduction and fast feature validation processes than traditional methods. The research solves conventional prioritization framework weaknesses through an innovative methodology that matches startup environments.

Traditional prioritization models (e.g. MoSCoW, Rice, AHP and Kano) are highly used, but limited in dynamic or resource-constrained environments. Such practices are usually rigid, and are based on subjective criteria, and do not include important aspects like technical viability, delivery schedule, and team confidence on realizing the outcome. Such models can lead to erroneous priorities especially in situations of startups or agile, where frequent pivots, changing stakeholder input, and unpredictable outcomes are the norm. As an example, a feature with a high value might not be viable in the current team capacity or it might be associated with unknown risks of integration. In a bid to fill these gaps, this paper presents RECTV, a configurable multi-factor prioritization model that combines Value, Confidence, Effort, Risk, Technical Feasibility and Time to Value (TtV) into a common scoring scale. Time to Value is the rate at which a feature is likely to create observable value upon release and Confidence-based prioritization is a method of assigning scores according to how sure the team is that a stated value will come to pass- whether through data, experience or history. Through its consideration of these subtle (aspects), RECTV provides a more concrete and flexible method to fit in the real life product development situations.

The remaining part of this paper consists of multiple sections. This section reviews available prioritization models alongside their identified shortcomings. Section III explains the methods that led to RECTV framework creation. The RECTV model's design and components are explained in Section IV along with an explanation of its inclusion of startup-specific decision-making aspects. Section V delivers a comparative assessment to show how RECTV operates compared to traditional methods. The final section analyzes important outcomes from this research along with its constraints as well as provides conclusions for the study and proposes new research paths.

II. RELATED WORK

The most valuable and feasible software features require implementation as a primary step during requirements prioritization processes. Multiple frameworks exist to assist organizations in their requirement selection process through systematic ranking methods. Among popular priority setting models are MoSCoW and RICE alongside AHP and the Kano Model. The structured prioritization approaches present problems when used with startups because they face financial restrictions during rapid market evolution alongside immediate requirement validation needs. This part evaluates current prioritization frameworks to recognize their shortcomings with startup conditions before presenting the RECTV framework as an effective solution which combines cost control with risk assessments and feasibility studies and rapid value delivery.

The MoSCoW method represents a common method for requirement classification which separates needs into absolute requirements and optional features [1]. Qualitative analysis works best in agile projects that require flexible and easy-to-apply priority decisions. Because MoSCoW does not include numerical weightings it becomes difficult to perform objective feature comparisons [2]. Numerous research documents describe MoSCoW's weaknesses as a structured decision-making framework while pointing out its limitations when making decisions in quantitative-scoring environments.

The extensive use of MoSCoW as a qualitative prioritization method requires additional assessment models that merge cost-effectiveness evaluations with data-based strategic approaches [3]. The subjectivity of MoSCoW classification leads to inconsistent requirement prioritization since stakeholders determine priorities through their personal and departmental preferences instead of business value alignment according to study [4]. Startups requiring financially sustainable decisions lack essential cost and risk evaluation tools within the MoSCoW method since it omits confidence-based and cost assessments. The absence of quantitative weighting in MoSCoW hinders systematic feature comparison according to [5] which reduces efficient resource management capabilities. Startups implementing MoSCoW could choose features based on personal judgments of urgency instead of true business value. The MoSCoW method does not address financial boundaries and risk assessment needs of startups as [6] states in their argument about sustainable startup development.

The RICE model represents a widespread prioritization approach that determines ranking through evaluation of Reach, Impact, Confidence and Effort factors [7]. RICE adds a numerical scoring approach to prioritization that provides more structure over MoSCoW methods. The main weakness of RICE is its failure to consider expense and risk elements which are crucial factors for resource-constrained startup organizations. Many studies have focused on the shortcomings of the RICE model in the case of startups struggling with finances and having to manage significant risks [8]. Startups end up wasting resources and spending more money because there are not enough tools for assessing risks. RICE is more objective than MoSCoW in assessing priorities, but it requires additional techniques to take both risk and financial aspects into account. Researchers in [13] argue that RICE offers a way to rank features, but it does not have the right tools for risk assessment, so startups often find it difficult to sustain their efforts in innovation. While Confidence Level (C) is a factor in the model, a method is not provided to prioritize established features over unproven ones. The findings [14] reveal that C in the RICE model is not able to tell confirmed features apart from unconfirmed ones, so startups might choose risky but significant projects. Companies can still decide to focus on high-impact requirements, even if it isn't clear if they can be achieved, leading to useless spending [15].

With AHP, organizations can assign weights to different criteria using a comparison process that pairs each criterion [16]. Using this approach, businesses can learn about the effects of different design decisions on their products. It has been documented in numerous studies focused on startups that AHP is limited when used in rapidly changing business environments [9]. The way AHP organizes priorities is not effective in startups since it has two major issues: the process needs to be very precise and the calculations are very difficult. Working with AHP takes time and involves many steps, causing difficulties in fast software development [17]. Since AHP is both complicated and requires a lot of time to process, it is not practical for startups who need to act fast. AHP is inflexible mainly because its structure is fixed and researchers have found that this is most suitable for organizations with definite aims, unlike startups that must adjust to the market [10]. Because AHP has a set structure, it works better for companies that have stable products, rather than startups in constantly changing environments. While the pairwise comparison approach is effective for creating weighted results, it is hard to use when considering numerous features, resulting in inefficiencies [19].

The Kano Model method groups product features into satisfaction classifications. The model organizes requirements into essential Basic Needs followed by Performance Features which enhance user experience and the additional Excitement Features that create delightful surprises [11]. Some studies demonstrate Kano Model limitations when startups need to evaluate financial and operational feasibility. The Kano Model successfully divides features according to customer satisfaction but lacks structured cost-benefit assessment therefore making it hard for startups to achieve sustainable business alongside desirable features according to [12]. The Kano Model provides businesses with tools to distinguish between qualities that customers need (Basic Needs) and those that enhance satisfaction (Performance and Excitement features) [16]. Research by [13] demonstrates that the model makes the flawed assumption that customer preferences stay constant over time but such stability cannot be expected in fast-changing markets thus creating development gaps with real user needs [14]. The Kano Model serves well in determining customer expectations yet its insufficient assessment of costs and feasibility bars its independent use in startup decision processes. Kano lacks the ability to assess feasibility or cost alongside risk factors which limits its usefulness for startups focusing on business viability above customer preference. The qualitative focus of Kano's approach according to [15] creates problems for startups because they face difficulties determining which features maximize their return on investment when working with limited resources.

Over the last few years, there have been some new frameworks of prioritization introduced, which build on the existing models adding data analytics, machine learning, and multi-criteria decision-making. A framework [16] uses machine learning models to extrapolate past backlog data to determine the feature impact, which however needs large amounts of data and might not be transparent in terms of the decision rationale. Fuzzy AHP-TOPSIS hybrids [17] have been applied to manage imprecision and imprecise stakeholder inputs, although these are computationally complicated and tend to require specialized expertise to implement. RBF-PM [18] has the ability to dynamically incorporate business rules and constraints but it is not as flexible as changing priorities that rapid and agile environments are likely to have. Also, the models that consider technical debt or the risk of user churn are narrower in scope and more difficult to extrapolate. Such more recent techniques are innovative; however, they are often niche problem space, need special tooling, or have poor flexibility. On the other hand, RECTV offers a scalable, explainable and flexible scoring system that maintains a balance between value, confidence, risk, feasibility and speed of delivery. It is therefore easier to be used by teams within a fast-paced or resource-limited environment, yet it is also possible to combine it with data sources and customize the weighting according to the context.

The current models which use structured prioritization practices do not resolve the special difficulties which startup companies face [19]. Current startup frameworks do not include necessary features for cost evaluation and risk assessment. Startups function within tight financial restrictions so their developed features must fulfill both budget requirements and business expansion sustainability [20]. The traditional decision-making methods MoSCoW and RICE omit financial assessment and risk evaluation of building new features thus causing startups to unnecessarily invest in unprofitable features [21].

The analysis suffers from a major drawback due to its lack of structured confidence-based feature prioritization methodology. Startup projects operate under significant uncertainty because they use validated requirements from customer data and market research yet need to speculate about other requirements [22]. The current priority assessment tools fail to differentiate between features with strong validation evidence and those with limited evidence which leads startups to invest in unverified characteristics. Startups require a prioritization model which uses Confidence Level (C) as an essential decision input to move resources toward features displaying actual success potential based on factual data [23].

Traditional methods of prioritization fail to deliver the required adaptability needed by startups operating in dynamic business environments [24]. Startups need to redirect their operations due to audience responses and monetary sources and market dynamics that change regularly. The inflexible nature of AHP and Kano methods prevents them from accommodating changes in requirements which makes startup operations with their agile development process incompatible with these approaches [25]. A startup-friendly prioritization model must adapt swiftly to new information and feature this dynamic reprioritization ability while guaranteeing the development process stays committed to business objectives [26]. Although such models as MoSCoW, RICE, AHP, and Kano have been effective, the RECTV model solves some of the main weaknesses of these models in an agile and innovation-driven environment. MoSCoW is not granular in scoring and it is very subjective. RICE does not consider such aspects as feasibility, risk, and team capacity although it is a quantitative scoring system. AHP offers the robust pairwise comparisons that are hard to scale due to frequent change. Kano measures customer satisfaction though it does not consider effort or technical feasibility. Conversely, RECTV integrates multi-dimensional scoring (value, risk, confidence, feasibility, TtV) in a flexible, customizable structure that is more appropriate to high velocity product situations where a variety of trade-offs will need to be balanced in real time.

The common assumptions of traditional prioritization models are that there are stable planning cycles and that there are well-established resource baselines; both of these are usually not the case in the startup or agile environment. As an example, startups are characterized by frequent pivoting, changing team capacity, as well as unpredictable value estimation. Such models as AHP or RICE can be too rigid or time-consuming though they are rather methodical. Lightweight MoSCoW and Kano are high dependent on subjective evaluation, and they do not support any technical limitations. RECTV deals with these issues by incorporating risk, feasibility, confidence and user-configurable weighting into the scoring mechanism, and by improving responsiveness and realism in fast changing environments. Startups require an adaptable evaluation system that helps them manage costs with risks and feasibility alongside clear value assessments despite fast-changing market conditions. The RECTV framework addresses these challenges through its structured numerical system which integrates financial limitations with risk management and confidence-based decisions [27]. The RECTV approach implements Confidence Level (C) as its foundation because it makes startups focus on proven features rather than theoretical ones. RECTV implements Time to Value (TtV) as a measurement method to determine how rapidly features create business value and thus startups should prioritize features that produce immediate measurable benefits instead of potential but uncertain long-term outcomes [28]. VECSTR solves these important startup barriers to create an operational framework that optimizes business prioritization [29]. The framework creates a scalable structured and adaptable solution which helps startups make decisions through cost-risk-technical feasibility assessments during market adjustments. The subsequent section describes the conceptual development of the RECTV framework through an explanation of its scoring structure and priority evaluation components for startup applications [30].

III. PROPOSED MODEL

The development of the RECTV framework for requirements prioritization is based on a conceptual research approach that includes the theoretical model construction, the expert validation of the theoretical model and the comparative analysis. The three main stages of the methodology are: development based on literature review, theoretical model construction, and validation based on Delphi. This was done by designing these stages to make sure RECTV is theoretically sound, practically applicable, and better than other prioritization models.

The first part of the study entailed an in depth literature review of existing prioritization methods, namely MoSCoW, RICE, AHP as well as the Kano Model. This review also helped identify major shortcomings in the current models,

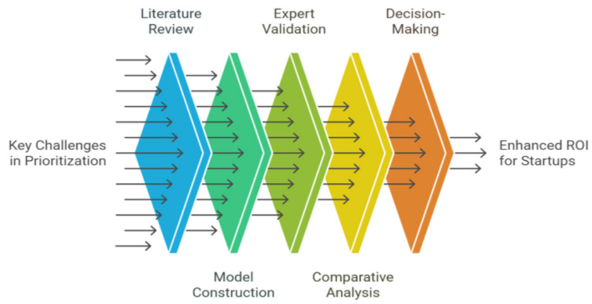

FIGURE 1. **RECTV framework for startup prioritization.**

including no structured cost and risk assessment, no use of confidence based prioritization, and no ease of adaptation to agile and startup environment. Based on the findings from the literature review, the RECTV framework was designed with the V, E, C, T, and R as core prioritization factors.

The second stage is the theoretical model construction, for which a structured scoring mechanism is developed to make sure the prioritization decisions are quantifiable, transparent and data driven. RECTV does not rely on subjective categorization or pairwise comparisons like traditional models, but rather employs a formula based approach to balance business impact, feasibility, and risk mitigation. Mathematical foundation of the model allows the organization to rank features consistently and objectively without the bias and inconsistencies.

The third stage involved validating the RECTV framework through the use of the Delphi method, a structured expert consensus method applied in model evaluation and refinement. The Delphi process was conducted with a panel of 12 domain experts, product managers, software engineers, business strategists, UX researchers, etc. The experts iterated three times through rounds of evaluation and feedback to refine the definitions of RECTV's factors, adjust their relative weight, and their ability to be adapted to different industries. By the final round, all experts agreed that RECTV solves the most important prioritization challenges (100%) and that RECTV is robust and relevant (100%) to the industry. After the Delphi validation, RECTV was compared with MoSCoW, RICE, AHP, and the Kano model using 15 evaluation criteria such as quantifiable scoring, risk consideration, confidence based prioritization and time to value assessment. The results of this showed that RECTV is more transparent, adaptive, and structured than traditional models for feature selection and product development, and therefore is a superior framework. In order to be theoretically grounded, empirically validated and industry ready, this methodology guarantees that RECTV is a scalable, structured, and risk aware prioritization model for organizations.

A. LITERATURE REVIEW-BASED DEVELOPMENT

The framework's development strategy includes an extensive review of existing requirements models to understand their strengths and limitations when used in startup environments. The efficient organization of resources to business-critical features stands as a fundamental practice of software development and product management. Different models like MoSCoW and RICE along with AHP and the Kano Model are extensively used throughout industries. Various traditional methods face substantial limitations when implemented for startup decision-making due to startup conditions which include economic restrictions along with quick strategic adjustments and considerable operational unpredictability. Studies of existing business feature priority frameworks show their models use subjective importance assessments while lacking standard measurable criteria and neglect cost and risk together with validation confidence as essential decision elements. New businesses need flexible decision-making systems with structured scoring capabilities that maintain their ability to adapt to changing business requirements. The literature review serves as the basis for the RECTV framework by developing a data-based prioritization system that serves startup requirements.

1) MOSCOW METHOD

A very popular prioritization approach in agile project management and software development is the MoSCoW method (Must-have, Should-have, Could-have, Won't-have) [31]. Then, it determines a feature is one of four categories. These are the features that are required in the system to work.

- Important features that add significant value but are not mandatory are termed as should-have.
- Non-essential features which gives more benefits but has low priority could have.
- Features that will not be included in the current scope, but can be revisited later.

However, MoSCoW is intuitive and easy to implement, but it has a number of major shortcomings. Secondly, MoSCoW does not have a structured numerical score method which leads to prioritization of requirements being very subjective. Features are broken down based on stakeholder input; therefore there is no way to statistically differentiate two features in the same category [32]. It can result in inconsistent decision making, especially when there is disagreement among different stakeholders with their priorities. Second, MoSCoW does not take into consideration financial feasibility and risk assessment. The technique presumes that all must be features should be implemented, but does not supply a way to verify that a corporation features the capital, technical assets, and time to build these features. Overcommitment, unrealistic timelines, and poor resource allocation are effects of this. Lastly, MoSCoW is not flexible for dynamic environments. Once a feature has been categorized, it becomes hard to reevaluate the priority on that feature as the business changes. The lack of flexibility makes MoSCoW unfeasible in a high paced, continuously evolving product development environment such as startups or agile teams that update their product roadmap by feedback from the market.

Therefore, though MoSCoW continues to be a useful tool for simple prioritization, it is not suitable for data driven decision making and risk aware prioritization and we need to explore more structured frameworks.

2) RICE METHOD

The RICE (Reach, Impact, Confidence, Effort) method is one of the widely used scoring based prioritization framework in product management and software development [33]. It assesses requirements on the basis of four key factors.

- R = Reach (estimated number of users or customers who will benefit from the feature).
- Impact (I): This indicates the business value or the customer satisfaction generated by the feature.
- Certainty of the estimates offered for Reach and Impact (C)
- Effort (E): The resources, time, and complexity required to implement the feature.

RICE scores are calculated using the formula:

$$\text{RICEScore} = \frac{(\text{Reach} * \text{Impact} * \text{Certainty})}{\text{Efforts}} \tag{1}$$

This scoring system enables product teams to pick features that give maximum value for the least amount of work. Although RICE has many advantages, however it has some limitations. A disadvantage of RICE is that it does not take any risk factors into account specifically. The Confidence metric does not assess dependencies, technical challenges or business risks, but indirectly accounts for uncertainty. This can result in high risk features being pushed above lower risk, more stable ones [34]. Also, RICE does not include a structured cost assessment. Lessons learned from the Effort factor is that it takes into account the development effort, but does not include the financial feasibility. Even if it goes beyond the budget constraints or needs resources still not available, the feature with high impact and high effort will be prioritized.

The Confidence metric is not well-defined and has a subjective input in another limitation. Confidence is usually expressed in percentage terms, with the estimates accuracy as the base, and some teams may subconsciously play with the score to justify the priority choices. RICE is an improvement over MoSCoW as it introduces a scoring system, but when we talk about high uncertainty or financial constraints, RICE is not good enough because it does not account for risk or cost.

3) ANALYTIC HIERARCHY PROCESS (AHP)

The Analytic Hierarchy process (AHP) is a decision making framework where weights are assigned to prioritization factors through pairwise comparison [35]. Unlike RICE and MoSCoW, AHP does not rely on a predefined scoring system but allows decision makers to judge requirements on their relative importance. AHP follows a structured approach:

- Pairwise Comparison: Features are based on one to one comparison and prioritization is done with respect to some criteria (say business value vs. development effort).
- Each criterion is given a weight according to its importance.
- Score Calculation: Rank features by weighted aggregation of the scores.

Although AHP is a mathematically rigorous means for prioritization, it suffers several major limitations. AHP is first highly complex and time consuming. It involves decision makers comparing each feature to each other feature, making it not practical for agile teams working on rapidly changing requirements. Second, AHP lacks adaptability [36]. After prioritization weights are assigned, it is difficult to change these weights without repeating the whole process, so we are stuck and not easily able to adapt to changing business conditions. Finally, AHP does not take into account cost and risk assessment. It is a structured decision making framework, but it does not consider financial feasibility, uncertainty, dependencies, all of which are critical to prioritization. These challenges make AHP too rigid for the modern, agile driven environments, so a more flexible, risk aware model like RECTV is needed.

4) KANO MODEL

The Kano Model is a prioritization based on customer satisfaction method which categorizes features in relation to their impact on user perception and experience. It categorizes the requirements into three categories: Basic Needs (Basic features that customers expect and need the product to have), Performance Needs (Features that improve customer satisfaction when improved or optimized) and Excitement Needs (innovative or unexpected features that provide competitive advantage and significantly increase user delight) [37]. The Kano Model is useful to know what customers want and what features add value versus what features are necessary (in other words, what customers will pay for, versus what they expect for free), but is not useful for structured prioritization.

The Kano Model is one of the major disadvantages in terms of not having a structured ranking methodology to prioritize the features. Unlike the scoring based model such as RICE and AHP; Kano does not provide a numeric assessment to compare the features. It rather uses qualitative classification, which may result in ambiguities of decision making especially when resources are limited [38]. Additionally, the model only takes into account customer satisfaction and disregards technical feasibility, financial impact, and business risk – all of which matter when deciding if a feature should be prioritized. Furthermore, the Kano Model does not give a way to estimate development effort, cost constraints, or risk factors that might be associated with implementing features. Therefore, although it is a good tool for grouping and understanding user expectations, it misses the evaluation criteria for prioritization in agile product development and business decisions. As a result, the Kano Model is best suited as an additional framework rather than as a stand alone prioritization method that assists teams in recognizing features that improve user satisfaction and using other structured approaches for resource allocation and feasibility assessment.

To address these limitations, the RECTV framework integrates principles from multiple decision-making theories, ensuring a comprehensive, data-driven approach to prioritization. Each factor in RECTV is grounded in established theoretical foundations, making it a scientifically valid model for prioritization.

TABLE 1. Justification for proposed model.

RECTV Factor	Theoretical Foundation	Justification
Risk (R)	Risk Management Frameworks [39]	Allows teams to assess uncertainty and dependencies before committing resources.
Effort (E)	Theory of Constraints [40]	Helps minimize development bottlenecks and optimize resource allocation.
Confidence (C)	Lean Startup Methodology [41]	Prioritizes validated features over speculative ideas to reduce risk.
Technical Feasibility (T)	Technology Adoption Model [42]	Ensures that selected features are technically feasible.
Value (V)	Cost-Benefit Analysis [43]	Ensures prioritization aligns with business impact and ROI.

These theoretical foundations provide scientific credibility to the RECTV framework, ensuring that each prioritization factor is backed by established decision-making principles.

B. THEORETICAL MODEL CONSTRUCTION

The RECTV framework is developed using a structured, theoretically grounded approach such that it can be effective in feature prioritization and decision making. Based on the theoretical foundations of the RECTV factors and the identified traditional prioritization model's gaps, the next step was to build the theoretical model that consists of defining the core prioritization factors, the scoring mechanism and the weighting strategy. The approach taken in RECTV is a structured one that enables balancing of business impact, technical feasibility, risk management and resource constraints, making it scientifically valid and practically applicable.

The RECTV model, is made up of five key prioritization factors that include Value (V), Effort (E), Confidence (C), Technical Feasibility (T) and Risk (R). The selection of these factors was based on an extensive literature review and their alignment with the existing decision making theories so as to provide a multi-dimensional feature prioritization. To overcome the limitations mentioned in existing models, each factor was designed to be incorporated in a structured, quantitative approach to ranking features.

1. Risk (R) takes into account the uncertainties, dependencies and possible obstacles that might occur in the process of implementation. Features with high risk need to be scrutinized more before they are prioritized, so that resources are allocated in a smart way.
2. The Effort (E) factor represents the complexity of the development and the resources that would need to be invested in order to implement a feature. RECTV incorporates effort as a criterion into prioritization, thus preventing teams from over-committing on high effort but low ROI features.
3. The C (Confidence) factor is how certain or validated a feature is. A lot of the prioritization methods treat feature ideas equally, but in a real world product management there are features that are based on strong user research and market validation and others are just speculation. The Confidence factor prevents high risk and unvalidated ideas from rising to the top over initiatives supported by evidence.
4. Technical Feasibility (T) tests whether a feature is possible to build within the existing technological, infrastructure, and engineering limitations. This stops teams from focusing on high value features which would be infeasible to deliver in their current ecosystem.
5. Value (V) factor explains the business impact of feature, so that high priority points match with company objectives and customer wants. Higher value scores for a feature implies that this feature contributes more in terms of generation of revenue, competitive differentiation and user engagement.

Even though RECTV is independent in scoring each of its five prioritization parameters Value (V), Effort (E), Confidence (C), Technical Feasibility (T) and Risk (R), it takes practical interrelationships of these parameters into consideration specifically in startup scenarios. There is a significant dependency between Value (V) and Confidence (C). Features may seem very valuable in early-stage product development because of market opportunity or strategy vision. But where the features are not supported by user validation, stakeholder alignment, or historical precedent (i.e., low C), their true business effect is unknown. To respond to this, the formula used in RECTV clearly takes Confidence as a moderator of Value:

To turn these factors into a structured decision-making model, a scoring formula was created to give a quantifiable and repeatable mechanism to rank features. In the following, the proposed RECTV scoring formula is:

$$\text{RECTV Score} = (V \times C \times TtV) - (E \times R) + T \qquad (2)$$

This equation allows us to focus on the highest value and confidence features and takes effort, risk and feasibility into account. Value (V) and Confidence (C) are multiplied to make sure high value features that have strong validation to take high priority, and Effort (E) and Risk (R) are multiplied to punish complex and high risk features. The addition of Technical Feasibility (T) helps in adding more weight to features that are feasible, thereby improving the model that is applied in real world development environments. Furthermore, an additional weighting system is able to be implemented to adjust the importance of the factor according to organizational priorities. For instance, the startups interested in rapid experimentation may give more value to Confidence (C) and Time to Value (TtV), while the enterprises looking for stability may emphasize Risk (R) and Technical Feasibility (T). This makes the framework of RECTV scalable and adaptable to different business models and industries.

The inclusion of Time to Value (TtV) in the RECTV scoring formula is official because it considers how fast a feature can provide measurable results. It multiplies Value-Confidence term and is expressed as a normalized value ranging between 0.1 (slow impact) and 1.0 (fast or immediate impact). Delivery forecasts, feature lead times or past deployment velocity can be used to estimate TtV. This integration makes sure that the features which have a potential to provide early business or user value are preferred over those features which have long returns. In example, a lightweight UX feature that has $V = 0.6$, $D = 0.8$, and $PtV = 1.0$ will produce a value-confidence product of 0.48.

RECTV can incorporate TtV directly in the formula and be more responsive to time-based business requirements and agile decision-making, a process where speed of impact is as critical as the likely value. Featuring these elements, RECTV allows for a well-structured and adjustable way to set priorities for features that help deal with important issues in traditional models. The Delphi method was then applied by experts to make sure that the RECTV framework is valid and workable. This formulation makes it such that the speculative features, even those perceived to have high values are de-prioritized when they are not backed by confidence. E.g. a feature that has a high Value score (V = 0.9) but low Confidence (C = 0.2) would only have a weighted value impact of 0.18 and lower its overall prioritization score. This movement stops the teams in early stages of development to over-commit to unvalidated ideas and focuses on those features that are not only valuable but also justified by real-life evidence. This interdependency can add strength to validation-driven and risk-minded approach to prioritization, which explains why RECTV is particularly applicable in agile and startup environments.

1) WEIGHT ASSESSMENT FOR EACH FACTOR

The sub-components of each primary factor of Risk (R), Effort (E), Confidence (C), Technical Feasibility (T) and Value (V) in the RECTV framework are also considered. These sub components capture the complexity of decision making in startup environments through technical, operational and strategic perspectives. A weighted aggregation method is used to derive a unified score for each of the factors. This method gives each sub component a relative importance and thus the final factor score will reflect empirical evidence but in the context of priorities at the startup. Weighted scoring allows the RECTV model to be made more flexible, repeatable and adaptable across varying startup settings and product domains.

Not all subcomponents in a factor are equally important to decision making. For instance, effort (E) factor includes development effort factor that dominates other factors such as testing or integration complexity. If all sub-components were treated equally (i.e., simple average), the final factor score would not capture the true development burden. In order to address this imbalance, the RECTV framework uses a weighted prioritization to achieve more accurate prioritization. Stakeholders can use differentiated weights to bring attention to those dimensions that have the greatest impact based on domain knowledge, project constraints or startup maturity level. In addition, weighted aggregation facilitates better accuracy in estimation and reduces the inherent subjectivity involved in expert based scoring.

TABLE 2. Delphi-derived weight distribution for RECTV factors.

Factor	Sub-Component	Symbol	Weight
Risk (R)	Technical Risk	R_T	0.4
	Market Risk	R_M	0.3
	Resource Risk	R_R	0.3
Effort (E)	Development Effort	E_D	0.5
	Integration Complexity	E_I	0.3
	Testing Effort	E_T	0.2
Confidence (C)	User Validation	C_U	0.5
	Stakeholder Confidence	C_S	0.3
	Historical Similarity	C_H	0.2
Feasibility (T)	Stack Compatibility	T_T	0.4
	Codebase Readiness	T_C	0.3
	Team Capability	T_P	0.3
Value (V)	Revenue Potential	V_R	0.4
	Customer Impact	V_C	0.4
	Strategic Fit	V_S	0.2

The total risk score is a weighted sum of three sub components of technical risk, market risk and resource risk

$$R = R_T * 0.4 + R_M * 0.3 + R_R * 0.3 \tag{3}$$

Resource Risk is an element of total Risk (R) factor and it is computed by using three measurable parameters which are Developer Availability (DA), Skill Match (SM), and Integration Complexity (IC). DA is the proportion of necessary developers that can be employed to realize the feature (scaled between 0 to 1), SM is the degree to which the skills of the team match the technical requirements in the feature (0 = no match, 1 = perfect match), and IC is the anticipated complexity of incorporating the feature into the existing systems (0 = easy, 1 = highly complex). The following subcomponents are compounded with the weighted formula:

$$R_R = w_1 \times \text{(1-DA)} + w_2 \times (1 - SM) + w_3 \times IC \tag{4}$$

where the default weights are w 1=0.4, w 2=0.3 and w=0.3. It is a formula that has punished low resource availability and skill alignment and considered the difficulty of integration, which provides a viable and measurable method of measuring risks of resources in feature prioritization.

Effort Score is computed using:

$$E = E_D * 0.5 + E_I * 0.3 + E_T * 0.2 \tag{5}$$

Confidence Score is computed using:

$$C = C_U * 0.5 + C_S * 0.3 + C_H * 0.2 \tag{6}$$

Feasibility Score is computed using:

$$T = T_T * 0.4 + T_C * 0.3 + T_H * 0.3 \tag{7}$$

Value Score is computed using:

$$V = V_R * 0.4 + V_C * 0.4 + V_S * 0.2 \tag{8}$$

The empirically derived weight assignments used in the RECTV framework were developed through a Delphi study with 12 experts in five early stage and growth phase startups. The participants were product managers, software architects, quality assurance leads and also startup founders with domain expertise in agile product delivery. A structured Likert scale was used to independently rate the relative importance of sub-components within each RECTV factor, by each participant. The scores were added and normalized to produce a sum of 1.0 for the weights in each factor. The resulting weight distribution ultimately results from the concensus driven prioritization logic one typically observes from a startup environment which requires the balancing of development constraints, business impact and feasibility trade offs. The final weights are shown in Table 2.

The weights of RECTV subcomponents (e.g., Stack Compatibility = 0.4 under the Technical Feasibility) were calculated systematically in a Delphi process of two expert reviews. The rating of the importance of each subcomponent was conducted by 12 domain experts (product managers, technical leads and startup advisors) who individually rated each subcomponent on the scale of importance using their practical experience on various software projects. Their normalized responses were averaged to calculate their final weights, with a balanced display of real-world priorities.

To better understand the impact of these weights, two competing features are evaluated under Technical Feasibility (T) that comprises three subcomponents and they are Stack Compatibility (weight = 0.4), Codebase Readiness (0.3), and Team Capability (0.3). Suppose the following scores (0 1 scale):

Using the weighted sum:

- Feature A Feasibility = $(0.9 \times 0.4) + (0.8 \times 0.3) + (0.7 \times 0.3) = 0.81$
- Feature B Feasibility = $(0.5\times0.4) + (0.7\times0.3) + (0.8\times 0.3) = 0.65$

Despite Feature B having slightly better team capability, the stronger Stack Compatibility of Feature A, emphasized by the 0.4 weight, leads to a higher overall feasibility score. This prioritizes features that are less risky and more technically aligned, which is especially critical in high-dependency environments like Fintech or regulated industries.

TABLE 3. Example calculation of technical feasibility scores for two competing features using weighted subcomponents.

Subcomponent	Feature A	Feature B
Stack Compatibility	0.9	0.5
Codebase Readiness	0.8	0.7
Team Capability	0.7	0.8

C. EXPERT VALIDATION THROUGH THE DELPHI METHOD

The Delphi method was used to confirm the theoretical and practical aspects of the RECTV framework, as well as its usefulness in the industry. The Delphi technique involves repeatedly getting experts' opinions to combine their experience and enhance a model. This approach was used by the researchers as it has previously proven effective in looking at decision-making approaches and reducing human errors in the models. Experts in different specialties use the Delphi method to ensure both academic and real-world value in the development of RECTV.

The Delphi study consisted of three sequential rounds which incorporated twelve subject matter experts who specialized in product management together with software engineering and risk assessment alongside business strategy. The selection of experts focused on their expertise in agile product development together with decision analysis and feature prioritization experience. The feedback process in each round allowed continuous framework development to improve both effectiveness and industry compatibility of the RECTV framework.

1) ROUND 1: INITIAL EXPERT EVALUATION

In the first Delphi round, the initial expert feedback was collected on the relevance, completeness, and applicability of the RECTV prioritization factors. Detailed documentation for framework was provided to the experts with definitions on the five prioritization factors: Value (V), Effort (E), Confidence (C), Technical Feasibility (T), and Risk (R). An explanation was also provided to them about the proposed RECTV scoring formula, its weighting and a comparative analysis between RECTV and other prioritization models like MoSCoW, RICE, AHP, Kano. This round aimed to find out whether the dimensions of feature prioritization contained in the RECTV framework were sufficient, whether the scoring formula reflected the real life decision making in a proper way and if there were any critical factors missing.

The first round findings showed that 83% of experts agreed that the five dimensions of the RECTV factors grouped together sufficiently covered the dimensions for prioritization. Nevertheless, there were some experts who proposed to revise the weights of Confidence (C) and Risk (R) to match actual decision making, particularly in those industries

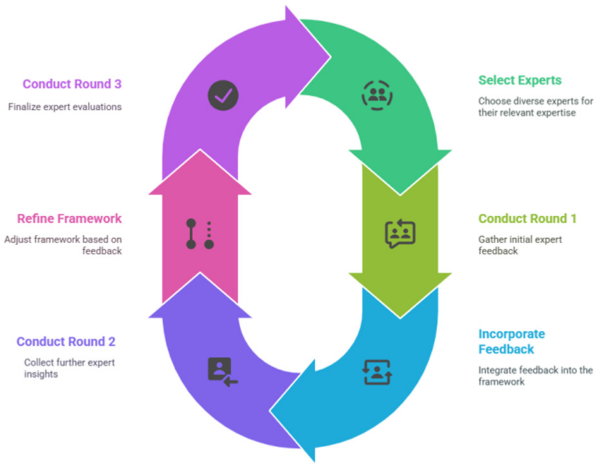

FIGURE 2. Delphi method validation cycle.

that rely heavily on uncertainty. Experts from some of those organizations, in fact, also suggested the addition of a sensitivity adjustment that would enable organizations to adjust factor weights according to their own tolerance to risk. With these insights in hand, we refined C weighting to distinguish between early stage and mature stage feature validation, introduced an optional mechanism to adjust weights with respect to organization's sensitivity to risk, and clarified the role of TtV within the model.

2) ROUND 2: REFINEMENT AND MODEL ADJUSTMENT

The refined RECTV model was presented in the second round to the expert panel for reevaluation. This phase assessed whether updated weighting and factor adjustments increased or decreased decision accuracy and practical applicability. A sample set of feature prioritization cases were given to experts to apply RECTV scoring to, and its performance was compared against their real world decision making challenges. Results from Round 2 revealed that 92% of experts believe that the adjusted version of the model is more in line with the needs of prioritization in different business settings. Fintech and healthcare industries sought a higher R value, while startup industries argued for more C and less TtV. Large enterprises, meanwhile, used a structured approach to evaluate V and T.

Another change was added to increase RECTV's flexibility: the weights of the factors can be adjusted according to what the industry requires. Now, each industry could use RECTV in the way that was best for them. The scoring process was made more accurate as extra adjustments were made to ensure decisions were clear and the prioritization works well for both agile and enterprise processes.

3) ROUND 3: FINAL MODEL VALIDATION AND CONSENSUS

The final round of the Delphi study aimed to confirm that the updated and final RECTV framework performs better than before and ensures all changes from the previous rounds were helpful. Overall, experts were asked to rate if the model was balanced in its approach to business impact, feasibility, risk and confidence.

Round 3 showed that all experts agreed that RECTV incorporates all the vital elements for structured prioritization. They concluded that the framework is versatile and it provides better results than other models used for prioritization. A formula was used during the Delphi validation process to determine the amount of agreement among experts regarding different RECTV framework elements. The scientists solved the problem by using this equation:

$$\text{Consenses Percentage} = \left(\frac{\text{No of experts who rated 4 or 5}}{\text{Total no of experts}}\right) *100 \tag{9}$$

This ensured that valid factors for refining the framework were only those that were strongly agreed upon. Consensus was determined by averaging expert ratings on a fivepoint Likert scale, and ratings of 4 (Agree) and 5 (Strongly Agree). If an evaluation criterion was agreed more often across successive rounds, this meant that the framework was refined successfully according to expert recommendations. Progressively, consensus percentages increased with refinements in each Delphi round as RECTV's prioritization methodology became more accurate, transparent, and adaptable with expert input. By Round 3, all core prioritization factors had attained near or full consensus, which proved the theoretical and practical applicability of RECTV as a structured decision making model.

From the successful completion of the Delphi study, RECTV was shown to be theoretically and practically sound, which means that it is both conceptually robust and empirically applicable. The expert driven refinement process gave the framework enough strength to provide agile and data driven prioritization that is very useful for product managers, software teams and business strategists. The next section compares RECTV with existing prioritization models and shows that RECTV outperforms them in terms of decision making and industry relevance.

IV. COMPARATIVE ANALYSIS OF RECTV AGAINST TRADITIONAL PRIORITIZATION MODELS

The comparative analysis of RECTV with traditional prioritization models was conducted to evaluate its effectiveness in addressing critical limitations observed in existing frameworks such as MoSCoW, RICE, AHP, and the Kano Model. Traditional prioritization techniques have been widely utilized in software development and product management; however, they exhibit several shortcomings, including the absence of structured risk assessment, the lack of cost considerations, and failure to integrate confidence-based decision-making. Consequently, RECTV was created to use a well-structured, measurable and risk-conscious method for prioritizing to improve the accuracy of decisions. The evaluation of RECTV against previous prioritization models was based on a review of 15 important criteria. Assessment

TABLE 4. Delphi evaluation results.

Evaluation Criteria	Round 1 Consensus (%)	Round 2 Consensus (%)	Round 3 Consensus (%)	Final Observations
Sufficiency of RECTV factors (V, E, C, T, R)	83%	92%	100%	All core factors validated as essential.
Accuracy of RECTV scoring formula	78%	92%	98%	Scoring mechanism refined and validated.
Risk Awareness and Sensitivity	60%	85%	100%	Risk assessment now fully optimized.
Adaptability Across Industries	70%	85%	100%	Customization mechanism successfully introduced.
Confidence-Based Prioritization	75%	88%	100%	Confidence weighting improved for better prioritization.
Startup and Agile Suitability	80%	90%	100%	Framework confirmed as ideal for fast-paced environments.
Enterprise-Level Scalability	60%	78%	95%	Large-scale decision support now validated.
Transparency and Repeatability of Decisions	68%	88%	98%	RECTV model ensures structured, data-driven prioritization.

was conducted by considering 15 essential factors that are required for an effective prioritization model. To check if a model is unbiased, look for a numerical scoring system that is used to rank the options. Systems such as MoSCoW do not use scores to decide priorities, so the results can be unpredictable and different each time. The process used by RECTV ensures that stakeholders can make repeated priority decisions in an efficient manner.

Any decision-making process should account for risk, since it must handle unknown elements and related issues and barriers linked to specific requirements. Models that support prioritization in the field do not involve risk assessment, but RECTV does by including risk evaluation in the way it evaluates assets. When resources are scarce, companies must rely on analyzing costs to make the best decisions during the evaluation process. By using RECTV, organizations are able to choose features that help their finances, whereas traditional methods often overlook such concerns.

To support the transparency and data-driven character of RECTV we provide a small example where a product team used both RECTV and RICE on the same 5 features. RECTV emerged a lower amount of effort feature with a high confidence and quick Time to Value as the paramount priority, but RICE preferred a higher-value but riskier feature. Once the RECTV-prioritized feature was in place, it introduced the earlier engagement of users, proving the model was correct, and the delivery impact was real. This shows that RECTV is superior in balancing measurable factors within the agile environments.

There is a main method in evaluation known as confidence-based prioritization that helps separate reliable requirements from those that are not. Unlike the RICE approach, most common methods fail to identify the difference between validated and uncertain features. The RECTV model addresses this issue by requiring confidence from the team, so businesses only focus on features that have been proven in the market. Because these companies operate at a fast pace, they must show value as quickly as possible. This model is superior to older ones since it steers agile teams and startups to work on features that help them achieve fast results, something traditional approaches do not handle well. Evaluating decision systems, we focus on their flexibility, since most organizations perform their tasks using dynamic and repeated steps. Its structure in the RECTV model is flexible so that important business tasks can be changed quickly, unlike the inflexible way AHP handles these changes.

Run tests on the model to check if it is suitable for companies working in dangerous conditions and with limited resources. Traditional models are suited for corporations that have been in business for years, while RECTV is geared towards start-up businesses that need to make decisions quickly and cost-effectively. It is possible to use this model for companies of every size and sector because of its scalability.

Although in Table 5 the comparison criteria have been equally displayed to ensure transparency, they might not be relevant to all domains. As an example, a regulated industry such as healthcare or a financial process can have more value to Risk Awareness and Technical Feasibility, whereas a consumer-facing SaaS can put more emphasis on UX Impact and Customer Validation. Practitioners who use it are urged to exercise contextual weighting of the criteria of comparison following their priorities in the field. The way developers organize tasks depends on user experience impact, as this factor helps determine how well customers and user experience are considered. While most models today focus on

TABLE 5. Comparative findings of RECTV vs. Traditional models.

Criteria	MoSCoW	RICE	AHP	Kano	RECTV
Quantifiable Scoring	✗	✓	✓	✗	✓
Risk Awareness	✗	✗	✓	✗	✓
Cost Consideration	✗	✗	✓	✗	✓
Confidence-Based Prioritization	✗	✓	✗	✓	✓
Time to Value (TtV) Consideration	✗	✗	✗	✗	✓
Adaptability to Agile Environments	✗	✓	✗	✗	✓
Startup-Friendly Prioritization	✗	✓	✗	✗	✓
Scalability Consideration	✗	✗	✓	✗	✓
User Experience Impact	✗	✗	✗	✓	✓
Revenue & Market Demand	✗	✗	✗	✗	✓
Customer Validation Influence	✗	✓	✗	✓	✓
Effort-Based Prioritization	✗	✓	✓	✗	✓
Integration Complexity Consideration	✗	✗	✓	✗	✓
Decision Transparency	✗	✓	✓	✗	✓
Data-Driven Prioritization	✗	✓	✓	✗	✓

whether something is technically possible and how it affects the business, they do not consider what customers require. RECTV introduces user experience elements into its framework so that usability is treated with the desired attention. The order of importance in profit-based sectors depends largely on demand and the production of revenue. When using the RECTV system, organizations can include market-related topics in their decisions to form strategies that boost business revenue and satisfy market needs.

For feature placement, the customer's input should be a key consideration. While other models do not check for customer feedback, RECTV relies on information from real customers to steer decisions, rather than depending on ideas from within the organization. Ensuring that models are both valuable for business and well-managed in their development prevents companies from spending a lot on minor features. RECTV applies effort requirements to ensure that resource use supports the effectiveness of the business.

It is important to assess how complex system integration is in technical environments, as leaders use this information to decide how different features are arranged in the current infrastructure. Traditional prioritization methods neglect system integration intricacies so implementation encounters unanticipated problems when delays occur. RECTV includes an evaluation system to determine integration complexity which allows for appropriate selection of features that are technically feasible. Decision transparency constitutes an essential requirement for stakeholders to comprehend and understand the basis of their decided priorities. The RECTV structured scoring model enables transparent prioritization by delivering repeatable and data-based explanations for decision-making processes. The final principle of RECTV-based decision-making depends on data-driven methods to direct decisions toward empirical observations and escape the use of subjective opinions. Traditional approaches about prioritization utilize expert opinions without quantitative validation while RECTV implements structured data inputs which lead to measurable and objective prioritization methods.

The analyses of the 15 factors using the evaluation criteria are found in Table 2. The model RECTV is better than regular models because it is flexible, uses structured scoring, assesses risks, adapts to start-ups, takes into account financial restrictions and relies on confidence in decision making. RECTV is a preferred choice in areas that require quick decisions and use of data, especially when resources and time are scarce. According to the study, RECTV usually excels over traditional prioritization models as it uses a well-structured, risk-focused and data-based approach. Unlike the other two methods, RECTV depends on data and uses a method that is both clear and unbiased. The process of AHP is exact, but it often becomes complex and rigid in real-life situations. Even though RICE uses a prioritization approach, it does not take into account risk assessment and feasibility which are necessary in situations that change frequently. Although RECTV is excellent in a wide range of assessment criteria, it demands structured data entry and a little more learning time compared to a basic framework such as MoSCoW or Kano. Accordingly, depth and usability are a trade-off, organizations have to choose between the accuracy and the time and effort they have to invest into adopting the model successfully.

V. DISCUSSION

Based on the research, RECTV is a well-structured system that uses information to handle prioritizing requirements, because it addresses the weaknesses found in MoSCoW, RICE, AHP and Kano. Many aspects that contribute to comparison prove that RECTV is ideal for startups and agile groups due to its flexibility, sensitivity to risks, cost analysis and reliance on confidence. After outlining how the findings can be used, this paper covers RECTV's strengths and weaknesses and suggests ways for future research.

A. PRACTICAL IMPLICATIONS OF THE RECTV FRAMEWORK

The essential findings indicate that common prioritization approaches do not consider monitoring risks, budgeting and level of confidence while making decisions. Since MoSCoW and AHP depend on people's opinions, the evaluation results may often differ from each other. The scoring approach in RECTV ensures the selection of priorities is measurable, open and can be reproduced. The RECTV process uses reliability measures to ensure that projects focus on data-backed items rather than those that are simply guessed at and unproven. Since startups are short on resources and regularly develop their product, every feature should bring direct value to the business and help the company grow in the market. Using Time to Value (TtV) makes it possible for organizations to quickly get results from the features they implement to meet the demands of an evolving industry.

In the RICE model, the confidence factor is included but it does not assess risks, but RECTV accounts for risk assessment for all its decisions. The structure provides companies with a good balance between their planning and the difficulties they may face in carrying out those plans. The focus on finances in RECTV ensures that funds are not used for projects that will not be profitable.

The RECTV model provides complete customization of factor weight, and an organization can prioritize based on its own particular objectives, domain demands, and product maturity plans. This ability to flex makes the model capable of responding to the different risk tolerances, time to market pressures or technical limitations without changing the basic framework. As an example, a company working in a highly regulated industry can place more emphasis on Risk and Technical Feasibility and focus on features that reduce integration difficulties and guarantee compliance. The weight ratio in such a situation may be: Value (0.2), Confidence (0.2), Effort (0.1), Risk (0.3), and Feasibility (0.2). Conversely, an organization that has fast-paced market delivery may value more on Value and Confidence with less emphasis on risk, and the weight used may be: Value (0.4), Confidence (0.3), Effort (0.1), Risk (0.1), and Feasibility (0.1). The examples provide how RECTV achieves scoring consistency in favor of domain-sensitive prioritization. This dynamic feature is in tandem with the general modeling practice where tuning of weights improves the relevance, robustness, and accuracy of the decision of the model in diverse operating conditions.

B. STRENGTHS OF THE RECTV FRAMEWORK

Using Table 4, the study demonstrates how the RECTV framework exceeds traditional models due to its many benefits. The main benefit of using RECTV is that it assigns numerical ratings, unlike the broad categories found in MoSCoW and Kano. Through this system, decisions are made in an unbiased way using data, allowing the same priorities to be chosen each time. With the RECTV model, decision-makers can explain their choices and the final results are easier for both business and technical stakeholders to relate to.

The greatest benefit of RECTV is that it covers both the evaluation of risks and the analysis of costs, unlike other systems. According to RECTV, financial restrictions and risks are considered during decisions, whereas traditional systems do not consider these factors. The approach prevents both startups and organizations that watch their budget from starting very expensive risky projects without first having the resources for them. Thanks to its risk assessment feature, Vectors helps companies prevent unplanned issues that may arise during implementation.

Due to RECTV, organizations are able to recognize which of their ideas are truly valuable. RETV follows an approach where features supported by research and customer opinions are given more priority. With RECTV, resources are distributed to projects with the highest likelihood of success and few risks. Using validated data, RECTV ensures industries with short development periods get the most value from their decisions.

C. LIMITATIONS AND CHALLENGES

Rectification of RECTV's flaws does not mean all its problems have been solved. The biggest weakness is that companies have to assign figures to V, E, C, T, R factors, while assessing confidence and risk depends on personal opinions. Any company that has trouble assessing confidence and risk accurately finds it challenging to use this model since it requires making decisions based on facts and not guessing. Because the scoring system in RECTV is used regularly, it should be updated from time to time to ensure its weighting factors match the requirements of the business. After some period, a high-risk feature may be assessed differently which can result in new research findings that make the feature low-risk and alter the number of score points assigned. Researchers in the future should design systems that let machine learning models update their weights based on what they learn while operating in real life.

While RECTV is highly adaptable for agile work, using it in very large companies often requires individual programming. The field should develop methods to use RECTV in various decision issues by taking both work procedures and standards, as well as satisfaction of clients, into account.

Although RECTV offers a systematic and flexible model, it is necessary to mention the possible difficulties in implementation. Specifically, such score allocation as of the factor like Confidence or Risk could be rather subjective and in the case of a project in an early phase of the development, or in case of data-poor conditions. To overcome this, it is advisable to use consensus-based methods (e.g. expert panels, Delphi method) or reference to historic benchmarks in order to calibrate inputs. Reducing bias and increasing consistency among the stakeholders can also be achieved through the application of such techniques as weighted averages or confidence intervals.

VI. CONCLUSION

The RECTV framework implements an organized quantitative model for requirements prioritization which fixes various weaknesses found in standard models such as MoSCoW and RICE and AHP and Kano. The combination of measurable scores along with risk identification along with cost analysis and confidence-driven decisions and flexible framework structure makes RECTV produce transparent repeatable prioritization decisions that follow business goals. The evaluation of RECTV demonstrates outstanding capability in managing business effectiveness with practical feasibility and financial limitations and technological dependencies for startups and agile development teams and restricted resource situations. Future advancements in RECTV technology will include automated scoring using AI systems and immediate data processing to enhance its both accuracy and applicability aspects despite facing present challenges with scorer subjectivity and system recalibration requirements. The RECTV framework creates a scalable system that lets organizations undertake strategic decisions while remaining aware of financial risks in their quest to succeed in a changing competitive market. Future work will be the empirical validation of RECTV in a real-world environment and longitudinal research on RECTV and pilot implementations in the industry. Such studies will assist in evaluating the long-term effects of the framework, its ease of adoption, or its ability to scale in terms of organizations of various sizes, business area, and development culture.

REFERENCES

[1] K. Schlögel, *Moscow*. Hoboken, NJ, USA: Wiley, 1937.

[2] K. S. Ahmad, N. Ahmad, H. Tahir, and S. Khan, "Fuzzy_MoSCoW: A fuzzy based MoSCoW method for the prioritization of software requirements," in *Proc. Int. Conf. Intell. Comput., Instrum. Control Technol. (ICICICT)*, Jul. 2017, pp. 433–437.

[3] F.-F. Chua, T.-Y. Lim, B. Tajuddin, and A. P. Yanuarifiani, "Incorporating semi-automated approach for effective software requirements prioritization: A framework design," *J. Informat. Web Eng.*, vol. 1, no. 1, pp. 1–15, Mar. 2022.

[4] W. Benjamin, *Moscow Diary*. Cambridge, MA, USA: Harvard Univ. Press, 1986.

[5] J. A. Moscow and K. H. Cowan, "Multidrug Resistance1," *JNCI J. Nat. Cancer Inst.*, vol. 80, no. 1, pp. 14–20, Mar. 1988.

[6] G. W. B. Smith, *My Three Years in Moscow*. Canada: Pickle Partners, 2018.

[7] R. Confalonieri, G. Bellocchi, S. Bregaglio, M. Donatelli, and M. Acutis, "Comparison of sensitivity analysis techniques: A case study with the Rice model WARM," *Ecological Model.*, vol. 221, no. 16, pp. 1897–1906, Aug. 2010.

[8] F. Pattyn, "Identifying key requirements prioritization criteria for software startups survival: A systematic literature review," Ghent Univ., Ghent, Belgium, Tech. Rep., 2024.

[9] F. Pattyn and P. Goetz, "Time to value (TtV) as a missing requirements prioritization criterion for startup survival: Insights from founders vs. systematic literature review," *Int. J. Sustainability Manage. Inf. Technol.*, vol. 11, no. 1, pp. 45–66, Apr. 2025.

[10] J. Ali Khan, I. U. Rehman, Y. H. Khan, I. Javed Khan, and S. Rashid, "Comparison of requirement prioritization techniques to find best prioritization technique," *Int. J. Mod. Educ. Comput. Sci.*, vol. 7, no. 11, pp. 53–59, Nov. 2015.

[11] R. Sonani, "Hierarchical multi-agent reinforcement learning framework with cloud-based coordination for scalable regulatory enforcement in financial systems," *Spectr. Res.*, vol. 3, no. 2, p. 15, 2023.

[12] J. Mikulić and D. Prebežac, "A critical review of techniques for classifying quality attributes in the Kano model," *Manag. Service Quality, Int. J.*, vol. 21, no. 1, pp. 46–66, Jan. 2011.

[13] A. Shahin, M. Pourhamidi, J. Antony, and S. Hyun Park, "Typology of Kano models: A critical review of literature and proposition of a revised model," *Int. J. Quality Rel. Manage.*, vol. 30, no. 3, pp. 341–358, Mar. 2013.

[14] V. Govindarajan, R. Sonani, and P. S. Patel, "Secure performance optimization in multi-tenant cloud environments," *Ann. Appl. Sci.*, vol. 1, no. 1, p. 9, 2020.

[15] M. Szymczak and K. Kowal, "The Kano model: Identification of handbook attributes to learn in practice," *J. Workplace Learn.*, vol. 28, no. 5, pp. 280–293, Jul. 2016.

[16] G. Ferdinands, R. Schram, J. De Bruin, A. Bagheri, D. L. Oberski, L. Tummers, J. J. Teijema, and R. Van De Schoot, "Performance of active learning models for screening prioritization in systematic reviews: A simulation study into the average time to discover relevant records," *Systematic Rev.*, vol. 12, no. 1, p. 100, Jun. 2023.

[17] M. Z. Khan, M. Shoaib, M. S. Husain, K. Ul Nisa, and M. T. Quasim, "Enhanced mechanism to prioritize the cloud data privacy factors using AHP and TOPSIS: A hybrid approach," *J. Cloud Comput.*, vol. 13, no. 1, p. 42, Feb. 2024.

[18] R. R. De Almeida, R. Do Nascimento Ribeiro, C. Treude, and U. Kulesza, "Business-driven technical debt prioritization: An industrial case study," in *Proc. IEEE/ACM Int. Conf. Tech. Debt (TechDebt)*, May 2021, pp. 74–83.

[19] D. L. Moody, "Theoretical and practical issues in evaluating the quality of conceptual models: Current state and future directions," *Data Knowl. Eng.*, vol. 55, no. 3, pp. 243–276, Dec. 2005.

[20] S. Fatichi, E. R. Vivoni, F. L. Ogden, V. Y. Ivanov, B. Mirus, D. Gochis, C. W. Downer, M. Camporese, J. H. Davison, B. Ebel, N. Jones, J. Kim, G. Mascaro, R. Niswonger, P. Restrepo, R. Rigon, C. Shen, M. Sulis, and D. Tarboton, "An overview of current applications, challenges, and future trends in distributed process-based models in hydrology," *J. Hydrol.*, vol. 537, pp. 45–60, Jun. 2016.

[21] A. Baklanov et al., "Online coupled regional meteorology chemistry models in Europe: Current status and prospects," *Atmos. Chem. Phys.*, vol. 14, no. 1, pp. 317–398, Jan. 2014.

[22] N. B. Robinson, K. Krieger, F. Khan, W. A. Huffman, M. Chang, A. Naik, R. Yongle, I. Hameed, K. H. Krieger, L. N. Girardi, and M. Gaudino, "The current state of animal models in research: A review," *Int. J. Surgery*, vol. 72, pp. 9–13, Jan. 2019.

[23] M. Pelowski, P. S. Markey, J. O. Lauring, and H. Leder, "Visualizing the impact of art: An update and comparison of current psychological models of art experience," *Frontiers Human Neurosci.*, vol. 10, p. 160, Apr. 2016.

[24] A. Maul, "Rethinking traditional methods of survey validation," *Meas., Interdiscipl. Res. Perspect.*, vol. 15, no. 2, pp. 51–69, Apr. 2017.

[25] S. Greenland, R. Daniel, and N. Pearce, "Outcome modelling strategies in epidemiology: Traditional methods and basic alternatives," *Int. J. Epidemiol.*, vol. 45, no. 2, pp. 565–575, Apr. 2016.

[26] B. Dong and X. Wang, "Comparison deep learning method to traditional methods using for network intrusion detection," in *Proc. 8th IEEE Int. Conf. Commun. Softw. Netw. (ICCSN)*, Jun. 2016, pp. 581–585.

[27] R. Renau and M. Luisa, "A review of the traditional and current language teaching methods," *Int. J. Innov. Res. Educ. Sci.*, vol. 3, no. 2, pp. 82–88, 2016.

[28] M. Eid, C. Geiser, and T. Koch, "Measuring method effects: From traditional to design-oriented approaches," *Current Directions Psychol. Sci.*, vol. 25, no. 4, pp. 275–280, Aug. 2016.

[29] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, p. 116.

[30] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.

[31] S. T. Motlagh, A. Ibrahim, S. S. Heydari, and K. El-Khatib, "Multi-objective optimization for managing disruption risk in SDN," in *Proc. 20th Int. Conf. Design Reliable Commun. Netw. (DRCN)*, May 2024, pp. 1–8.

[32] S. Vijayakumar, K. Prasad, and R. Holla, "Assessing the effectiveness of MoSCoW prioritization in software development: A holistic analysis across methodologies," *EAI Endorsed Trans. Internet Things*, vol. 10, p. 9, Oct. 2024. [Online]. Available: https://publications.eai.eu/index.php/IoT/article/view/6515/3439

[33] O. Nesterenko and O. Trofymchuk, "Decision support for requirements prioritization in software engineering," Eur. School Bus., Ukraine, Tech. Rep. 3806, 2024, pp. 50–61.
[34] B. K. Eldrandaly, "A framework of type-2 neutrosophic for requirements prioritization," *Neutrosophic Sets Syst.*, vol. 53, pp. 421–432, Mar. 2023.
[35] M. Kassab and N. Kilicay-Ergin, "Applying analytical hierarchy process to system quality requirements prioritization," *Innov. Syst. Softw. Eng.*, vol. 11, no. 4, pp. 303–312, Dec. 2015.
[36] J. Ali, W. Afzal, Izaz-Ur-Rehman, I. Qasim, S. Poor, and Y. Hayat, "An evaluation of requirement prioritization techniques with ANP," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 7, p. 11, 2016.
[37] A. Bakhitar, A. Hannan, A. Basit, and J. Ahmad, "Prioritization of value based services of software by using AHP and fuzzy Kano model," in *Proc. Int. Conf. Comput. Social Sci.*, vol. 8, 2015, pp. 25–27.
[38] E. O. C. Mkpojiogu and N. L. Hashim, "Quality based prioritization: An approach for prioritizing software requirements," *J. Telecommun.*, vol. 9, no. 2, pp. 17–21, 2017.
[39] M. L. Frigo and R. J. Anderson, "Risk management frameworks: Adapt, don't adopt: Here's a primer on how to use two well-known approaches," *Strategic Finance*, vol. 95, no. 7, pp. 49–54, 2014.
[40] Z. T. Şimşit, N. S. Günay, and Ö. Vayvay, "Theory of constraints: A literature review," *Proc.-Social Behav. Sci.*, vol. 150, pp. 930–936, Sep. 2014.
[41] R. F. Bortolini, M. Nogueira Cortimiglia, A. D. M. F. Danilevicz, and A. Ghezzi, "Lean startup: A comprehensive historical review," *Manage. Decis.*, vol. 59, no. 8, pp. 1765–1783, Aug. 2021.
[42] P. Lai, "The literature review of technology adoption models and theories for the novelty technology," *J. Inf. Syst. Technol. Manage.*, vol. 14, no. 1, pp. 21–38, Jun. 2017.
[43] E. J. Mishan and E. Quah, *Cost-Benefit Analysis*. Evanston, IL, USA: Routledge, 2020.

FRÉDÉRIC PATTYN (Member, IEEE) was born in Kortrijk, Belgium, in 1987. He is currently pursuing the Ph.D. degree in applied economic sciences with Ghent University, Belgium. He has been a Visiting Researcher with the Free University of Bozen-Bolzano, Italy, since 2024. His research interests include the intersection of product management, software startups, and artificial intelligence (AI).

RAFAQAT KAZMI received the Ph.D. degree from the University of Technology Malaysia, in 2017. He is currently an Associate Professor with the Department of Software Engineering, The Islamia University of Bahawalpur, Pakistan. He was also part of the Project "End-to-End Cohesive Approach for Agile Software Development in a Cloud Computing Environment" funded by Ministry Science Malaysia. He contributed more than 40 peer-reviewed articles, with a focus on theoretical computing with practical applications. His current research interests include software testing, requirements engineering, agile methods, social media, machine learning, deep learning, social engineering, and the development of intelligent and smart systems. He received the International Doctoral Fellowship twice during the Ph.D. degree.

PETER GOETZ was born in Antwerp, Belgium, in 1970. He is currently the Founder and the CEO of New Data Vision Ltd., and the President of the New Data Research Institute, where he focuses his professional and academic work on the intersection of deep pattern analysis, financial and quantum technologies, software startups, and artificial intelligence (AI). He is also an Active Investor in research, emerging technologies, and visionary tech founders with the potential to drive global impact.

AFNAN ALDHAHRI received the B.Sc. degree (Hons.) in computer science from Umm Al-Qura University (UQU), Mecca, Saudi Arabia, in 2008, and the M.Sc. degree (Hons.) in computer science and the Ph.D. degree in software engineering from the University of Colorado Boulder, CO, USA, in 2016 and 2020, respectively.

Her study involved cloud-based software architecture development and big data analytics. She is currently an Assistant Professor with the Department of Software Engineering, UQU. Her research interests include intelligent software engineering, cloud computing, and big data infrastructure.

AQSA received the M.S. degree in computer science from The Islamia University of Bahawalpur, Pakistan, in 2022. She is currently pursuing the Ph.D. degree with Wuhan University. She previously served as a Lecturer with the Department of Computer Science, COMSATS University Islamabad, Sahiwal Campus. Her research interests include the integration of Internet of Things (IoT), machine learning, and artificial intelligence (AI) to develop intelligent systems for real-world applications, including environmental monitoring, precision agriculture, healthcare, and digital forensics.