

Economic sciences

UDC 005.8:004

Yatsenko Viktoriia

Product Manager at Engine AI

Taras Shevchenko National University of Kyiv

ORCID: 0009-0008-8774-3339

COMPARATIVE ANALYSIS OF FEATURE PRIORITISATION METHODS: ICE, RICE, WSJF, MOSCOW AND THEIR APPLICABILITY FOR STARTUPS

Summary. *Key feature prioritization methods (ICE, RICE, WSJF, MoSCoW) are analyzed in the context of startups. Based on a practical case study, it is demonstrated how the choice of a framework influences product strategy. A significant discrepancy is identified between theoretical models and the actual implementation order, which is driven by business needs. It is substantiated that frameworks serve as decision-support tools, and their selection should align with the company's stage of development.*

Key words: *feature prioritization, startup, product management, ICE, RICE, WSJF, MoSCoW.*

Problem Statement. In today's highly competitive business environment, characterized by rapid technological changes and dynamic market conditions, startups face a unique set of challenges that threaten their survival. Unlike mature corporations, startups operate under conditions of acute resource scarcity – financial, human, and temporal – which makes every strategic decision critically important. In such circumstances, feature prioritization ceases to be a routine product management task and transforms into a fundamental survival mechanism and a key instrument of strategic management. The inefficient allocation of

limited resources to the development of features that do not meet real market needs is one of the main reasons for the failure of young companies, leading to the depletion of financial reserves before reaching the break-even point or achieving product-market fit [1]. Thus, the ability to make informed decisions about what to build and in what sequence becomes the defining factor that separates successful innovative projects from those that disappear from the market.

The choice of a prioritization methodology is not merely a tactical decision but a profoundly strategic one that directly impacts the product's development trajectory, its market position, and its long-term viability. An inappropriate framework can lead to the creation of a feature-laden product that fails to solve users' key problems, resulting in a diluted value proposition, a diminished user experience, and the waste of precious development resources. Conversely, a well-considered approach to prioritization allows the team to focus its efforts on features that generate the most value for the target audience, thereby increasing customer satisfaction and retention rates, which are critical for sustainable growth. The prioritization framework chosen by a startup serves as an indicator of its organizational maturity, data-driven culture, and depth of understanding of the economic principles of product development. In the early stages, when data is limited and decisions are made centrally based on the founders' vision, companies often rely on intuition or informal methods, such as HiPPO (Highest Paid Person's Opinion). The transition to qualitative yet structured approaches, like MoSCoW, or fast and subjective ones, like ICE, signifies a move towards more formalized thinking aimed at defining an MVP and ensuring a high velocity of experimentation. Thus, a framework is not merely a tool but also a mirror reflecting the startup's evolution from an idea to an economically rational organization.

Aim and Objectives. The primary aim of this article is to conduct a comprehensive comparative analysis of the ICE, RICE, WSJF, and MoSCoW

feature prioritization frameworks to determine their relative advantages, disadvantages, and optimal conditions for application in the environment of technology startups. To achieve this aim, a series of objectives must be addressed. First, to deconstruct the theoretical foundations, key principles, and operational mechanisms of each of the four frameworks. Second, to critically evaluate the ability of each methodology to effectively respond to the key challenges of startups, including resource constraints, high levels of uncertainty, risk management, and the need for rapid iterations. Third, to compare the frameworks based on key criteria such as objectivity, data dependency, ease of implementation, and alignment with strategic economic goals. Fourth, to examine the practical application of the ICE methodology in organizing a startup's workflow, as well as the potential effect of replacing this methodology with other frameworks – RICE, WSJF, and MoSCoW.

Methodology. The research is based on the methodology of qualitative comparative analysis. To formulate the theoretical foundation, a review of academic literature in the fields of product management, software engineering, innovation management, and entrepreneurship was conducted. The described theoretical concepts were tested in practice by examining a real-world case of applying the ICE methodology to solve the problem of task prioritization within a startup project.

Results and Discussion. In startups, where time, budget, and personnel resources are limited, the prioritization of product features becomes a critical component of success. Prioritization allows for the concentration of available resources on the most important aspects of the product, avoiding unnecessary expenditure on low-value features. One of the most common prioritization methods today is the MoSCoW method, which originates from the Dynamic Systems Development Method (DSDM) agile methodology and focuses primarily on scope management and aligning stakeholder expectations. Its name is an acronym for the four categories into which requirements or functional elements

are divided: Must Have, Should Have, Could Have, and Won't Have (this time) [2, p. 768]. The «Must Have» category includes essential requirements, without which the product or release would be considered non-functional and would have no value. «Should Have» refers to important features that significantly enhance the product's value, but their absence is not critical, and a temporary workaround may exist for them. «Could Have» are desirable but less important elements that will be implemented only if time and resources are available. Finally, the «Won't Have» category clearly defines those features that are deliberately excluded from the current scope of work, which helps to prevent «scope creep». Fundamentally, MoSCoW is not a tool for detailed ranking but rather a qualitative framework for broad classification that helps the team and stakeholders reach a consensus on a minimum viable feature set.

For early-stage startups, the main advantage of the MoSCoW method lies in its simplicity and direct application in defining the Minimum Viable Product (MVP). The «Must Have» category naturally outlines the core functionality required for the initial launch, ensuring that the team focuses on delivering a truly viable solution that can be introduced to the market. This approach significantly simplifies communication with founders, investors, and early customers, as it clearly articulates what will and will not be implemented in the upcoming iteration. The explicit definition of the «Won't Have» category is a powerful tool for managing expectations, helping to avoid misunderstandings and disappointments in the future [3, p. 2]. Due to its qualitative nature, the method does not require complex calculations or large volumes of data, making it ideal for an environment where uncertainty prevails and decisions are often based on hypotheses and qualitative insights.

Despite its advantages, MoSCoW has significant limitations that become apparent as the complexity of the product and the team grows. The main drawback is the high level of subjectivity and the lack of granularity. The line between the «Should Have» and «Could Have» categories is often blurred and can become a

source of prolonged discussions and conflicts within the team. More critically, the framework offers no mechanism for prioritization within a single category. For example, if there are ten «Should Have» features on the list, MoSCoW does not provide an answer as to which one should be implemented first. This problem becomes particularly acute under tight resource constraints when difficult trade-off decisions need to be made. Furthermore, the method does not account for economic factors such as development cost, potential revenue, risks, or audience reach, which makes it insufficiently effective for making decisions based on the optimization of business metrics [4, p. 2]. The successful application of MoSCoW largely depends on the team's ability to reach a consensus, which can be a challenging task in a dynamic startup environment with differing stakeholder views.

In turn, the ICE framework, popularized by Sean Ellis, one of the founders of the growth hacking concept, is a prioritization model designed for the needs of rapid experimentation and iterative development in the dynamic environment of startups. The acronym ICE stands for Impact, Confidence, and Ease [5]. Each of these three parameters is evaluated on a scale, typically from 1 to 10, and the final score is calculated as the product or arithmetic mean of these scores. «Impact» reflects the potential effect of an idea on key product metrics. «Confidence» is the team's level of certainty that the hypothesis about the impact is correct. «Ease» assesses how easily and quickly the idea can be implemented in terms of resource expenditure. The main advantage of ICE for startups in the product-market fit search stage is its extreme simplicity and speed [6]. It does not require deep data analysis and allows the team to quickly sort a large number of hypotheses, selecting those that promise the greatest return for the least effort for immediate testing.

However, the main strength of ICE – its simplicity – is also its greatest weakness. The framework is extremely subjective, which creates significant risks for the quality of the decisions made. The scores for each component – «Impact»,

«Confidence», and «Ease» – are largely based on the intuition, personal experience, and assumptions of team members. This can lead to significant discrepancies in the evaluation of the same idea by different people, making the final result inconsistent and difficult to justify to stakeholders. Subjectivity also opens the door for cognitive biases, where «favorite» ideas or ideas proposed by more authoritative team members receive unjustifiably high scores. Because of this, ICE is considered a «good enough» tool for quickly sorting ideas for experiments, but an unreliable method for building a long-term, well-founded product plan that requires greater objectivity and transparency [7].

The RICE framework, developed and implemented at Intercom, is a direct evolution of ICE, created to overcome its key shortcomings by adding quantitative and more objective criteria. The acronym RICE stands for Reach, Impact, Confidence, and Effort [8, p. 96]. The key innovations are the replacement of the abstract «Ease» with the more concrete «Effort» and the addition of a new component – «Reach». «Reach» is a quantitative metric that estimates how many users an initiative will affect over a specific period (e.g., number of users per month). This parameter forces product teams to turn to analytics and real data rather than relying on assumptions. «Effort» is estimated in person-months, which provides a clearer understanding of the cost of implementing an idea. The final score is calculated using a formula that is essentially an assessment of the benefit-to-cost ratio [8, p. 96]:

$$RICE = \frac{ach \times Impact \times Confidence}{Effort}$$

The RICE method is an ideal tool for startups in the growth stage, i.e., those that already have an initial user base and the infrastructure to collect analytical data. The availability of real data to calculate «Reach» significantly increases the objectivity of the prioritization process. This framework provides a structured, consistent, and reproducible process that helps minimize the influence of personal biases and allows product managers to defend their decisions with sound

arguments before management and other stakeholders [9]. RICE strikes an optimal balance between the speed inherent in ICE and the analytical depth of more complex economic models. It enables more balanced and data-informed decisions without requiring overly complex economic calculations, making it a practical and effective tool for product teams striving for sustainable and manageable growth.

The final method we will consider is Weighted Shortest Job First (WSJF), which is based on a fundamental economic concept formulated by Donald Reinertsen in his work «The Principles of Product Development Flow» [10]. The central element of this approach is the Cost of Delay (CoD) – a metric that quantifies the economic loss (in monetary terms) that a company incurs for each unit of time (week, month) that the release of a particular feature or product is delayed [11, p. 12]. The formula for determining WSJF is as follows:

$$WSJF = \frac{Cost\ of\ Delay}{Job\ Duration}$$

This simple formula allows for the maximization of the total profit over the product's lifecycle, as it prioritizes those tasks that deliver the greatest economic value in the shortest amount of time. The application of WSJF forces teams to change their thinking paradigm: instead of focusing separately on potential revenue or development costs, they begin to think in terms of the economic impact of time, which is key to making optimal decisions under resource constraints.

A comparative analysis of the frameworks will be carried out in Table 1 to determine the optimal stage for their practical application.

The analysis of the four prioritization frameworks – MoSCoW, ICE, RICE, and WSJF – leads to a key conclusion: there is no single universally best method. The effectiveness of each depends on the unique conditions in which a startup operates, particularly its stage of development, data availability, organizational maturity, and strategic priorities. Attempting to apply a complex economic model at a stage when the company does not yet have a product is as inappropriate as

using simple qualitative classification to manage multi-million dollar development investments. Therefore, we can state the necessity of a situational approach, where the choice of a prioritization tool is a deliberate strategic decision corresponding to the company's current challenges and capabilities.

Table 1

Comparative analysis of MoSCoW, ICE, RICE, and WSJF prioritization methods

Criterion	MoSCoW	ICE	RICE	WSJF
Core principle	Scope categorization	Velocity of experimentation	Objective cost-benefit analysis	Economic sequencing optimization
Key components	Must, Should, Could, Won't	Impact, Confidence, Ease	$(\text{Reach} \times \text{Impact} \times \text{Confidence}) / \text{Effort}$	CoD / Job Size
Primary strength	Clear MVP definition; expectation management	Simplicity and speed; low data dependency	Balanced and objective; defensible decisions	Economic rigor; focus on Cost of Delay
Primary limitation	Subjectivity; no ranking within categories	High subjectivity; risk of bias	Requires data for "Reach"; more complex than ICE	Complexity; abstraction from real currency
Ideal startup context	Pre-Seed or Seed: Defining the MVP	Early Stage: Searching for PMF)	Growth Stage: Scaling and optimization	Scaling Stage: Portfolio management

Let us consider the application of these methods using the example of the author's experience in a startup operating in the competitive B2B sector and focused on providing services under a subscription model. At the time of the analysis, the company's key strategic priority was identified as increasing customer retention and overall satisfaction. Product managers faced a classic dilemma for such conditions: a large backlog of potentially valuable initiatives with the limited resources of the development team. Choosing the wrong direction could lead not only to financial losses but also to the churn of users who did not feel the product was improving. To minimize risks and implement a systematic approach to decision-making, it was decided to use formalized prioritization

methods. Initially, preference was given to the ICE framework, which was explained by its simplicity, speed of application, and low barrier to entry for a team that had not previously used such tools on a consistent basis.

The list of tasks to be prioritized included sixteen unique features, which can be conditionally grouped into several areas (Table 2).

Table 2

List and description of tasks for prioritization

Area	Task Name	Brief Task Description
User Experience Enhancement	Task Management System Redesign	Redesign of the task management system for a more transparent and structured interface.
	Payments Page Redesign	Update of the payments page to give clients full control over their subscriptions.
	Talent Profile & Portfolio Pages	Creation of detailed profiles for talents with portfolios to increase client trust.
	Task Rematch Functionality	Addition of a feature allowing a client to quickly request a replacement talent for a task.
Efficiency & Automation	Automated Matching Algorithm	Full automation of the algorithm for matching talent to a task to eliminate manual work.
	Talent Task Approval Workflow	Implementation of a system where several talents receive a request, and the first to confirm takes the task.
	Automated Talent Hiring Flow	Automation of part of the talent hiring process to speed up the work of the People team.
	Calendar & Calls Integration	Integration of third-party services for independent scheduling of calls between clients and talents.
	Customer Success Process Automation	Automation of routine tasks for the Customer Success team (reminders, emails) via CRM.
Engagement & Retention	Referral Page	Creation of a page where users can get a referral link to attract new clients.
	Customer Tipping for Talents	Addition of a feature allowing clients to leave tips for talents for quality work.
	Customer NPS Integration	An integrated system for collecting the Net Promoter Score to proactively track client satisfaction.
	Voice & Video Comments in Tasks	The ability to leave voice and video comments in the chat to speed up communication.
Technical Infrastructure	Talents Mobile Application	Development of a mobile application for talents for quick receipt of notifications and updates.
	Talent App Redesign	Redesign of the interface for the part of the platform intended for talents to improve their experience.
	Technical Debt Resolution	Performing work to eliminate accumulated technical debt to improve system stability.

The application of the ICE methodology took place in several stages and required close collaboration between product managers, designers, and engineers. The Impact metric was assessed on a five-point scale, where the main criterion was the answer to the question: «How significantly will this feature improve key metrics of customer retention and satisfaction?». The Confidence score was also given on a five-point scale and reflected the team's level of certainty in the initiative's success. This parameter was based on a combination of factors: the availability of qualitative data from user research, the clarity of the formulated hypothesis, technical predictability, and the absence of significant external dependencies. The most objective component was Effort, which the development team estimated in relative units based on a preliminary analysis of the complexity and scope of work. This process compelled each participant to argue their position and synchronize their vision for each initiative.

After conducting the evaluation and calculating the ICE Score, the team received a clearly structured, prioritized list (Table 3).

Table 3

ICE Score for project tasks

№	Feature Name	Impact	Confidence	Effort
1	Automated Matching Algorithm	5	5	5
2	Talent Profile & Portfolio Pages	5	5	5
3	Task Rematch Functionality	5	5	2
4	Payments Page Redesign	5	4	4
5	Talent Task Approval Workflow	5	5	4
6	Calendar & Calls Integration	5	4	5
7	Customer NPS Integration	5	5	3
8	Task Management System Redesign	4	4	4
9	Talents Mobile Application	2	3	5
10	Talent App Redesign	2	3	5
11	Customer Tipping for Talents	4	5	2
12	Automated Talent Hiring Flow	5	5	5
13	Referral Page	4	4	3
14	Voice & Video Comments in Tasks	3	4	3
15	Customer Success Process Automation	3	4	5
16	Technical Debt Resolution	1	3	3

The resulting ranking became the basis for forming the product roadmap for the upcoming year (Figure 1). The priorities were distributed quarterly, which allowed for balancing the team's workload and ensuring consistent value delivery. The first quarter (Q1) was fully dedicated to implementing the top three to four features with the highest ICE Score. The goal was to quickly demonstrate positive changes to users and gather feedback that could influence future plans. The second quarter (Q2) included both continuing work on high-ranking client-facing features and implementing several important but more complex system improvements, such as the redesign of the task management system. The third and fourth quarters (Q3–Q4) were reserved for large-scale projects requiring significant effort, as well as for lower-priority initiatives considered «desirable but not critical».

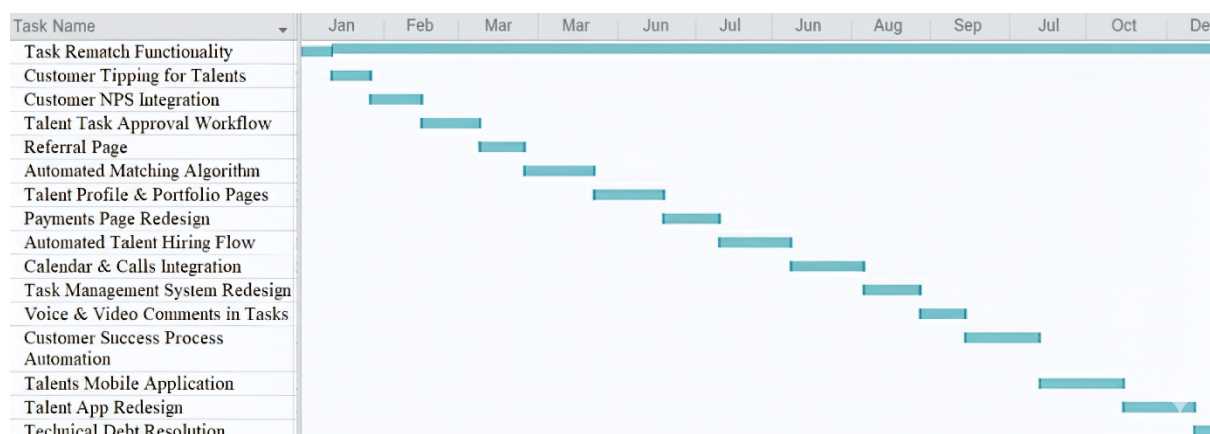


Fig. 1. Draft roadmap based on ICE prioritization

A roadmap built on the ICE framework is a clear example of a product strategy focused on «quick wins». It prioritizes initiatives that have a high potential for immediately increasing user satisfaction, even if this comes at the cost of postponing important internal optimizations. Such an approach is typical for many startups striving to prove their value to the market and customers as quickly as possible. At the same time, it carries certain risks related to the accumulation of technical debt and the neglect of projects whose effect is not immediate but is of fundamental importance for scaling the business.

It is precisely this trade-off that is the subject of further analysis when comparing with alternative prioritization methods. For this purpose, we will apply three alternative frameworks – RICE, WSJF, and MoSCoW – to the same set of sixteen features. This will not only allow us to obtain different priority lists but also to understand how the fundamental differences in the logic of each method affect the final product strategy. The analysis revealed significant discrepancies in the rankings (Table 4).

Table 4

Consolidated comparative analysis of prioritization methods

Feature Name	RICE Rank	ICE Rank	WSJF Rank	MoSCoW Category
Task Rematch Functionality	1	1	1	Must-have
Customer Tipping for Talents	2	2	2	Should-have
Customer NPS Integration	3	3	3	Must-have
Talent Task Approval Workflow	4	4	5	Should-have
Payments Page Redesign	5	6	5	Must-have
Automated Matching Algorithm	5	6	7	Should-have
Talent Profile & Portfolio Pages	5	6	7	Should-have
Task Management System Redesign	7	10	7	Must-have
Calendar & Calls Integration	7	10	8	Could-have
Referral Page	8	5	4	Could-have
Automated Talent Hiring Flow	9	6	9	Won't-have
Voice & Video Comments in Tasks	10	10	6	Could-have
Talents Mobile Application	11	14	10	Could-have
Talent App Redesign	11	14	10	Could-have
Customer Success Process Automation	13	13	12	Won't-have
Technical Debt Resolution	14	16	11	Won't-have

The analysis using the RICE framework was the first step toward expanding the initial ICE model. The key difference in this method is the addition of a fourth component – Reach, which quantitatively assesses how many users will encounter the new feature over a certain period. This parameter is intended to add objectivity to the evaluation process, reducing the weight of the subjective perception of the «Impact» metric. For a B2B platform where different features may target completely different user segments (e.g., account administrators versus daily users), this criterion is extremely important. It forces the team to think not only

about the potential strength of a feature's effect but also about the scale of that effect, which helps to avoid investing significant resources in functionality for a narrow, albeit active, audience. Applying RICE to our task list led to noticeable changes in priorities. Although the top-ranked features, such as «Task Rematch Functionality», maintained their positions due to a combination of high impact and wide reach, some other features significantly changed their standing. For example, «Referral Page», which had a fairly high ICE Score, dropped lower in the RICE ranking because its reach was assessed as medium – it is important for acquisition but is not used as frequently as the core functionality. Conversely, features embedded in the daily workflow of every user, such as «Talent Task Approval Workflow», gained additional weight.

The transition to the Weighted Shortest Job First (WSJF) method signified a paradigm shift: from evaluating individual parameters to calculating the economic cost of delay. This framework, originating from the Scaled Agile Framework (SAFe), proposes viewing priority as the ratio of the Cost of Delay to the Job Size. The Cost of Delay is the sum of three components: user-business value, time criticality, and risk reduction or opportunity enablement potential. This approach forces the team to think not so much about «what will this feature give us?» but rather about «how much will we lose if we don't do it now?». This is a fundamentally more strategic view that closely links the development team's decisions with the financial and market realities of the business.

The WSJF prioritization results once again reshuffled our backlog, although, as in the previous case, the top three leaders remained unchanged. This further underscores their exceptional value. However, beyond the top positions, certain changes occurred. A telling example is the «Referral Page» feature, which rose from 5th place in ICE to 4th in WSJF. This is because although its Reach is small, its Time Criticality for the business, related to attracting new clients, was rated very highly. At the same time, large-scale infrastructure tasks, such as the «Automated Matching Algorithm», dropped in the ranking because, despite their

high value, their urgency was deemed lower. A roadmap built on WSJF has a clear economic focus. It aims to maximize the ROI from development by sequentially implementing tasks with the highest cost of delay per unit of effort. This is an ideal tool for companies seeking to optimize their value stream and closely align development with business goals. The main challenge of this method lies in the complexity and subjectivity of estimating the components of the Cost of Delay, which requires the team to have a deep understanding of the market, business model, and strategic priorities, something that can be difficult for young teams.

Unlike the previous quantitative methods, MoSCoW is a qualitative categorization technique. It does not rank features in a linear order but rather distributes them into four categories. It does not answer the question «what to do next?» but instead helps to answer the question «what should go into the next release?». Its main strength lies in its ability to create a shared understanding and alignment among all stakeholders regarding the minimum viable scope of work. In our case, the MoSCoW categorization, based on the strategic goal of increasing customer retention, yielded interesting results. The Must-have category included not only the features with the highest RICE or WSJF scores but also those that were deemed absolutely critical for the product's survival and competitiveness. These included, for example, «Task Rematch Functionality» and «Payments Page Redesign», as they directly impacted client trust and satisfaction. The Won't-have category became a «sanitary zone» for important but not urgent internal improvements, such as hiring automation and resolving technical debt.

We can assert that MoSCoW is a powerful tool for release planning and managing expectations. It brings clarity and prevents the project scope from being «bloated» with less important tasks. However, its main weakness is the lack of detail. After the team identifies 10 features in the Should-have category, MoSCoW provides no tools for prioritizing them against each other. That is why it works most effectively in tandem with a quantitative method like RICE or WSJF. MoSCoW helps to determine «what we do», and RICE/WSJF determines

«in what order». Using MoSCoW in isolation can lead to a situation where the team executes tasks effectively but not in the optimal sequence.

The comparative analysis of four different prioritization methods clearly demonstrates that the choice of a suitable framework is not merely a technical task but a profoundly strategic one. The results show that there is no single «correct» way to rank a backlog. Each method is, in essence, a lens through which the team views its product and market. ICE and RICE focus on user value, WSJF on economic efficiency, and MoSCoW on defining the scope of work. Thus, the choice of framework implicitly determines which «wins» the company seeks to achieve first: those that are quick and noticeable to users, those that are most economically advantageous, or those that are strategically most important for a specific release. This transforms the prioritization process from a routine exercise into an act of shaping product strategy.

The pair of frameworks, ICE and RICE, represents a classic trade-off between speed and objectivity. ICE, due to its extreme simplicity, is an excellent tool for teams just beginning to implement systematic prioritization. However, its main drawback is high subjectivity. RICE attempts to solve this problem by adding the «reach» criterion, which forces a reliance on data. Yet, it remains vulnerable to undervaluing features for strategically important but small user segments. In contrast, WSJF is a significantly more complex but also a strategically deeper tool that links development with product economics. MoSCoW, meanwhile, serves as an excellent tool for communication and scope management but lacks granularity within its categories.

The most interesting finding of the research is that none of the theoretical models coincided with the final roadmap that the team actually worked with. This demonstrates a fundamental truth of product management: frameworks are tools to support decision-making, not immutable laws. The real business context, strategic imperatives, and unforeseen market opportunities always introduce their own adjustments. The roadmap generated by the ICE method suggested starting

with features that had the maximum impact on users with minimum effort. Nevertheless, in reality, this roadmap underwent significant changes (Table 5).

Table 5

The actual order of feature implementation and its justification

№	Feature Name	Justification for the Change in Implementation Order
1	Referral Page	The need to ensure sales growth and attract new clients.
2	Task Management System Redesign	The need to provide clients with a better interface and understanding of the task workflow.
3	Customer Tipping for Talents	The desire to create additional income and increase satisfaction for talents.
4	Automated Matching Algorithm	The aspiration to significantly reduce the team's workload and the amount of manual work.
5	Task Rematch Functionality	The need to increase client satisfaction and retention levels.
6	Payments Page Redesign	The need to give clients transparency and control over subscriptions, which increases trust.
7	Automated Talent Hiring Flow	The goal was to reduce the time to hire talents several-fold for rapid scaling.
8	Customer NPS Integration	The need to more easily track client satisfaction and improve retention.
9	Voice & Video Comments in Tasks	The desire to simplify and speed up communication on the platform for both parties.
10	Talent Task Approval Workflow	The goal was to accelerate the process of matching performers to tasks several-fold.
11	Talents Mobile Application	The need to speed up matching and responses to clients via mobile notifications.
12	Calendar & Calls Integration	The aspiration to save the Customer Success team's time by automating call scheduling.
13	Customer Success Process Automation	The goal was to save the Customer Success team's time for work on higher-priority tasks.
14	Talent Profile & Portfolio Pages	The desire to increase clients' trust and confidence in the quality of talents on the platform.
15	Talent App Redesign	Improving talent satisfaction, although it was not the highest priority.
16	Technical Debt Resolution	Was de-prioritized but completed to support the long-term stability of the system.

An analysis of the actual sequence shows how strategic business needs can override mathematically calculated priorities. For example, «Referral Page», which ranked only 5th according to ICE, was implemented first. The justification – «the need to ensure sales growth» – is a typical example of a business decision that goes beyond the standard criteria of the framework. Similarly, «Task

Rematch Functionality», the absolute leader in all theoretical rankings, was implemented only fifth because a certain foundation needed to be laid before its introduction (e.g., improving the task management system), which is also a factor that the frameworks do not account for.

This gap between theory and practice does not diminish the value of the frameworks. On the contrary, it underscores their role – to be a starting point for a deep, well-reasoned discussion. The results obtained using ICE or RICE are not a command to act, but a structured set of data that the team uses in negotiations with stakeholders. They allow for an objective assessment of what must be sacrificed when the business decides to change the order in favor of immediate commercial benefit. Without such an analysis, priority decisions would be made purely on intuition, which significantly increases the risk of errors. This is precisely why startups in the early stages of development most often choose simpler methods, such as ICE. At this stage, the speed of learning and the ability to react flexibly to market feedback are much more valuable than the precision of priorities. The cognitive load on the team from complex frameworks like WSJF is too high. A simple, «good enough» prioritization that allows for rapid movement while leaving room for strategic maneuvers is the optimal strategy for their survival and growth. The framework becomes the foundation upon which the product manager relies, always reserving the right to change the order upon seeing new opportunities or threats.

Conclusions. The conducted research confirms that the choice of a framework for feature prioritization in a startup context is not a tactical, but a fundamental strategic action. The empirical analysis based on a practical case study demonstrated that each of the considered methods – ICE, RICE, WSJF, and MoSCoW – shapes its own product development vector, emphasizing different aspects: from the speed of iterations and user reach to economic efficiency and scope management discipline. The key finding of the research is the identification of a significant discrepancy between the theoretically optimal priorities,

calculated using the models, and the actual sequence of implementation. This discrepancy indicates that, in practice, decisions are largely determined by external business imperatives, such as the need for immediate sales growth or the presence of technical dependencies, which are not always accounted for in formalized calculations.

Accordingly, the main value of prioritization frameworks lies not in creating an immutable and solely correct plan of action, but in their function as a tool for structuring thought and stimulating informed communication. They provide a transparent, data-based starting point for a strategic dialogue between the product team and other stakeholders. Using these methodologies allows for an objective assessment of the potential losses and trade-offs the company makes when consciously deviating from the mathematically optimal path in favor of urgent business needs. Thus, the role of the product manager is not to blindly follow the dictates of a model, but to skillfully balance its recommendations with the dynamic and often unpredictable context of the market environment.

Based on the analysis, a situational approach to selecting tools can be recommended, depending on the startup's maturity stage. For early-stage companies, it is advisable to use simpler models (ICE, MoSCoW) that prioritize the speed of learning over precision. As the company grows and accumulates data, a transition to more objective frameworks (RICE) is appropriate, and at the scaling stage, to economically oriented ones (WSJF). Promising directions for further research include the development of hybrid models that formally integrate qualitative strategic factors into quantitative calculations, as well as conducting longitudinal studies that examine the correlation between the choice of framework in the early stages and the company's long-term success metrics.

References

1. Pattyn F. Enhancing Startup Success Rates: Towards a Pragmatic Framework for Product Managers (PFPM). *2023 IEEE 31st International Requirements Engineering Conference (RE)*, Hannover, Germany, 4–8 September 2023. 2023. URL: <https://doi.org/10.1109/re57278.2023.00059> (date of access: 16.09.2025).
2. Identification and Prioritization of Challenges Faced by Village Organizations in Meghalaya: A MoSCoW Prioritization Approach / T. Akhtar et al. *Journal of Community Mobilization and Sustainable Development*. 2025. Vol. 20, no. 3. P. 767–772.
3. Kostev R. S. Challenges and Problems of the MoSCoW Method Application in ERP System Implementation. *2023 International Scientific Conference on Computer Science (COMSCI)*. 2023. P. 1–4.
4. RECTV: Enhancing Agile Decision-Making in Startups through Value-Driven Requirements Prioritization / F. Pattyn et al. *IEEE Access*. 2025. URL: <https://doi.org/10.1109/ACCESS.2025.3589942> (date of access: 18.09.2025).
5. Ellis S., Brown M. *Hacking Growth*. Penguin Random House USA Ex, 2017. 390 p.
6. Collins D. ICE Scoring | ProdPad. *ProdPad*. URL: <https://www.prodpad.com/glossary/ice-scoring> (date of access: 18.09.2025).
7. ICE Scoring Model. *Learning Loop*. URL: <https://learningloop.io/glossary/ice-scoring-model> (date of access: 18.09.2025).
8. Капліна А. І. Прийняття ризикованих рішень в умовах невизначеності. *Agrosvit*. 2024. № 22. С. 94–97. URL: <https://doi.org/10.32702/2306-6792.2024.22.94> (дата звернення: 18.09.2025).
9. RICE Scoring Model. *ProductPlan*. URL: <https://www.productplan.com/glossary/rice-scoring-model/> (date of access: 18.09.2025).

10. Reinertsen D. The Principles of Product Development Flow. Celeritas Publishing, 2009. 304 p.

11. An Overview of Cost-of-Delay Analysis:- Calculating Project Decision Rules / D. G. Reinertsen et al. *The Journal of Cost Analysis & Management*. 2002. Vol. 4, no. 1. P. 8–24. URL: <https://doi.org/10.1080/15411656.2002.10462236> (date of access: 18.09.2025).