



DOI 10.28925/2663-4023.2026.32.1119

УДК 004.4:005.6:378

Кучаковська Галина Андріївна

кандидат педагогічних наук, старший викладач кафедри комп'ютерних наук

Київський столичний університет імені Бориса Грінченка, Київ, Україна

ORCID: 0000-0002-4555-896X

h.kuchakovska@kubg.edu.ua

Хорольська Карина Вікторівна

PhD in Computer Science, доцент кафедри інформаційної та кібернетичної безпеки імені професора

Володимира Бурячка

Київський столичний університет імені Бориса Грінченка, Київ, Україна

ORCID:0000-0003-3270-4494

k.khorolska@kubg.edu.ua

МОДЕЛІ RICE ТА ICE ЯК ІНСТРУМЕНТИ ФОРМАЛІЗАЦІЇ ПРІОРИТИЗАЦІЇ ФУНКЦІОНАЛУ ІТ-ПРОДУКТУ В ПІДГОТОВЦІ МАЙБУТНІХ ІТ-ФАХІВЦІВ

Анотація. Актуальність дослідження зумовлена зростанням ролі продуктових рішень у розробці ІТ-продуктів та потребою формування у майбутніх ІТ-фахівців навичок обґрунтованої пріоритизації функціоналу в умовах обмежених ресурсів і високої невизначеності. Недостатня формалізація цього процесу в освітній підготовці призводить до домінування інтуїтивних рішень, що не відповідають сучасним вимогам Agile- та data-driven-підходів. Метою статті є, шляхом порівняльного аналізу та емпіричної перевірки моделей ICE та RICE, дослідити їх ефективність у формалізації процесу пріоритизації функціоналу ІТ-продуктів на ранніх стадіях життєвого циклу та визначити педагогічну доцільність їх застосування у підготовці майбутніх ІТ-фахівців. Методологія дослідження ґрунтується на емпіричному аналізі результатів виконання лабораторної роботи здобувачами вищої освіти третього курсу спеціальності 122 «Комп'ютерні науки» в межах дисципліни «Підприємництво та Start-up». У дослідженні застосовано методи порівняльного аналізу, формалізації та узагальнення результатів пріоритизації функціоналу з використанням моделей ICE та RICE у середовищі Huygger. Отримані результати засвідчили, що модель ICE є ефективним інструментом первинного відбору ідей на етапі концептуалізації продукту, тоді як модель RICE забезпечує більш об'єктивну та аналітично обґрунтовану оцінку пріоритетів за наявності кількісних даних. Такий підхід відображає зміщення акценту в ІТ-освіті від традиційного предметного навчання до проектно-орієнтованої освіти, у межах якої здобувачі опановують навички планування, командної взаємодії та оцінки ефективності розробки. Таким чином, результати дослідження демонструють, що інтеграція ICE та RICE у освітній процес сприяє підготовці конкурентоспроможних ІТ-фахівців, здатних діяти в парадигмі data-driven product development і відповідати вимогам індустрії 4.0.

Ключові слова: пріоритизація функціоналу; RICE; ICE; Product Management; Agile; розробка ІТ-продуктів; стартапи; методи оцінювання features.

ВСТУП

У сучасному світі інформаційних технологій розвиток ІТ-продуктів вимагає не лише технічної компетентності, а й уміння ефективно керувати ресурсами на етапах планування та пріоритизації функціоналу. Моделі RICE (Reach×Impact×Confidence/Effort) та ICE (Impact×Confidence×Ease) є потужними інструментами формалізації цього процесу, перетворюючи суб'єктивні рішення на об'єктивні розрахунки, що базуються на кількісних метриках. У контексті підготовки майбутніх ІТ-фахівців ці моделі набувають особливого значення, дозволяючи



формувати навички продуктового мислення поряд з програмуванням, як підкреслюють сучасні освітні підходи.

За даними дослідження CB Insights [1], близько 42% стартапів зазнають невдачі через відсутність ринкової потреби у продукті, а 29% – через вичерпання фінансових ресурсів до досягнення product-market fit (відповідність продукту ринку). Однією з ключових причин цих невдач є неефективна пріоритизація функціоналу на етапі розробки продукту [2].

У контексті обмежених ресурсів – часу, бюджету, кваліфікованих спеціалістів – команди розробників стартапів та менеджери продукту стикаються з критичним завданням: які фічі розробляти першочерговими, щоб максимізувати цінність продукту для користувачів та забезпечити життєздатність бізнес-моделі? Класичний підхід «розробити все, що хочуть стейкхолдери» виявляється нежиттєздатним у реаліях сучасних Agile-методологій [3], де цінність вимірюється не кількістю функцій, а здатністю продукту вирішувати реальні проблеми користувачів. Проблема пріоритизації набуває особливої ваги в екосистемі стартапів. За методологією Lean Startup [4], критично важливим є швидке тестування гіпотез та ітеративне вдосконалення продукту на основі зворотного зв'язку від користувачів. Однак без структурованого підходу до визначення пріоритетів команди ризикують витратити ресурси на розробку функцій з низьким впливом на ключові метрики, втрачаючи конкурентні переваги та можливості для масштабування [5].

Дослідження ProductPlan [6] показує, що лише 34% менеджерів продукту використовують формалізовані методи пріоритизації, натомість 51% покладаються на інтуїцію та думку найвищого керівництва (феномен HiPPO – Highest Paid Person's Opinion). Така ситуація призводить до систематичних помилок у плануванні розробки, переоцінки легко реалізованих фіч та недооцінки функцій з високим потенціалом впливу на користувацький досвід [7].

У навчальних програмах українських ЗВО, зокрема на спеціальностях 122 «Комп'ютерні науки» та 123 «Комп'ютерна інженерія», здобувачі вищої освіти отримують ґрунтовні знання з програмування, алгоритмів, архітектури програмних систем. Водночас аспекти розробки власного програмного продукту, створення та розвитку стартапу розглядається в рамках вивчення дисципліни «Підприємництво та Стартапи» та «Управління ІТ проектами», але часто залишаються поза увагою або викладаються фрагментарно саме питання пріоритизації функціоналу, без систематичного практичного застосування. Наукова новизна статті полягає у зміщенні акценту з технічного опису моделей ICE та RICE на педагогічну апробацію їх потенціалу як засобів навчання мисленню продуктового аналітика, здатного формалізувати процес ухвалення рішень у контексті реальної ІТ-розробки.

Мета статті. Ця стаття має на меті шляхом порівняльного аналізу та емпіричної перевірки моделей ICE та RICE, дослідити їх ефективність у формалізації процесу пріоритизації функціоналу ІТ-продуктів на ранніх стадіях життєвого циклу та визначити педагогічну доцільність їх застосування у підготовці майбутніх ІТ-фахівців.

Для досягнення мети дослідження поставлено такі завдання:

- проаналізувати теоретичні основи еволюцію класичних підходів до пріоритизації (MoSCoW, Kano Model, Value vs. Effort, WSJF) та визначити їх обмеження у сучасних умовах цифрової розробки;
- охарактеризувати зміст і аналітичні особливості моделей ICE та RICE, окреслити їх структуру, параметри оцінювання та алгоритми застосування;



- провести аналіз ефективності моделей ICE та RICE на прикладі їх використання у середовищі Nuygger, проаналізувавши результати ранжування функціоналу;
- визначити навчальні результати, що формуються у здобувачів вищої освіти під час використання моделей ICE та RICE у процесі розробки ІТ-продуктів, і встановити способи їх оцінювання;
- обґрунтувати педагогічну доцільність впровадження моделей ICE та RICE у процес підготовки ІТ-фахівців як інструментів формування аналітичного, продуктового та управлінського мислення.

Постановка проблеми. Попереднє опитування здобувачів вищої освіти третього курсу (N=47) показало, що 89% респондентів не знайомі з методами RICE та ICE, а 76% визнають, що у своїх курсових та навчальних проєктах вони визначають пріоритети функцій інтуїтивно, без використання структурованих підходів. Це створює розрив між академічною підготовкою та реальними потребами індустрії, де вміння обґрунтовано приймати рішення про пріоритизацію є критичною компетенцією Product Owner (власник ІТ-продукту), Scrum Master (фасилітатор, який допомагає команді ефективно використовувати методологію Scrum, щоб створювати цінний продукт) та інших ролей у Agile-командах.

Аналіз останніх досліджень і публікацій. Існує проблема відсутності чітких рекомендацій щодо вибору методу пріоритизації. Спеціалізована література [8; 9] описує різноманітні фреймворки – MoSCoW, Kano Model, Value vs. Effort Matrix, RICE, ICE, WSJF – однак не надають систематизованих критеріїв для прийняття рішення, який з методів застосовувати в конкретному контексті. Для новачків у продуктовому менеджменті, зокрема здобувачів вищої освіти ІТ-спеціальностей, це створює когнітивне навантаження та паралізує процес прийняття рішень через надлишок інформації.

Особливо гострою є проблема адаптації методів пріоритизації до різних стадій життєвого циклу ІТ-продукту. Метод RICE, розроблений командою Intercom [10] для зрілих продуктів з великою базою користувачів та розвиненою аналітикою, може виявитися надто складним та даних-інтенсивним для MVP-стадії стартапу. З іншого боку, простіший метод ICE [11] може не враховувати критично важливі фактори масштабу впливу (Reach) для продуктів у фазі росту. Відсутність формалізованого підходу до вибору між цими методами створює ризики неоптимальних рішень у пріоритизації на ранніх етапах розвитку ІТ-продукту. Управління беклогом продукту є центральним елементом методологій Agile та Scrum, де пріоритизація фіч визначає ефективність ітеративної розробки. Традиційні підходи еволюціонували від якісних до кількісних моделей, відображаючи зростання складності ІТ-продуктів у стартапах та зрілих компаніях [12].

Під час дослідження Н. Таснім [13] на основі систематичного огляду 87 наукових публікацій було виявлено, що пріоритизація вимог є однією з найскладніших активностей у Agile-розробці, особливо в контексті розподілених команд та невизначеності вимог на ранніх стадіях проєкту. Автори ідентифікували 23 різні методи пріоритизації, що використовуються в індустрії, і констатували відсутність консенсусу щодо того, який метод є оптимальним для конкретних типів проєктів.

Огляд літератури охоплює еволюцію підходів до управління беклогом продукту в рамках Agile та Scrum. Класичні методи, такі як Kanban та MoSCoW (Must-have, Should-have, Could-have, Won't-have), фокусуються на категоризації завдань, але не враховують кількісних метрик впливу. У Scrum backlog refinement передбачає оцінку в story points, однак бракує формалізованих формул для пріоритизації [14].



ВИКЛАДЕННЯ ОСНОВНОГО МАТЕРІАЛУ

Класичні методи, такі як MoSCoW (Must-have, Should-have, Could-have, Won't-have), базуються на категоризації вимог за критичністю, але страждають від суб'єктивності оцінок власника ІТ-продукту. Модель Капо класифікує фічі за впливом на задоволеність користувача (базові, продуктивні, захватні), фокусуючись на досвіді користувача, проте не враховує ресурси чи масштаб впливу. Матриця Value vs Effort відображає співвідношення цінності фічі до зусиль на реалізацію, використовуючи чотири квадранти (Quick Wins, Big Bets тощо), але залишається напівкількісною через довільні шкали. [12; 15]. Ці методи домінували в ранніх етапах Agile (2000-і роки), коли команди були малими, але з ростом продуктів виявили обмеження: відсутність кількісних метрик призводить до перевантаження беклогу та затримок спринтів [16].

Перехід до кількісних підходів відбувся у 2010-х роках завдяки growth hacking (злам зростання) та data-driven product management (управління продуктом на основі даних). Модель ICE (Impact, Confidence, Ease), запропонована Шоном Еллісом у 2014 році, призначена для швидкої пріоритизації growth-ініціатив. Вона оцінює вплив (0-10), впевненість у прогнозі (%) та легкість реалізації (0-10), дозволяючи ранжувати ідеї без глибоких даних [17; 18].

Формула $ICE = \frac{Impact * Confidence}{Ease}$, де Impact відображає вплив на ключові метрики (KPI), Confidence – впевненість у прогнозі, Ease – легкість реалізації, дозволяє швидко оцінювати ідеї за шкалою 1-10:

– «Близько нуля» – такої впевненості ми даємо від 0,01 до 0,1 бала. Це особиста впевненість у фічі, галузеві тренди, зовнішні дослідження.

– «Дуже низька» – від 0,1 до 0,5 бала. Сюди належать думки команди, інвесторів, зовнішніх експертів. Грубі підрахунки та прогнози, створені продакт-менеджером.

– «Низька» – від 0,5 до 1 бала. Це кілька зацікавлених клієнтів, наявність фічі у конкурента.

– «Трохи вище за низьку» – від 1 до 3 балів. Підтримують оцінку такі фактори: дослідження ринку, смоук-тестування (smoke tests), фіча є у всіх конкурентів. Для оцінки «3» потрібні реальні дані про продукт, інтерв'ю з 20+ користувачами, дослідження юзабіліті, MVP.

– «Підвищена» – від 3 до 7 балів. Це впевненість, підкріплена довгими дослідженнями користувача, масштабований MVP, A/B-тестами.

– «Висока» – від 7 до 10 балів. Цей рівень впевненості можливий, якщо продакт-менеджер вже тестував таку функціональність на цьому продукті.

Дослідження Productboard 2024 [19] показують, що 45% продакт-менеджерів використовують ICE на ранніх стадіях продукту через простоту.

Модель RICE, розроблена Intercom у 2014 році, розширює ICE, додаючи врахування масштабу охоплення (Reach) та нормалізація через зусилля (Effort у знаменнику), що дозволяє порівнювати фічі різної складності.):

$RICE = \frac{Reach * Impact * Confidence}{Effort}$. Ця модель орієнтована на consumer-продукти з доступними аналітикою (GA, Mixpanel), де Reach забезпечує об'єктивність.

Як зазначається у матеріалах Product School [20], метод RICE забезпечує об'єктивний, data-driven підхід до прийняття рішень, хоча й вимагає наявності надійних аналітичних даних. Карлос Гонсалес де Вільяумбросія Founder & CEO в Product School в своєму дослідженні підкреслює [21], що RICE особливо ефективний для команд з розвиненою продуктовою аналітикою та понад 10,000 активних користувачів, що



дозволяє надійно розрахувати компоненту Reach. Метод набув широкого поширення і за звітним ProdPlan «The 2025 State of Product Management Report» 2025 року став 5-м найпопулярнішим фреймворком пріоритизації серед продакт-менеджерів [22]. Віталій Царьов [23] наголошує, що ефективна пріоритизація завдань є критично важливою навичкою для менеджерів продуктів та проєктів. Основним недоліком використання фреймворку (методу) RICE визначає суб'єктивізація його алгоритму розрахунку, а також нехтування факторами, які можуть суттєво впливати на його значення.

Сучасні дослідження підкреслюють інтеграцію метрик у вибір методології розробки, але неформалізованість переходу між ICE/RICE. Критика включає суб'єктивність оцінок та ігнорування невизначеності в стартапах. Критичний аналіз обох методів представлено у матеріалах дослідження Вікторії Яценко [24], де вказують на ризик «scoring theater» – маніпуляції з цифрами для обґрунтування заздалегідь прийнятих рішень, а також на небезпеку надмірного фокусу на легко вимірюваних метриках на шкоду стратегічним ініціативам.

В освітньому контексті навчання майбутніх ІТ-спеціалістів важливо не просто ознайомити здобувачів вищої освіти із загальними принципами Agile-управління, а й сформувані у них вміння приймати рішення на основі кількісних критеріїв, аналізу даних і логіки бізнес-пріоритетів. Тож для покращення процесу вибору методу пріоритизації варто їх порівняти. Порівняльний аналіз методів MoSCoW, Kano Model, Value vs. Effort Matrix, WSJF, ICE і RICE дає змогу виявити не лише їхні аналітичні й практичні особливості, а й освітню доцільність у підготовці майбутніх ІТ-фахівців. Порівняльна методів пріоритизації функціоналу ІТ-продукту (MoSCoW, Kano, Value vs. Effort, WSJF, ICE, RICE), що подана в Таблиці 1 була побудована на основі системи критеріїв, релевантних процесу навчання майбутніх ІТ-фахівців, а також практичним етапам розробки ІТ-продукту у межах навчальних проєктів, лабораторних курсів чи стартап-інкубаційних програм.

Критерій «Тип оцінювання» описує, на якій логіці ґрунтується прийняття рішень у межах конкретного методу: якісній (експертній, категорійній) чи кількісній (формульній, обчислювальній). У контексті підготовки ІТ-фахівців цей параметр є важливим, оскільки формує у здобувачів вищої освіти навички аналітичного мислення. Методи, що мають чітку кількісну структуру (ICE, RICE, WSJF), дозволяють навчити майбутніх спеціалістів обґрунтовувати вибір функціоналу не інтуїтивно, а за числовими показниками, що наближає освітній процес до реалій ІТ-індустрії.

Критерій «Кількісна формалізація» визначає, наскільки обраний метод передбачає використання числових параметрів і математичних моделей для визначення пріоритетів. Для освітнього процесу цей аспект має особливе значення, адже дозволяє здобувачам вищої освіти освоїти елементи продуктової аналітики, системного мислення та data-driven підходу, що є ключовими у сучасному Product Management.

Рівень когнітивного навантаження та зрозумілості алгоритму для користувача оцінюється в межах характеристики критерію «Складність застосування». У педагогічному контексті важливо, щоб метод був доступний для засвоєння здобувачами вищої освіти з різним рівнем підготовки. Наприклад, MoSCoW і Value vs. Effort є інтуїтивними й корисними для початкового етапу, тоді як RICE і WSJF вимагають глибшого розуміння аналітики.

Виокремлення критерію «Необхідність емпіричних даних» ґрунтується на тому, що важливим є визначити, яку кількість вихідних даних потрібно мати для застосування обраного методу пріоритизації (результати опитувань, аналітику використання продукту, часові чи фінансові показники). У освітньому процесі це



важливо, оскільки здобувачі вищої освіти часто працюють із гіпотетичними або обмеженими даними.

Критерій «Фокус методу» показує, на який аспект розробки або управління продуктом орієнтований метод – користувацький досвід (Kano), бізнес-цінність (WSJF), внутрішня ефективність команди (ICE), чи стратегічна масштабність (RICE). У підготовці майбутніх IT-фахівців важливо показати взаємозв'язок між технічними рішеннями та цінністю для користувача/бізнесу. Так, метод ICE дозволяє розуміти вплив функціоналу на розвиток продукту, тоді як RICE допомагає мислити у категоріях ринку, охоплення та рентабельності.

І так як, кожен метод має свої педагогічні й практичні сильні та слабкі сторони. У таблиці вони подані для того, щоб допомогти суб'єктам освітнього процесу усвідомлено обирати підхід залежно від рівня навчання (бакалаврський чи магістерський курс), обсягу доступних даних і мети навчального завдання.

Метод WSJF, хоч і є точним з економічної точки зору, вимагає оперування складними фінансовими метриками (Cost of Delay), що робить його менш доступним для здобувачів вищої освіти початкового рівня підготовки.

Натомість ICE та RICE забезпечують оптимальний баланс аналітичності, простоти та універсальності. ICE дозволяє швидко порівняти потенційні функції за трьома критеріями – впливом, впевненістю та легкістю реалізації – і є ефективним інструментом формування критичного мислення при прийнятті рішень. RICE додає вимір Reach, що стимулює мислити у категоріях користувацької масштабності, бізнес-цінності та операційної ефективності.

Таблиця 1

Порівняльна характеристика методів пріоритизації функціоналу IT-продукту*

Метод	Тип оцінювання	Кількісна формалізація	Складність застосування	Необхідність емпіричних даних	Фокус методу	Основні переваги	Ключові недоліки
MoS-CoW	Категорійний (Must, Should, Could, Won't)	✘	Низька	Низька	Управління вимогами	Простота, зручність для планування	Відсутність кількісної обґрунтованості, суб'єктивність
Kano Model	Користувацько-поведінковий	Частково	Середня	Висока	Орієнтація на задоволення користувача	Враховує емоційні фактори користувачів	Потребує глибоких UX-досліджень, складний у реалізації
Value vs. Effort Matrix	Візуальний/якісний	Частково	Низька	Середня	Баланс цінності й витрат	Простота сприйняття, швидкість оцінки	Не враховує впевненість у результатах і масштаб впливу
WSJF (Weighted Shortest Job First)	Кількісний (Cost of Delay/Duration)	✓	Висока	Висока	Lean Portfolio Management	Формалізована формула, бізнес-орієнтація	Потребує бізнес-метрик, важкий для початківців



Продовження таблиці 1

ICE	Кількісний (Impact × Confidence × Ease)	✓	Низька/ середня	Середня	Growth, стартапи, MVP	Простота формули, гнучкість, швидке порівняння гіпотез	Не враховує масштаб користувацької бази
RICE	Кількісний (Reach × Impact × Confidence)/Effort)	✓	Середня	Середня / висока	Стратегічне планування, Product Roadmap	Враховує масштаб користувачів, впевненість та витрати	Потребує більше даних і розрахунків

*Джерело: Складено авторами.

Таким чином, у контексті освітнього процесу підготовки майбутніх ІТ-фахівців методи ICE та RICE сприяють розвитку компетентностей у data-driven product management, аналітичному мисленні, оцінці ринкових гіпотез та раціональному використанні ресурсів. Їх застосування в освітньому середовищі дає змогу перейти від теоретичного розуміння Agile-принципів до практичної навички формалізованої пріоритизації функціоналу на основі даних.

Аналіз можливостей використання методів ICE та RICE для ефективною пріоритизації функціоналу ІТ-продуктів на початковій стадії життєвого циклу, зосереджуючись на адаптації цих методів до проєктів, а також аналіз того, як застосування ICE та RICE сприяє формуванню структурованого підходу до прийняття продуктових рішень, обґрунтованого розподілу ресурсів і вибору пріоритетів у процесі розробки ІТ-продуктів.

Опишемо ілюстративний приклад застосування моделей ICE та RICE в процесі викладання навчальної дисципліни «Підприємництво та Start-Up» здобувачам вищої освіти 3 курсу спеціальності 122 «Комп'ютерні науки» та 123 «Комп'ютерна інженерія» Київського столичного університету імені Бориса Грінченка. Дисципліна обсягом 270 годин (9 кредитів), що викладалася протягом трьох семестрів, була спроектована на основі Всеукраїнського інтерактивного курсу «Створення та розвиток ІТ-продуктів» та інтегрована в освітній компонент відповідно до чинної акредитації. У попередніх дослідженнях авторів [25] уже було описано етапи формування Lean Canvas власного продукту, який передував практичному опануванню методів пріоритизації та базувався на засвоєнні понять цінності продукту, retention (ретеншн, стримання), product-market fit (PMF – маркетинговий показник, який відображає ступінь відповідності реалізованого компанією продукту інтересам цільової аудиторії), стартап-екосистеми, продуктової стратегії, аналізу цільової аудиторії та визначення MVP.

Ключовим емпіричним елементом дослідження стала лабораторна робота «Оцінка пріоритетів. Як використовувати RICE та ICE Scoring», виконання якої було спрямоване на формалізацію процесу прийняття продуктових рішень у межах ІТ-проєктів. Загальний обсяг роботи над лабораторною становив 20 годин і охоплював повний цикл: від виділення функціоналу власного ІТ-продукту до порівняльного аналізу результатів пріоритизації за методами RICE та ICE з використанням спеціалізованого програмного забезпечення для управління ІТ-проєктами Hygger.



У межах лабораторної роботи здобувачі вищої освіти, відповідно до поставлених навчальних цілей, виділяють не менше шести функцій обраного або запропонованого стартап-проєкту та здійснювали їх оцінювання за двома методиками. Метод RICE застосовувався як інструмент кількісної формалізації пріоритетів на основі чотирьох параметрів: охоплення (Reach), впливу на цінність продукту (Impact), впевненості в оцінці (Confidence) та трудовитрат (Effort). Особливий акцент робився на використанні реалістичних метрик охоплення, зокрема показників Feature Discovery (відсоток людей, які знайшли та скористалися фічею у продукті.), а також на усвідомленому виборі рівнів впливу та впевненості з урахуванням наявних даних, досліджень користувачів або гіпотез, сформованих у межах MVP.

Метод ICE, у свою чергу, використовувався як спрощена модель пріоритизації, що дозволяла здобувачам вищої освіти швидше оцінити ідеї за критеріями впливу (Impact), впевненості (Confidence) та легкості реалізації (Ease). Практична реалізація ICE дала змогу виявити, що здобувачі вищої освіти схильні на початкових етапах проєктування продукту переоцінювати вплив функцій і недооцінювати складність їх реалізації, що стало важливим підґрунтям для подальшого обговорення обмежень інтуїтивних підходів до пріоритизації. Використання інструменту Hugging забезпечило стандартизоване середовище для реалізації обох методів. Здобувачі вищої освіти створювали проєкти, працювали з дошками за шаблонами ICE та RICE, оцінювали фічі в табличному режимі та візуалізували результати пріоритизації. Порівняння підсумкових рейтингів функцій показало, що результати ICE та RICE не завжди збігаються: функції з високим суб'єктивним впливом і простотою реалізації за ICE часто знижували свій пріоритет у RICE через значні трудовитрати або низьку впевненість в оцінках.

Розглянемо етапи пріоритизації за методами RICE та ICE з використанням спеціалізованого програмного забезпечення для управління IT-проєктами Hugging на прикладі ідеї однієї з команд – Planify - «Розумний календар, який планує за тебе».

Перший етап передбачав реєстрацію в середовищі Hugging з використанням корпоративного акаунта та створення окремого проєкту для оцінки пріоритетів. Здобувачі вищої освіти працювали з дошкою, сформованою на основі шаблону ICE, що дозволяло уніфікувати підхід до подальшого оцінювання. У межах цього етапу всі попередньо визначені функції додавалися до проєкту в режимі дошки, що сприяло візуальному сприйняттю обсягу функціоналу та його складності.

На другому етапі здійснювався вибір власного стартап-проєкту або формалізація наявної ідеї IT-продукту, після чого виділяли перелік ключових його функцій (не менше шести), що потенційно формують цінність MVP. Були визначені наступні фічі:

- F1 Базовий календар (ручні події, перегляд день/тиждень);
- F2 Інтеграція Google Calendar (синхронізація подій);
- F3 Інтеграція Outlook/Microsoft 365 (синхронізація подій);
- F4 Telegram/месенджери: AI-парсинг подій із чатів;
- F5 LMS/Email: AI-парсинг дедлайнів (університет/пошта);
- F6 Smart Reminders: розумні нагадування + автооновлення змін;
- F7 Smart Scheduler: конфлікти + пропозиція альтернативних слотів;
- F8 Безпека та контроль доступів (мінімізація доступів, аудит тощо).

Даний етап ілюструється створення списку фіч у середовищі Hugging, які відображають перехід від концептуального опису продукту до структурованого функціонального представлення.

На третьому етапі здійснювалося оцінювання фіч за методом ICE шляхом послідовного визначення показників Impact, Confidence та Ease в табличному режимі Hygger та ранжування кожної фічі (рис. 1).

Як бачимо найвищий пріоритет (Score = 392) отримала функція «Базовий календар (ручні події, дні/тиждень)», що свідчить про її високий потенційний вплив на користувацький досвід та відносно просту реалізацію. Ця фіча перебуває у стадії розроблення (*In Development*), що узгоджується з її стратегічною важливістю.

Наступні за пріоритетом позиції – «Smart Reminders (розумні нагадування)» та «Інтеграція Google Calendar (синхронізація)» із показником 336. Обидві функції мають подібний рівень оцінок Impact (8) і Confidence (7-8), що свідчить про стабільно високий рівень користувацької цінності та впевненості команди у доцільності їх реалізації.

Task	Start Date	Due Date	Column	Impact	Confidence	Ease	Score
#1 Базовий календар (ручні події, дні/тиждень)			In Development (In progress)	7	8	7	392
#2 Smart Reminders (розумні нагадування)			Features to Score	8	7	6	336
#3 Інтеграція Google Calendar (синхронізація)			Scored Features	8	7	6	336
#4 LMS/Email: AI-парсинг дедлайнів			Done	9	6	4	216
#5 Інтеграція Outlook/Microsoft 365 (синхронізація)			In Development (In progress)	6	6	5	180
#6 Безпека та контроль доступів			Features to Score	6	7	4	168
#7 Smart Scheduler (конфлікти + слоти)			Features to Score	7	5	4	140
#8 Telegram: AI-парсинг подій із чатів			In Development (Done)	9	5	3	135

Рис. 1. Результати ранжування кожної фічі за методом ICE у табличному режимі у середовищі Hygger

Середньопріоритетні фічі, такі як «LMS/Email: AI-парсинг дедлайнів» (216) та «Інтеграція Outlook/Microsoft 365» (180), демонструють помірний вплив, але потребують більших зусиль для впровадження, що знижує їхній загальний показник ICE. Це типова ситуація, коли важливі функції з високою технічною складністю відкладаються до моменту завершення базового функціоналу.

Найнижчі значення отримали «Smart Scheduler (конфлікти з слотами)» (140) та «Telegram: AI-парсинг подій із чатів» (135). Незважаючи на потенційну інноваційність, низькі оцінки Ease і Confidence вказують на високий рівень технічної невизначеності або обмежений очікуваний ефект для користувача. Це демонструє коректність моделі ICE у виявленні фіч із сумнівною окупністю зусиль.

Аналіз робіт здобувачів вищої освіти показав, що на цьому етапі домінує інтуїтивний підхід до оцінки впливу та легкості реалізації, що є характерним для початкових стадій продуктового мислення. Водночас саме цей етап дозволив зафіксувати типові помилки в оцінюванні та створив підґрунтя для подальшої корекції підходів.

Наступний етап лабораторної роботи полягав у створенні дошки за шаблоном RICE та повторному оцінюванні того самого набору фіч із використанням параметрів Reach, Impact, Confidence та Effort (рис. 2).

Task ID	Task Name	Start Date	Due Date	Column	Reach	Impact	Confide	Score
#1	Базовий календар			Features to Score	1000	2	8	213
#2	Інтеграція Google Calendar			Next Iteration	800	2	7	187
#3	Smart Reminders	Nov 20, 2017 at 2:20 PM		Next Iteration	900	2	6	167
#4	Інтеграція Outlook/Microsoft 365			In Development (Done)	600	2	6	103
#5	LMS/Email: AI-парсинг дедлайнів			Features to Score	650	3	5	89
#6	Безпека та контроль доступів			Done	1000	1	7	78
#7	Smart Scheduler			Scored Features	750	2	5	75
#8	Telegram: AI-парсинг подій із чатів			Scored Features	700	3	5	75

Рис. 2. Результати ранжування кожної фічі за методом RICE у табличному режимі у середовищі Hygger

Особлива увага приділялася обґрунтуванню показника охоплення на основі реалістичних метрик, а також оцінці трудовитрат у людино-днях або людино-тижнях. Порівняно з ICE, застосування RICE вимагало від здобувачів вищої освіти залучення кількісних аргументів і демонструвало вищий рівень аналітичної складності.

У представлений таблиці видно, що найвищий пріоритет (Score = 213) отримала функція «Базовий календар», що свідчить про її суттєве охоплення користувачів (Reach = 1000), високу оцінку впливу (Impact = 2) і впевненість команди у правильності прогнозів (Confidence = 8). Цей результат демонструє, що базовий календар сприймається як ключовий елемент функціональності продукту, здатний забезпечити значну користувацьку цінність при помірних витратах ресурсів.

Другою за пріоритетом є функція «Інтеграція Google Calendar» із результатом 187, що також характеризується високим охопленням (Reach = 800) та стабільним рівнем впливу (Impact = 2). Це свідчить про важливість інтеграційних можливостей у забезпеченні безперервності користувацького досвіду. Дещо нижчі результати отримали «Smart Reminders (розумні нагадування)» із показником 167, що вказує на суттєвий потенціал з точки зору зручності користування, але з нижчим рівнем впевненості в оцінках (Confidence = 6), ймовірно через більшу технічну невизначеність у реалізації.

Наступні позиції займають «Інтеграція Outlook/Microsoft 365» (103) та «LMS/Email: AI-парсинг дедлайнів» (89), які демонструють нижчі значення охоплення та впливу. Це може бути пов'язано з тим, що їхня цінність виявляється лише для окремих груп користувачів або в межах конкретних бізнес-сценаріїв. Водночас нижчий Score не означає відсутності важливості – скоріше, ці функції розглядаються як другорядні елементи розвитку продукту після завершення основного функціоналу.

Найнижчі результати мають «Безпека та контроль доступів» (78), «Smart Scheduler» (75) і «Telegram: AI-парсинг подій із чатів» (75). Незважаючи на потенційну технічну складність і стратегічну значущість у довгостроковій перспективі, їхній низький показник RICE пояснюється меншим охопленням користувачів або високими витратами на реалізацію, що знижує ефективність інвестицій на поточному етапі розвитку продукту.

Завершальним етапом лабораторної роботи стало порівняння результатів пріоритизації, отриманих за методами ICE та RICE. Аналізувалися розбіжності в



рейтингах функцій, аргументували причини зміни пріоритетів та формулювали висновки щодо доцільності використання кожного з методів залежно від стадії розвитку продукту. У більшості проєктів було зафіксовано, що функції з високими оцінками за ICE втрачали свої позиції в RICE через значні трудовитрати або низьку впевненість у впливі, що свідчить про ефективність RICE як інструменту зниження суб'єктивності рішень.

Порівняння результатів пріоритизації функціоналу ІТ-продукту за методами ICE та RICE, представлених у середовищі Nygger, дозволяє виявити як спільні закономірності, так і суттєві відмінності в оцінці значущості окремих фіч. Обидва підходи спрямовані на визначення оптимальної послідовності реалізації функціоналу з урахуванням його впливу, доцільності та ресурсної складності, однак вони спираються на різні аналітичні акценти.

У межах ICE-таблиці найвищий підсумковий бал отримала функція «Базовий календар» (Score = 392), що характеризується високими оцінками за всіма трьома параметрами – Impact, Confidence та Ease. Аналогічна закономірність спостерігається і в RICE-таблиці, де ця ж функція посіла перше місце із результатом 213. Таким чином, можна стверджувати, що обидва методи сходяться у визначенні стратегічної важливості базового календаря як ключової складової функціональності продукту. Це свідчить про стабільність оцінки, коли фіча має високий користувацький потенціал і відносно низьку складність реалізації.

Разом із тим, подальші позиції у рейтингах демонструють суттєві відмінності. Метод ICE приділяє більшу увагу внутрішній оцінці впливу та впевненості команди, що визначається переважно експертним шляхом. Тому друге та третє місця посіли «Smart Reminders» та «Інтеграція Google Calendar» з ідентичними балами (336). Натомість у моделі RICE, яка враховує параметр Reach – тобто очікуване охоплення користувачів, – пріоритетність цих фіч зміщується: «Інтеграція Google Calendar» випереджає «Smart Reminders» завдяки ширшій потенційній аудиторії (Reach = 800 проти 900 у нагадувань) і вищій впевненості в цінності інтеграції для користувачів корпоративного сегменту.

Ще однією відмінністю є позиції функцій «Інтеграція Outlook/Microsoft 365» та «LMS/Email: AI-парсинг дедлайнів». У методі ICE вони отримали середній рівень пріоритетності (Score 180 і 216 відповідно), тоді як за RICE їхня важливість істотно знижується (103 і 89). Така розбіжність пояснюється тим, що метод RICE враховує обмежене охоплення цих фіч (Reach 600-650), тобто вони мають помітний ефект лише для частини користувачів. Це вказує на більшу маркетингову орієнтацію RICE, який оцінює продукт не лише з позицій команди розробки, а й з точки зору бізнес-цінності та впливу на ринок.

Особливо помітною є різниця у сприйнятті фіч із низьким пріоритетом. У моделі ICE останні місця займають «Smart Scheduler» і «Telegram: AI-парсинг подій із чатів», тоді як у RICE вони отримують подібні, але не найнижчі показники (75), що демонструє більш «згладжений» характер диференціації. Це пояснюється тим, що параметр Reach у RICE дозволяє компенсувати низькі оцінки Impact або Confidence за рахунок потенційної кількості користувачів. Таким чином, RICE показує себе як більш гнучкий і комплексний підхід, здатний поєднувати технічну доцільність і ринкову значущість.

Під час опрацювання описаної лабораторної роботи варто виокремити навчальні результати, які формуються у здобувачів вищої освіти через використання ICE та RICE: по-перше, розвиток аналітичного мислення та data-driven підходу, тобто обидві моделі



навчають ухвалювати рішення на основі кількісних показників, а не інтуїції, що сприяє оволодінню навичками роботи з метриками (impact, reach, confidence, effort), що є базовими для сучасного product management; по-друге, це формування системного продуктового мислення завдяки аналізу взаємозалежностей між користувацькою цінністю, трудовитратами та масштабом впливу; по-третє, це уміння обґрунтовано визначати пріоритети функціоналу – формує практичну компетентність у пріоритизації беклогу продукту, що є одним з ключових умінь для ролей Product Owner, Scrum Master та Business Analyst; по-четверте, порівняння результатів двох моделей допомагає усвідомити суб'єктивність власних оцінок, виявити типові помилки (наприклад, переоцінку впливу або недооцінку трудовитрат) і зробити висновки щодо реалістичності своїх продуктових гіпотез; по-п'яте, це робота у середовищі Hygger формує вміння використовувати професійні платформи для аналітики та управління беклогом, що підвищує технологічну готовність до роботи у Scrum/Agile-командах.

У цілому, результати порівняння свідчать, що метод ICE краще підходить для внутрішнього планування невеликих або стартап-команд, де рішення приймаються на основі експертної оцінки ефекту й витрат. Натомість RICE є більш науково обґрунтованою моделлю, яка забезпечує багатофакторну оцінку завдяки параметру Reach, що відображає масштаб користувацького впливу. З педагогічної точки зору, поєднане використання цих двох підходів у освітньому процесі є надзвичайно корисним: ICE дозволяє здобувачам вищої освіти опанувати базові принципи продуктової аналітики, тоді як RICE формує глибше розуміння ринкових аспектів, кількісного моделювання й формалізації рішень.

З огляду на вищезазначене, моделі RICE та ICE можуть вважатися найбільш релевантними інструментами для формалізації процесу пріоритизації функціоналу власного IT-продукту в освітній підготовці майбутніх IT-фахівців. Вони забезпечують комплексне поєднання простоти, кількісної строгості та концептуальної узгодженості з сучасними підходами до управління IT-продуктами. Їхнє застосування в освітньому процесі не лише підвищує якість навчання у сфері продуктової аналітики, а й формує у здобувачів вищої освіти системне бачення циклу створення цінності в цифровій економіці.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Отримані результати засвідчили, що застосування RICE сприяє глибшому аналітичному мисленню та формуванню у майбутніх IT-фахівців навичок аргументованого прийняття рішень на основі даних, тоді як ICE є доцільним інструментом для первинного відбору ідей на ранніх етапах розробки продукту. Саме етап порівняння двох методів, передбачений завданням лабораторної роботи, виявився найбільш цінним з освітньої точки зору, оскільки стимулював до рефлексії щодо доцільності вибору тієї чи іншої методики залежно від стадії життєвого циклу IT-продукту та наявності даних.

Виявлено, що методи RICE (Reach, Impact, Confidence, Effort) та ICE (Impact, Confidence, Ease) забезпечують раціональний баланс між простотою застосування та аналітичною строгістю. Вони дають змогу кількісно оцінювати гіпотези розвитку продукту, тим самим підвищуючи об'єктивність прийняття рішень у межах обмежених ресурсів. В процесі підготовки майбутніх IT-фахівців це сприяє формуванню навичок формалізації проблеми, логічного структурування даних і побудови аналітичних моделей. Здобувачі вищої освіти, які опановують ці підходи, засвоюють базові



принципи прийняття рішень на основі даних, що є невід'ємною складовою сучасного інженерного мислення.

Важливо підкреслити, що обидва методи стимулюють розвиток системного мислення через застосування мультикритеріальної оцінки. Кожен параметр моделі має чітку інтерпретацію, яка поєднує як кількісні, так і якісні аспекти. Наприклад, змінна *Impact* дозволяє оцінити потенційний вплив функціоналу на кінцевого користувача, тоді як *Confidence* відображає рівень впевненості у правильності попередніх оцінок. Такий підхід формує у розуміння невизначеності у процесі прийняття рішень і привчає до обґрунтованого врахування ризиків. У цьому сенсі RICE та ICE виступають не лише інструментами продуктової аналітики, а й ефективними моделями навчання критичного мислення. Таким чином, ICE виступає як педагогічний інструмент початкової формалізації мислення та швидкого оцінювання гіпотез, тоді як RICE – як інструмент поглибленого аналітичного прийняття рішень на основі даних. Разом вони створюють безперервну навчальну траєкторію – від інтуїтивного до обґрунтованого рівня прийняття рішень, що відображає реальну логіку розвитку продуктів в ІТ-індустрії.

Додатковою перевагою застосування цих моделей у підготовці ІТ-фахівців є їх міждисциплінарність. Вони об'єднують елементи математичного аналізу, економічної оцінки та інформаційного моделювання, що відповідає сучасним вимогам до T-shaped спеціалістів. Завдяки цьому здобувачі вищої освіти одночасно розвивають технічну компетентність і бізнес-орієнтоване мислення, необхідне для успішної роботи в кросфункціональних командах. Формула RICE, у своїй простій математичній структурі, слугує базою для подальшого розширення у напрямі застосування методів оптимізації, машинного навчання та статистичного прогнозування у процесах продуктової аналітики.

Не менш важливим є комунікативний аспект використання моделей RICE та ICE у навчальному середовищі. Процес колективного оцінювання параметрів моделі вимагає аргументованих дискусій, порівнянь та обґрунтувань рішень на основі метрик, а не інтуїції. Це сприяє розвитку *soft skills*, формуванню культури командної взаємодії та прийняття рішень у середовищі невизначеності, що є ключовими складовими сучасного професійного профілю ІТ-спеціаліста.

Таким чином, отримані у статті результати узгоджуються з сучасними тенденціями розвитку ІТ-освіти, які орієнтовані на формування у здобувачів вищої освіти не лише технічних, а й аналітико-управлінських компетентностей, необхідних для роботи у продуктивних командах. Застосування моделей ICE та RICE як інструментів формалізації процесу пріоритизації функціоналу безпосередньо відповідає принципам компетентнісного підходу, оскільки забезпечує розвиток інтегрованих умінь – від аналітичного мислення до прийняття рішень на основі даних. Сучасний ІТ-ринок вимагає фахівців, здатних не лише програмувати, а й розуміти бізнес-логіку продукту, оцінювати його цінність для користувача, ефективно розподіляти ресурси та працювати в умовах обмежень. Саме ці компетентності формуються під час практичного застосування моделей ICE і RICE у навчальних середовищах типу *Nugger*, де здобувачі вищої освіти виконують ролі *Product Owner* або аналітиків. Водночас такий підхід відображає зміщення акценту в ІТ-освіті від традиційного предметного навчання до проєктно-орієнтованої освіти, у межах якої здобувачі опановують навички планування, командної взаємодії та оцінки ефективності розробки. Таким чином, результати дослідження демонструють, що інтеграція ICE та RICE у освітній процес сприяє підготовці конкурентоспроможних ІТ-фахівців, здатних діяти в парадигмі *data-driven product development* і відповідати вимогам індустрії 4.0.



СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. CB Insights. (n.d.). *The top 12 reasons startups fail*. <https://www.cbinsights.com/research/report/startup-failure-reasons-top/>
2. Pichler, R. (2016). *Strategize: Product strategy and product roadmap practices for the digital age*. Pichler Consulting. <https://agilerequirements.ir/wp-content/uploads/2021/07/Strategize-Product-Strategy-and-Product-Roadmap-Practices-for-the-Digital-Age.pdf>
3. Schwaber, K., & Sutherland, J. (2020). *The Scrum guide: The definitive guide to Scrum: The rules of the game*. Scrum.org. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
4. Ries, E. (2011). *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Crown Business. <https://ia800509.us.archive.org/7/items/TheLeanStartupErickRies/The%20Lean%20Startup%20-%20Erick%20Ries.pdf>
5. Maurya, A. (2022). *Running lean: Iterate from Plan A to a plan that works*. O'Reilly Media. <https://belinitiative.com/wp-content/uploads/2021/07/Ash-Maurya-Running-Lean-Iterate-from-Plan-A-to-a-Plan-That-Works.pdf>
6. ProductPlan. (n.d.). *The 2021 state of product management annual report*. <https://www.productplan.com/state-of-product-management-report/>
7. GrowthMethod. (n.d.). *The HiPPO effect: A dangerous animal in growth*. <https://growthmethod.com/hippo-effect/>
8. Cagan, M. (2017). *Inspired: How to create tech products customers love*. Wiley.
9. Olsen, D. (2015). *The lean product playbook: How to innovate with minimum viable products and rapid customer feedback*. Wiley.
10. Intercom. (n.d.). *RICE: Simple prioritization for product managers*. <https://www.intercom.com/blog/rice-simple-prioritization-for-product-managers/>
11. Lumi Studio. (n.d.). *ICE scoring: How to prioritise competing ideas for startups*. <https://www.lumi.studio/blog/ice-scoring>
12. PPM Express. (n.d.). *From RICE to WSJF: 13 prioritization techniques to improve your project's workflow*. <https://www.ppm.express/blog/13-prioritization-techniques>
13. Tasneem, N., Zulzalil, H. B., & Hassan, S. (2025). Enhancing agile software development: A systematic literature review of requirement prioritization and reprioritization techniques. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2025.3539357>
14. Seniv, M. M. (2021). Tool for selecting software development methodology considering project metrics. *Scientific Bulletin of UNFU*, 31(3), 120–125. <https://doi.org/10.36930/40310318>
15. Buijs, R. (n.d.). RICE vs ICE: Which prioritization framework should you use? <https://www.productlift.dev/blog/rice-vs-ice>
16. Atlassian. (n.d.). *Backlogs in Scrum*. <https://www.atlassian.com/agile/scrum/backlogs>
17. Savio. (n.d.). *What is the ICE scoring framework?* <https://www.savio.io/product-roadmap/ice-scoring-model/>
18. Zeda.io. (n.d.). *Decoding ICE scoring prioritization model*. <https://zeda.io/blog/ice-scoring-prioritisation-model>
19. Productboard. (n.d.). *2024 product management trends*. <https://www.productboard.com/ebook/2024-product-management-trends/>
20. ProductPlan. (n.d.). *RICE scoring model*. <https://www.productplan.com/glossary/rice-scoring-model/>
21. Product School. (n.d.). *How to use the RICE framework for better prioritization*. <https://productschool.com/blog/product-fundamentals/rice-framework>
22. ProductPlan. (n.d.). *The 2025 state of product management report*. <https://www.productplan.com/2025-state-of-product-management-report/>
23. Tsarov, V. M. (2023). Some aspects of modifying task prioritization frameworks in product IT. *Scientific Notes*, 31(2), 263–272. https://doi.org/10.33111/vz_kneu.31.22.02.23.159.165
24. Yatsenko, V. (2025). Comparative analysis of feature prioritization methods: ICE, RICE, WSJF, MoSCoW and their applicability for startups. *Nauka Online*, (10). <https://nauka-online.com/publications/economy/2025/10/06-29/>
25. Kuchakovska, H., & Khorolska, K. (2025). Integration of Lean Canvas into IT specialist training as a mechanism for simplifying complex decision-making. *Cybersecurity: Education, Science, Technique*, 2(30), 305–315. <https://doi.org/10.28925/2663-4023.2025.30.976>

**Halyna Kuchakovska**

Candidate of Pedagogical Sciences, Senior Lecturer at the Department of Computer Science
Borys Grinchenko Kyiv Metropolitan University, Kyiv, Ukraine
ORCID: 0000-0002-4555-896X
h.kuchakovska@kubg.edu.ua

Karyna Khorolska

PhD in Computer Science, Associate Professor at the Department of Information and Cyber Security named after Prof. Volodymyr Buriachok
Borys Grinchenko Kyiv Metropolitan University, Kyiv, Ukraine
ORCID: 0000-0003-3270-4494
k.khorolska@kubg.edu.ua

RICE AND ICE MODELS AS TOOLS FOR FORMALIZING PRIORITIZATION OF IT PRODUCT FUNCTIONALITY IN TRAINING FUTURE IT SPECIALISTS

Abstract. The relevance of the study is determined by the growing role of product solutions in the development of IT products and the need to develop skills in future IT-specialists for the rational prioritization of functionality in conditions of limited resources and high uncertainty. Insufficient formalization of this process in educational training leads to the dominance of intuitive decisions that do not meet the modern requirements of Agile and data-driven approaches. The article aims to investigate the effectiveness of ICE and RICE models in formalizing the process of prioritizing IT product functionality in the early stages of the life cycle through comparative analysis and empirical testing, and to determine the pedagogical feasibility of their application in training future IT-specialists. The research methodology is based on an empirical analysis of the results of laboratory work performed by third-year students majoring in 122 “Computer Science” within the discipline “Entrepreneurship and Startups.” The study uses methods of comparative analysis, formalisation, and generalisation of the results of functional prioritisation using ICE and RICE models in the Hygger environment. The results showed that the ICE model is an effective tool for the initial selection of ideas at the product conceptualization stage, while the RICE model provides a more objective and analytically sound assessment of priorities when quantitative data is available. It was concluded that the combined use of both models contributes to the formation of systematic product thinking and skills in students. The results showed that the ICE model is an effective tool for the initial selection of ideas at the product conceptualization stage, while the RICE model provides a more objective and analytically sound assessment of priorities when quantitative data is available. This approach reflects a shift in IT education from traditional subject-based learning to project-oriented education, in which students learn planning, teamwork, and assessment of development effectiveness. Thus, the study's results demonstrate that integrating ICE and RICE into the educational process prepares competitive IT specialists to operate within the data-driven product development paradigm and meet Industry 4.0 requirements.

Keywords: feature prioritization; RICE; ICE; Product Management; Agile; IT product development; startups; feature evaluation methods.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. CB Insights. (n.d.). *The top 12 reasons startups fail*. <https://www.cbinsights.com/research/report/startup-failure-reasons-top/>
2. Pichler, R. (2016). *Strategize: Product strategy and product roadmap practices for the digital age*. Pichler Consulting. <https://agilerequirements.ir/wp-content/uploads/2021/07/Strategize-Product-Strategy-and-Product-Roadmap-Practicals-for-the-Digital-Age.pdf>
3. Schwaber, K., & Sutherland, J. (2020). *The Scrum guide: The definitive guide to Scrum: The rules of the game*. Scrum.org. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
4. Ries, E. (2011). *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Crown Business.



- <https://ia800509.us.archive.org/7/items/TheLeanStartupErickRies/The%20Lean%20Startup%20-%20Erick%20Ries.pdf>
5. Maurya, A. (2022). *Running lean: Iterate from Plan A to a plan that works*. O'Reilly Media. <https://belinitiative.com/wp-content/uploads/2021/07/Ash-Maurya-Running-Lean-Iterate-from-Plan-A-to-a-Plan-That-Works.pdf>
 6. ProductPlan. (n.d.). *The 2021 state of product management annual report*. <https://www.productplan.com/state-of-product-management-report/>
 7. GrowthMethod. (n.d.). *The HiPPO effect: A dangerous animal in growth*. <https://growthmethod.com/hippo-effect/>
 8. Cagan, M. (2017). *Inspired: How to create tech products customers love*. Wiley.
 9. Olsen, D. (2015). *The lean product playbook: How to innovate with minimum viable products and rapid customer feedback*. Wiley.
 10. Intercom. (n.d.). *RICE: Simple prioritization for product managers*. <https://www.intercom.com/blog/rice-simple-prioritization-for-product-managers/>
 11. Lumi Studio. (n.d.). *ICE scoring: How to prioritise competing ideas for startups*. <https://www.lumi.studio/blog/ice-scoring>
 12. PPM Express. (n.d.). *From RICE to WSJF: 13 prioritization techniques to improve your project's workflow*. <https://www.ppm.express/blog/13-prioritization-techniques>
 13. Tasneem, N., Zulzalil, H. B., & Hassan, S. (2025). Enhancing agile software development: A systematic literature review of requirement prioritization and reprioritization techniques. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2025.3539357>
 14. Seniv, M. M. (2021). Tool for selecting software development methodology considering project metrics. *Scientific Bulletin of UNFU*, 31(3), 120–125. <https://doi.org/10.36930/40310318>
 15. Buijs, R. (n.d.). RICE vs ICE: Which prioritization framework should you use? <https://www.productlift.dev/blog/rice-vs-ice>
 16. Atlassian. (n.d.). *Backlogs in Scrum*. <https://www.atlassian.com/agile/scrum/backlogs>
 17. Savio. (n.d.). *What is the ICE scoring framework?* <https://www.savio.io/product-roadmap/ice-scoring-model/>
 18. Zeda.io. (n.d.). *Decoding ICE scoring prioritization model*. <https://zeda.io/blog/ice-scoring-prioritisation-model>
 19. Productboard. (n.d.). *2024 product management trends*. <https://www.productboard.com/ebook/2024-product-management-trends/>
 20. ProductPlan. (n.d.). *RICE scoring model*. <https://www.productplan.com/glossary/rice-scoring-model/>
 21. Product School. (n.d.). *How to use the RICE framework for better prioritization*. <https://productschool.com/blog/product-fundamentals/rice-framework>
 22. ProductPlan. (n.d.). *The 2025 state of product management report*. <https://www.productplan.com/2025-state-of-product-management-report/>
 23. Tsarov, V. M. (2023). Some aspects of modifying task prioritization frameworks in product IT. *Scientific Notes*, 31(2), 263–272. https://doi.org/10.33111/vz_kneu.31.22.02.23.159.165
 24. Yatsenko, V. (2025). Comparative analysis of feature prioritization methods: ICE, RICE, WSJF, MoSCoW and their applicability for startups. *Nauka Online*, (10). <https://nauka-online.com/publications/economy/2025/10/06-29/>
 25. Kuchakovska, H., & Khorolska, K. (2025). Integration of Lean Canvas into IT specialist training as a mechanism for simplifying complex decision-making. *Cybersecurity: Education, Science, Technique*, 2(30), 305–315. <https://doi.org/10.28925/2663-4023.2025.30.976>

Отримано редакцією журналу / Received: 17.01.26

Прорецензовано / Revised: 02.02.26

Схвалено до друку / Accepted: 26.03.26

