

DOI: 10.36910/6775-2524-0560-2019-37-2
УДК: 004.891

Асеева А.В., Кулаковська І. В.

Чорноморський національний університет імені Петра Могили.

АНАЛІЗ ПРОБЛЕМ ВИБОРУ ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Асеева А.В., Кулаковська І.В. Аналіз проблем вибору технології для розробки програмного забезпечення. У розробників є досить широкий вибір технологій для реалізації розробки програмного забезпечення інформаційних систем, з часом зростає і складність таких проектів. Проведено аналіз найвідоміших методологій Agile і Waterfall, щоб визначити, яка є найбільш підходящою для різних програмних проектів, обрано критерії оцінювання. Проведено практичне дослідження шляхом опитування експертів, аналізу результатів та розробленої моделі з метою врахування досвіду розробників за вищезгаданими методологіями. В статті проаналізовані існуючі технології розробки програмного забезпечення, обрано математичну модель для створення системи підтримки прийняття рішень, досліджено питання вибору в умовах багатокритеріальності. Методом для вирішення поставленої задачі став метод Парето-оптимальності та метод звуження множини Парето на основі інформації про відносну важливість критеріїв. Методом дослідження є комп'ютерна програма для моделювання СППР. Практичне значення системи полягає у використанні її у разі прийняття рішення вибору технології розробки ПЗ ІС на основі методу Парето-оптимальності.

Ключові слова: програмне забезпечення, розробка, сервіс, життєвий цикл програмного забезпечення, Парето-оптимальність.

Асеева А.В., Кулаковская И. В. Анализ проблем выбора технологии для разработки программного обеспечения. У разработчиков есть достаточно широкий выбор технологий для реализации разработки программного обеспечения информационных систем, со временем растет и сложность таких проектов. Проведен анализ известных методологий Agile и Waterfall, чтобы определить, которая является наиболее подходящей для различных программных проектов, выбраны критерии оценки. Проведено практическое исследование путем опроса экспертов, анализа результатов и разработанной модели с целью учета опыта разработчиков по вышеуказанным методологиями. В статье проанализированы существующие технологии разработки программного обеспечения, выбрано математическую модель для создания системы поддержки принятия решений, исследован вопрос выбора в условиях многокритериальности. Методом для решения поставленной задачи стал метод Парето-оптимальности и метод сужения множества Парето на основе информации об относительной важности критериев. Методом исследования является компьютерная программа для моделирования СППР. Практическое значение системы заключается в использовании ее в случае принятия решения выбора технологии разработки ПО ИС на основе метода Парето-оптимальности.

Ключевые слова: программное обеспечение, разработка, сервис, жизненный цикл программного обеспечения, Парето-оптимальность.

Asieieva A.V., Kulakovska I.V. Analysis of technology selection problems for software development. The developers have a wide choice of technologies for the implementation of information systems software development, and over time the complexity of such projects increases. The most well-known Agile and Waterfall methodologies are analyzed to determine which is most appropriate for different software projects, and the evaluation criteria are selected. Practical research was conducted by interviewing experts, analyzing the results and developing a model to take into account the experience of developers using the methodologies mentioned above. The article analyzes the existing technologies of software development, the mathematical model for creation of the decision support system, the choice of choice in terms of multicriteria. The Pareto-optimality method and the Pareto set narrowing method based on information about the relative importance of the criteria became the method for solving this problem. The method of research is a computer program for modeling the DSS. The practical importance of the system is to use it when deciding to choose the technology of IP software development based on the Pareto-optimality method.

Keywords: software, development, service, software life cycle, Pareto optimality.

Постановка наукової проблеми. На сьогоднішній день у розробників є досить широкий вибір технологій для реалізації програмного забезпечення інформаційних систем. Також з вибором зростає і складність таких систем. На даний момент, виходячи з аналізу літературних даних, розробка програмного забезпечення (ПЗ) залишається не на достатньому рівні. Відомо, що 30-40% всіх проектів не завершуються взагалі. 70% проектів не реалізують поставленої задачі повністю, а середній проект завершується з запізненням в 220%. В 10% проектів результат не відповідає вимогам. В 12% замовник недостатньо залучався до роботи для того щоб забезпечити характеристики продукту. В 22% проектів не всі зміни які вносилися приймалися до уваги.

При реалізації проектів з розробки програмного забезпечення інформаційних систем (ПЗ ІС) доцільно використовувати різні технології, які зменшать відсоток невдалих проектів та зменшить час виконання проекту. Залишається відкрите питання вибору технології для розробки програмного

забезпечення. Одна технологія може бути кращим варіантом для однієї задачі, а інша технологія може навпаки загальмувати або взагалі призвести до краху проекту. Існує багато методологій управління проектами, які можна вибрати, починаючи новий проект. Метою цієї статті є аналіз найвідоміших методологій Agile і Waterfall, щоб визначити, яка є найбільш підходящою для програмного проекту. Планується практичне дослідження шляхом аналізу результатів опитування експертів, розробленого з метою врахування досвіду розробників з вищезгаданими методологіями. Ми зупинимося на методах Agile та Waterfall для аналізу результатів дослідження. Метою роботи є поліпшення вибору технології програмного забезпечення шляхом створення системи підтримки прийняття рішень, що веде до зменшення ризиків невдало завершених проектів.

Для цього необхідно проаналізувати існуючі технології розробки програмного забезпечення. обрати актуальну платформу для створення системи підтримки прийняття рішення, дослідити питання вибору в умовах багатокритерійності. Проаналізувати математичну модель прийняття рішення та створити власну систему підтримки прийняття рішення, проаналізувати отримані результати та виробити рекомендації для їх практичного застосування. Методом для вирішення поставленої задачі став метод Парето-оптимальності та метод звуження множини Парето на основі інформації про відносну важливість критеріїв. Методом дослідження є комп'ютерна програма для моделювання СППР.

Аналіз досліджень. Дослідження базується на працях таких видатних авторів у галузі розробки програмного забезпечення як К. Петерсона [1], Д. Шора [2], Л. Марка [3], К. Бека [4], Б. Бохема [5]. У своїх дослідженнях Вільямс [6,7] представляє кілька відомих планових та швидких SDM (System Development Model). Порівнюють дві моделі Agile і Waterfall в дослідженні [12].

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження
Найчастіше, коли необхідно прийняти рішення про вибір методології занадто багато різномірної інформації і важко зрозуміти, що саме краще підійде для проекту. Існує схема вибору методології, що дозволяє звернути увагу на деякі найбільш важливі аспекти.

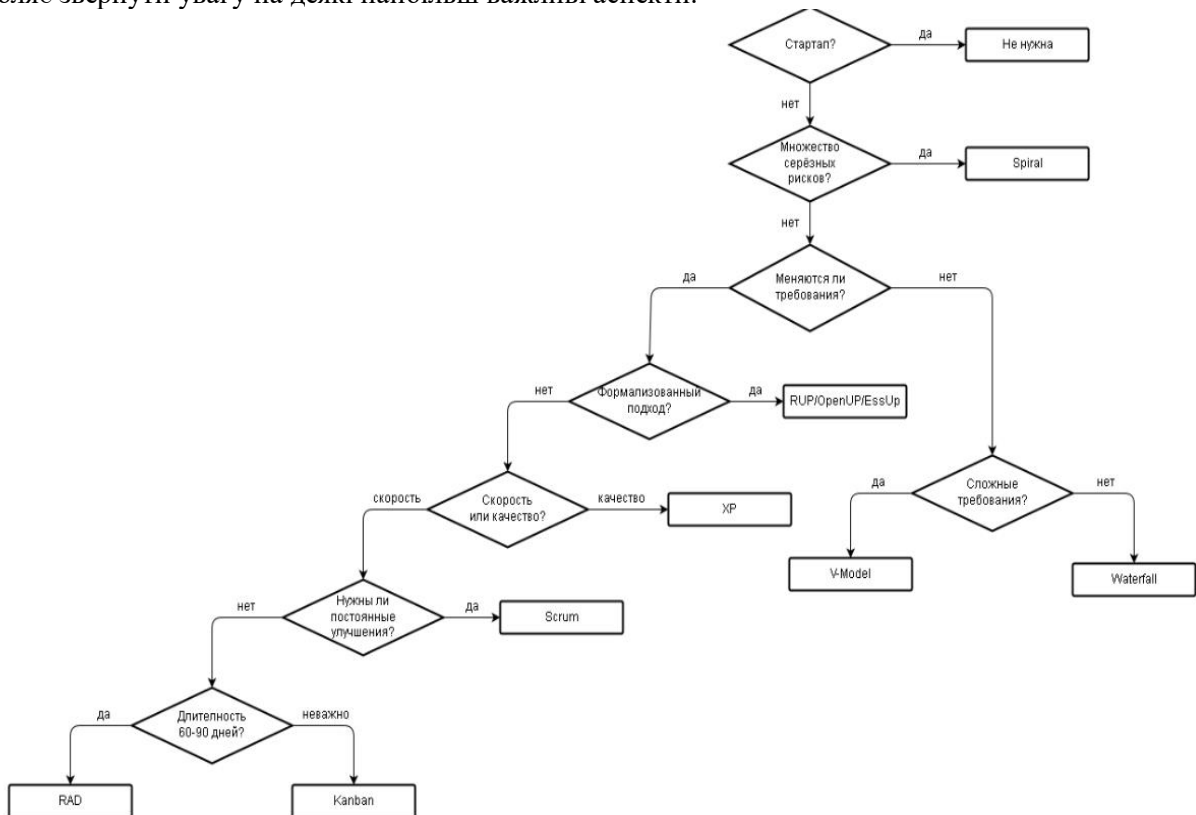


Рис. 1. Схема вибору методології розробки ПЗ

Універсального набору умов для всіх ситуацій при виборі тієї чи іншої методології не існує. у кожному разі РМ(ОПР) повинен орієнтуватися на специфіку свого проекту. Блок-схема лише

позначає головні аспекти та дозволить пригадати особливості основних методологій. Неможливо ставитися до неї, як до єдиного правильного керівництва за вибором методології. Тим більше не варто поширювати її на такі складні проекти яких потребують сучасні замовники. Вибравши методологію її потрібно адаптувавши під свій проект. Щось можна викинути, щось додати з інших методологій, внести щось своє.

Каскадна методологія

Однією з перших почала застосовуватися каскадна модель, де кожна робота виконується один раз і в такому порядку, який подано на рис.2.а.[8]

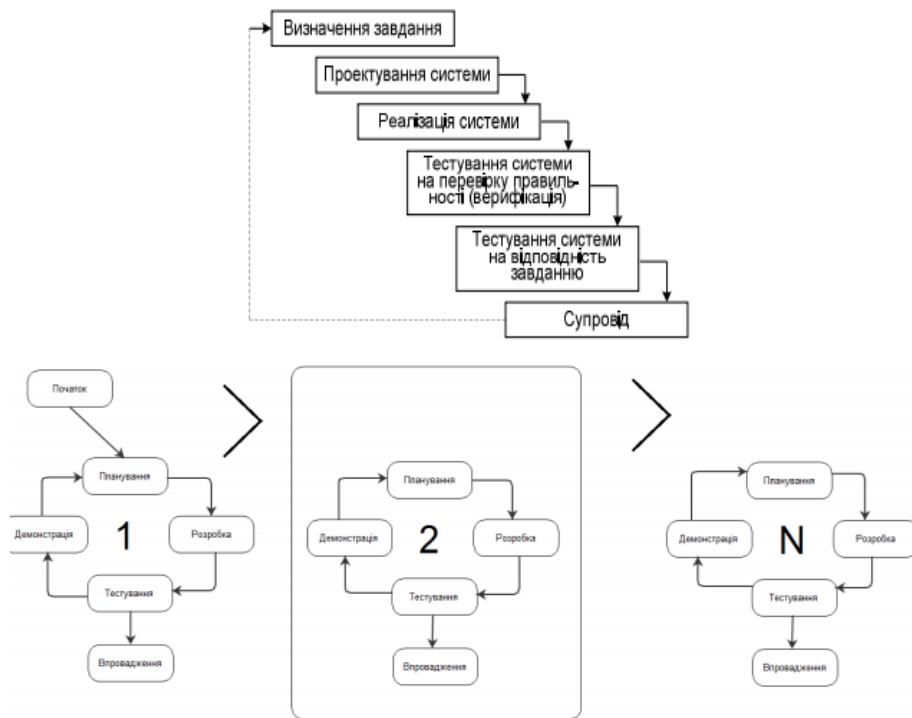


Рис. 2.

а) Каскадна модель ЖЦ програмних систем

б) Життєвий цикл гнучкої методології

Вважається, що кожна робота має бути виконана настільки ретельно, що після її закінчення і переходу до наступного етапу, повертатися до попереднього не буде потреби. Розробник перевіряє проміжний результат відомими методами верифікації і фіксує його як готовий еталон для наступного процесу. Згідно з даною моделлю роботи і завдання процесу розроблення зазвичай виконуються послідовно, як це наведено у схемі. Проте допоміжні і організаційні процеси (контроль вимог, керування якістю і ін.), як правило, виконуються разом з процесами розробки ПЗ. У даній моделі повернення до початкового процесу передбачається після супроводження і виправлення помилок. Особливість такої моделі полягає у фіксації послідовних процесів розроблення програмного продукту. В її основу покладена модель фабрики, де продукт проходить стадії від задуму до виробництва, потім його передають замовнику у вигляді готового виробу, де заміна не передбачена, хоча можна подати інший подібний виріб.

Недоліки цієї моделі такі: – процес створення ПЗ не завжди вкладається в таку жорстку форму і послідовність дій; – не враховуються змінювані потреби користувачів, нестабільні умови зовнішнього середовища, які впливають на зміни вимог до ПЗ під час її розроблення; – значний розрив між часом внесення помилки (наприклад, на процесі проектування) і часом її виявлення (при супроводі), що призводить до суттєвої переробки ПЗ.

При застосуванні Waterfall моделі можуть спостерігатися такі чинники ризику: – вимоги до ПЗ недостатньо чітко сформульовані, або не враховують перспективи розвитку ОС, середовищ і т.п.; –

громіздка система, що не допускає компонентної декомпозиції, може викликати проблеми щодо розміщення її в пам'яті або на платформах, не передбачених у вимогах; – внесення швидких змін до технології і у вимоги може погіршити процес розроблення окремих частин системи або системи в цілому; – обмеження на ресурси (людські, програмні, технічні і ін.) в ході розробки можуть звужити окремі можливості реалізації системи; – отриманий продукт може виявитися не придатним для застосування внаслідок нерозуміння розробниками вимог або функцій системи або недостатньо проведеного тестування. [1]

Переваги реалізації системи за допомогою каскадної моделі такі: – всі завдання підсистем і системи реалізуються одночасно, завдяки чому не можна забути жодного завдання, а це сприяє встановленню стабільних зв'язків між ними; – повністю розроблену систему з документацією на неї легше супроводжувати, тестувати, фіксувати помилки і вносити зміни не хаотично, а цілеспрямовано, починаючи з вимог, наприклад, додавати або замінювати деякі функції і повторювати процес.

Каскадну модель можна розглядати як модель ЖЦ, придатну для створення першої версії ПЗ з метою перевірки реалізованих в ній функцій. При супроводі і експлуатації можуть бути виявлені різного роду помилки, виправлення яких спричинить повторне виконання всіх процесів, починаючи з уточнення вимог.

Гнучка методологія

Гнучка методологія використовується для проектування впорядкованого управління процесом розробки котрий дозволяє вносити постійні зміни в розробку проекту [8,9,10,11]. Ця методологія являється однією з концептуальних основ для створення різних проектів в галузі розробки програмного забезпечення. Ця модель використовується для максимального зменшення ризику при розробці продукту в короткі часові проміжки котрі називаються ітераціями і зазвичай тривають від одного тижня до одного місяця. Життєвий цикл методології наведено на рис. 2.б. Цю модель слід застосовувати коли потреби користувачів постійно уточнюються в динамічному процесі. Зміни на Agile реалізуються за меншу ціну із-за постійних спринтів. На відміну від каскадної моделі, в гнучкій моделі для старту проекту досить лише невеликого планування. [2,9,11]

Переваги моделі: — гнучка методологія має адаптивний підхід котрий дозволяє змінювати вимоги клієнтів; — безпосередній зв'язок та постійні відгуки замовників або їх представників не залишають місця для невизначеностей.

Недоліки моделі:— ця методологія зосереджена на створенні програмного забезпечення раніше, ніж документації, звідси може бути нестача документації; — процес розробки може вийти з під контроль, якщо замовник чітко не представляє кінцевий результат проекту.

В статті [12, Summer 2019] проведено аналіз найвідоміших методологій Agile і Waterfall для визначення, яка є найбільш підходящою для програмного проекту. Запропоновано практичне дослідження шляхом аналізу результатів опитування, розробленого з метою врахування досвіду розробників з вищезгаданих методологій. З огляду на результати дослідження, дійшли висновку, що ідеального рішення не існує, оскільки необхідно враховувати численні фактори. Результати порівняння наведено в таблиці 1.

Таблиця 1. Порівняння гнучкої та каскадної технологій

	Agile	Waterfall
Зміст, створення, розробки	Гнучка модель розробки, заснована на ітеративних принципах, 2001 р., Група ІТ-фахівців (США)	Каскадна система розробки, заснована на жорсткій послідовності процесу розробки, 1956 року, 1961 р, 1970 г. Г. Беннінгтон, Хозьєр, В. Уолкер Ройс
принципи застосування	<ul style="list-style-type: none"> • найвищий пріоритет в задоволенні потреб замовника • протягом усього проекту команда і замовник щодня взаємодіють між собою і один з одним • працює продукт - головний показник прогресу • роботу можна довірити тільки самоорганізованої, мотивованій команді • оптимальні терміни випуску робочого 	<ul style="list-style-type: none"> • жорстка послідовність етапів розробки • перехід до нового етапу - тільки після успішного завершення попереднього • фіксована вартість продукту • замовник не залучається до безпосереднього процесу розробки • зміни можуть бути внесені тільки після завершення всього процесу розробки.

	продукту - від 2 тижнів до 2 місяців.	
Переваги	<ul style="list-style-type: none"> високий рівень взаємодії між членами команди проекту швидкий результат (робочий код) в результаті «спринтів» стимулювання зміни і поліпшень продукту під час його розробки безпосереднє залучення замовника до робочого процесу. 	<ul style="list-style-type: none"> зрозуміла і чітка схема робочого процесу можливість прорахунку точної кількості витрачених на проект ресурсів не вимагає витрат по налагодженню комунікацій між усіма членами команди.
недоліки	<ul style="list-style-type: none"> ризик нескінченних змін продукту велика залежність від рівня кваліфікації та досвіду команди практично неможливо точно підрахувати підсумкову вартість проекту. 	<ul style="list-style-type: none"> пріоритет формального підходу до послідовності процесу роботи неможливість внесення змін замовником до закінчення розробки продукту в разі нестачі ресурсів страждає якість проекту через скорочення етапу тестування.
Критерії так	<ul style="list-style-type: none"> над проектом працює досвідчена, висококваліфікована команда ви працюєте над стартапом необхідно швидко отримати робочу версію продукту замовник виступає в якості партнера, а не інвестора продукт розробляється в сфері, схильною до постійних змін. 	<ul style="list-style-type: none"> більшість або вся робота над проектом проводиться на аутсорс у вас є чітка концепція продукту, який хочете отримати ви не обмежені в часі і ресурсах створення продукту створення продукту або бізнесу побудовано на дотриманні суворой послідовності виконання завдань.
ні	<ul style="list-style-type: none"> ви не готові витрачати додаткові ресурси на налагодження щоденної стабільної комунікації між усіма учасниками процесу продукт повинен бути створений до конкретного терміну бюджет проекту строго обмежений вам потрібна детальна документація по всім процесам розробки. 	<ul style="list-style-type: none"> ви хочете створити інноваційний продукт або великий проект ви не впевнені в концепції пропонованого проекту фінансові ресурси не є ключовим обмежувачем в вашому проекті.

Проблеми прийняття рішень в останній час все більше привертають до себе увагу науковців. Проблема вибору технології створення програмного забезпечення інформаційних систем не залишилося осторонь. На даний момент все більше і більше нових технологій й вони стають все складнішими та являються рішенням для малого кругу завдань. Тож керівники (ОПР) стикаються з проблемою прийняття рішення щодо застосування окремої технології для створення кінцевого продукту. Прийняття рішення – це комплексний та неоднозначний у часі динамічний процес, що виникає у випадку, коли необхідно обрати найкращий у певному сенсі варіант серед множини альтернативних варіантів для досягнення бажаного або заданого результату.

Для реалізації використання критерії, на які звертають уваги експерти, яких долучали до попереднього опитування. Концепція ПР БКЗПР полягає у свідомому виборі з множини альтернатив однієї. Цей вибір робить ОПР, яка прагне до досягнення своєї певної цілі. В ролі ОПР може виступати чи конкретний індивід, чи група осіб, яка одночасно приймає дане рішення. Метод для розв'язання поставленої задачі обрано метод Парето-оптимальності та метод звуження множини Парето на основі інформації про відносну важливість критеріїв. Сформовано критерії для вирішення задачі. Проаналізовано багатокритерійні задачі.

Було виділено наступні локальні критерії оцінки технології розробки ПЗ.

1. Можливість виділення окремих частин програм у вигляді модулів.
2. Контроль коректності роботи з типами даних.
3. Робота з даними складної структури.
4. Контроль інтерфейсів програмних модулів при роздільній компіляції.
5. Читабельність програм.
6. Захист від помилки програміста.
7. Гнучкість технології.

8. Повнота реалізації функціональності.

Вибір множини Парето проводиться таким чином:

- всі альтернативи попарно порівнюються одна з одною за всіма критеріями;
- якщо при порівнянні будь-яких альтернатив виявляється, що одна з них не краща за іншу ні за одним критерієм, то її можна виключити з розгляду;
- виключену альтернативу не потрібно порівнювати з іншими альтернативами, так як вона явно безперспективна.

Скорочення множини відбувається наступним чином:

1. Перш за все, необхідно встановити пари «нерівноцінних» на думку ОПР критеріїв. Нехай, наприклад, серед них виявилася пара, що складається з i -го і j -го критерію і при цьому згідно інтуїтивним уявленням ОПР про важливість для нього i -й критерій важливіший, ніж j -й.

2. Тепер можна приступити до визначення конкретної величини коефіцієнта відносної важливості i -го критерію в порівнянні з j -м. При цьому потрібно враховувати той факт, що чим більше виявиться цей коефіцієнт, тим змістовнішим буде інформація i , тим самим, на більшу ступінь звуження множини Парето можна розраховувати. Ступінь звуження розраховуються за наступною формулою:

$$\theta_{ij} = \frac{w_{i+1}}{w_j} \quad (1)$$

3. Припустимо, що зазначеним вище способом виявлено цілий набір інформації про відносну важливість критеріїв, що складається в тому, що i_k -й критерій важливіше j_k -го критерію з заданим коефіцієнтом відносної важливості $\theta_{ij} \in (0,1)$, $k = 1, 2, \dots, M$, де $M \leq m/2$. При цьому вважається, що жоден з критеріїв не може бути важливіше самого себе, тобто ні для якого номера $k = 1, 2, \dots, m$ не виконується рівність $i_k = j_k$. Далі слід перерахувати всі менш важливі критерії (номерів певної набору j_1, j_2, \dots, j_m) за формулою

$$\hat{f}_j = \theta_{ij} f_i + (1 - \theta_{ij}) f_j \quad (2)$$

і підставити їх в вихідний векторний критерій f замість колишніх f_{j_k} . В результаті виконаної підстановки утворюється новий векторний критерій f . Далі потрібно знайти множини Парето щодо цього нового векторного критерію. Тим самим, відбудеться звуження множини Парето за рахунок використання набору взаємно незалежної інформації про відносну важливість критеріїв.

Певні аспекти застосування експертних методів багатокритеріальної ОПР потребують подальшої розробки. В майбутніх дослідженнях з теми пропоную зосередити увагу на:

- забезпеченні повноти та узгодженості експертних даних;
- підвищенні рівня довіри до них з боку осіб, що приймають рішення (ОПР);

Розв'язання задачі з допомогою методу Парето-оптимальності

В даній задачі вибору ми будемо вибирати між 4 технологій розробки програмного забезпечення інформаційних систем. $y^{(1)}$ – Каскадна технологія, $y^{(2)}$ – Спіральна, $y^{(3)}$ – V-подібна, $y^{(4)}$ – Ітераційна технологія. Для рішення даної задачі було залучено 20 експертів, які дали свою експертну оцінку для кожної технології за восьмома критеріями: f_1 – модульний принцип, f_2 – контроль коректності роботи з типами даних, f_3 – робота з даними складної структури, f_4 – контроль інтерфейсів програмних модулів при роздільній компіляції, f_5 – удобочитаємість програм, f_6 – захист від помилки програміста, f_7 – гнучкість технології, f_8 – повнота реалізації функціональності. Експерти виставляли свою оцінку за 10-ти бальною шкалою в 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 балів.

Кількість критеріїв $m=8$. Позначемо множини з п'яти можливих векторів (оцінок) відповідних технологій розробки програмного забезпечення інформаційних систем через $Y = \{y^{(1)}, y^{(2)}, y^{(3)}, y^{(4)}\}$.

Для усереднення підрахунку біллів для всіх критеріїв використовуємо формулу:

$$\bar{a}_{ij} = \frac{\sum_{i=0}^n a_{ij}}{n}; \quad (3)$$

де a_{ij} – оцінка експерта, n – кількість експертів.

В результаті отримали таблицю:

Таблиця 2. Таблиця усереднених оцінок експерта

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
$y^{(1)}$	5,5	6,3	5	4,5	6,3	6,4	4,1	3,5
$y^{(2)}$	4,5	7,8	4,5	6,3	5,4	6	9,0	5
$y^{(3)}$	7,3	4,5	5,5	5	5,7	5,5	5,0	3,5
$y^{(4)}$	7,1	9,8	5,4	7,2	5,3	6,5	9,4	5,3

У відповідності до описаних вище алгоритмом вважаємо, що $P(Y) = Y$ й починаємо порівнювати вектори з іншими. Порівнюємо вектор $y^{(1)}$ з векторами $y^{(2)}, y^{(3)}, y^{(4)}$

Пару $y^{(1)}, y^{(2)}$ не можна порівняти по відношенню \geq . Пару $y^{(1)}, y^{(3)}$ не можна порівняти по відношенню \geq . Пару $y^{(1)}, y^{(4)}$ також не можна порівняти по відношенню \geq .

Порівнюємо вектор $y^{(2)}$ з векторами $y^{(3)}, y^{(4)}$. Пару $y^{(2)}, y^{(3)}$ не можна порівняти по відношенню \geq . Пару $y^{(2)}, y^{(4)}$ не можна порівняти по відношенню \geq .

Порівнюємо вектор $y^{(3)}$ з векторами $y^{(4)}$. Пару $y^{(3)}, y^{(4)}$ не можна порівняти по відношенню \geq .

Отже, $P(Y) = \{y^{(1)}, y^{(2)}, y^{(3)}, y^{(4)}\}$. Звуження не відбулося і з всіх чотирьох технологій і слід робити вибір.

Припустимо, що ОПР прийняв таке рішення, що критерій f_8 – повнота реалізації функціональності є важливішим ніж f_3 – робота з даними складної структури на другому. ОПР дав такі параметри: $w_8=4, w_3=1$. Розрахуємо коефіцієнти відносної вартості, за формулою (1) :

$$\theta_{83} = \frac{1}{4/1 + 1} = \frac{1}{5} = 0.2$$

Розрахуємо нові коефіцієнти за формулою (2) \hat{f}_3 :

Таблиця 3. Перерахунок коефіцієнта \hat{f}_3

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
$y^{(1)}$	5,5	6,3	$3,5*0,2+(1-0,2)*5$	4,5	6,3	6,4	4,1	3,5
$y^{(2)}$	4,5	7,8	$5*0,2+(1-0,2)*4,5$	6,3	5,4	6	9,0	5
$y^{(3)}$	7,3	4,5	$3,5*0,2+(1-0,2)*5,5$	5	5,7	5,5	5,0	3,5
$y^{(4)}$	7,1	9,8	$5,3*0,2+(1-0,2)*5,4$	7,2	5,3	6,5	9,4	5,3

В результаті отримали наступну таблицю:

Таблиця 4. Результат перерахунку коефіцієнта \hat{f}_3

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
$y^{(1)}$	5,5	6,3	4,7	4,5	6,3	6,4	4,1	3,5
$y^{(2)}$	4,5	7,8	4,84	6,3	5,4	6	9,0	5
$y^{(3)}$	7,3	4,5	5,1	5	5,7	5,5	5,0	3,5
$y^{(4)}$	7,1	9,8	5,38	7,2	5,3	6,5	9,4	5,3

Як ми бачимо, всі вектори залишилися Парето-оптимальними. Наступне рішення ОПР було: критерій f_4 – контроль інтерфейсів програмних модулів при роздільній компіляції важливіший ніж f_1 – модульний принцип на четвертому. ОПР дав такі параметри: $w_8=6, w_3=1$. Розрахуємо коефіцієнти відносної вартості, за формулою (1) :

$$\theta_{41} = \frac{1}{6/1 + 1} = \frac{1}{7} = 0.14$$

Розрахуємо нові коефіцієнти за формулою (2) \hat{f}_1 :

Таблиця 5. Перерахунок коефіцієнта \hat{f}_1

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
$y^{(1)}$	$4,5*0,14+(1-0,14)*5,5$	6,3	4,7	4,5	6,3	6,4	4,1	3,5
$y^{(2)}$	$6,3*0,14+(1-0,14)*4,5$	7,8	4,84	6,3	5,4	6	9,0	5
$y^{(3)}$	$5*0,14+(1-0,14)*7,3$	4,5	5,1	5	5,7	5,5	5,0	3,5
$y^{(4)}$	$7,2*0,14+(1-0,14)*7,1$	9,8	5,38	7,2	5,3	6,5	9,4	5,3

В результаті отримали наступну таблицю:

Таблиця 6. Результат перерахунку коефіцієнта \hat{f}_1

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	
$y^{(1)}$	5,36	6,3	4,7	4,5	6,3	6,4	4,1	3,5	
$y^{(2)}$	4,75	7,8	4,84	6,3	5,4	6	9,0	5	
$y^{(3)}$	6,3	4,5	5,1	5	5,7	5,5	5,0	3,5	≤
$y^{(4)}$	7,17	9,8	5,38	7,2	5,3	6,5	9,4	5,3	

Порівнюючи вектори, ми можемо переконатися в тому, що $y^{(3)}$ вийшов з Парето оптимальності. Тобто цей вектор ми виключаємо.

Отримали наступну таблицю:

Таблиця 7. Таблиця після скорочення вектора $y^{(3)}$

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
$y^{(1)}$	5,36	6,3	4,7	4,5	6,3	6,4	4,1	3,5
$y^{(2)}$	4,75	7,8	4,84	6,3	5,4	6	9,0	5
$y^{(4)}$	7,17	9,8	5,38	7,2	5,3	6,5	9,4	5,3

Далі ОПР прийняв рішення згідно яким критерій f_2 – контроль коректності роботи з типами даних на п'ятому є важливішим ніж критерій f_5 – удобочитаємість програм. ОПР дав такі параметри: $w_2=3$, $w_5=1$. Розрахуємо коефіцієнти відносної вартості, за формулою(1) :

$$\theta_{25} = \frac{1}{3/1 + 1} = \frac{1}{4} = 0.25$$

Розрахуємо нові коефіцієнти \hat{f}_5 :

Таблиця 8. Перерахунок коефіцієнта \hat{f}_5

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
$y^{(1)}$	5,36	6,3	4,7	4,5	$6,3*0,25+(1-0,25)*6,3$	6,4	4,1	3,5
$y^{(2)}$	4,75	7,8	4,84	6,3	$7,8*0,25+(1-0,25)*5,4$	6	9,0	5
$y^{(4)}$	7,17	9,8	5,38	7,2	$9,8*0,25+(1-0,25)*5,3$	6,5	9,4	5,3

В результаті отримали наступну таблицю:

Таблиця 9. Результат перерахунку коефіцієнта \hat{f}_5

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	
$y^{(1)}$	5,36	6,3	4,7	4,5	6,3	6,4	4,1	3,5	
$y^{(2)}$	4,75	7,8	4,84	6,3	6	6	9,0	5	
$y^{(4)}$	7,17	9,8	5,38	7,2	6,425	6,5	9,4	5,3	≥

Порівнюючи вектори, ми можемо переконатися в тому, що вектори $y^{(1)}$ та $y^{(2)}$ більше не є Парето-оптимальні. Тобто ці вектор ми виключаємо. Тобто в результаті звуження множин Парето на основі інформації про відносну важливість критеріїв ми прийшли до висновку, що в даному випадку найбільш оптимальною є технологія $y^{(4)}$, тобто Ітераційна технологія.

Висновки дослідження. Зростаючий попит на програмне забезпечення створює необхідність правильної постановки процесів в підрозділах розробки, аби вони могли задовольняти потреби бізнесу та створювати якісний, підтримуваний та функціональний продукт. Існуючі методології розробки були створені та перевірені часом, але їх недоліки інколи є критичними для їх користувачів, що відкриває простір для покращення та створення новий підходів до розробки програмного забезпечення.

Були представлені дві СРПЗ, орієнтовані на модель Waterfall та Agile методологій. Обидві моделі мають свої переваги та недоліки. Невеликі проекти завжди підходять для Agile підходу і майже ніколи для підходу Waterfall. Великий і складний проект з декількома командами, що працюють одночасно над різними частинами програми, майже завжди є проектом Waterfall. Така неоднозначність потребує зміти процедури в до підходу вибору технології розробки ПЗ. Питання вибору технології для розробки програмного забезпечення ми вирішуємо за допомогою аналізу думок експертів в галузі розробки та привівши їх оптимізацію за Парето. Концепція ПР БКЗПР полягає у свідомому виборі з множини альтернатив однієї. Цей вибір робить ОПР, яка прагне до досягнення своєї певної цілі. В ролі ОПР може виступати чи конкретний індивід, чи група осіб, яка одночасно приймає дане рішення.

В результаті запропоновано використання методу Парето оптимальності до звуження критеріїв за рахунок використання набору взаємно незалежної інформації про відносну важливість критеріїв. Певні аспекти застосування експертних методів багатокритеріальної ОПР потребують подальшої розробки. В майбутніх дослідженнях з теми пропоную зосередити увагу на забезпеченні повноти та узгодженості експертних даних та підвищенні рівня довіри до них з боку осіб, що приймають рішення (ОПР).

Список бібліографічного опису

1. Петерсон К. Модель водоспаду у великомасштабному розвитку; Кай Петерсон. - 386 с
2. Шор, Д. Мистецтво спритного розвитку, Джеймс 440, 440 с.
3. Марк Л. Agile Управління проектами для манекенів; Лейтон Марк. - Hoboken, NJ: Для чайників, 2012. - 360 с.
4. Бек К. Екстремальне програмування пояснено: обійняти зміну; К. Бек, К. Андрес. - 224 сек.
5. Bohem B. Модель спіральної інкрементальної відданості: принципи та практики для успішних систем та програмного забезпечення ;B. Bohem, D. Lane, S. - Kulmanojong, R. Turner, 2014. - 336 с.
6. Вільямс Л. Огляд методологій розвитку, орієнтованої на плани, 2004 р.
7. Вільямс Л. Огляд методів гнучкої розробки, 2007 р.
8. RabbitMQ: Вступ до AMQP [Інтернет-ресурс] - Режим доступу до ресурсів: <https://habrahabr.ru/post/64192/>.
9. Майк Кон. Scrum: Гнучка розробка програмного забезпечення = Успіх з Agile: Розробка програмного забезпечення за допомогою Scrum (серія підписів Addison-Wesley). - М.: Вільямс, 2011. - С. 576. - ISBN 978-5-8459-1731-7.
10. Офіційний веб-сайт AngularJS. - Режим доступу <https://angularjs.org>
11. Cohen D., Lindwal M., Costa P. Agile Розробка програмного забезпечення. Звіт про DACS SOAR, 11, 2003.
12. Андрій, Богдан-Олександр та ін. Дослідження водоспадних та спритних методів управління програмними проектами; Андрей, Б.А., Касу-Поп, А.С., Георге, С.С., Боянгіу, С. А - Журнал інформаційних систем та управління операціями (2019): 125-135

References

1. Peterson K. The Waterfall Model in Large-Scale Development ;Kai Peterson. - 386 p
2. Shore, D. The Art of Agile Development, James 440, 440 p.
3. Mark L. Agile Project Management For Dummies ; Leighton Mark. - Hoboken, NJ: For Dummies, 2012. - 360 p.
4. Beck K. Extreme Programming Explained: Embrace Change; K. Beck, K. Andres. - 224 sec.
5. Bohem B. The Incremental Commitment Spiral Model: Principles and Practices for Successful Systems and Software / B. Bohem, D. Lane, S. - Kulmanojong, R. Turner, 2014. - 336 p.
6. Williams L. A survey of plan-driven development methodologies, 2004.
7. Williams L. A survey of agile development methodologies, 2007.
8. RabbitMQ: Introduction to AMQP [Online Resource] - Resource Access Mode: <https://habrahabr.ru/post/64192/>.
9. Mike Cohn. Scrum: Flexible Software Development = Succeeding with Agile: Software Development Using Scrum (Addison-Wesley Signature Series). - М.: Williams, 2011. - P. 576. - ISBN 978-5-8459-1731-7.
10. AngularJS Official Website. - Access mode <https://angularjs.org>
11. Cohen D., Lindwal M., Costa P. Agile software development. DACS SOAR Report, 11, 2003.
12. Andrei, Bogdan-Alexandru, et al. A study on waterfall and agile methods in software project management./ Andrei, B. A., Casu-Pop, A. C., Gheorghe, S. C., & Boianu, C. A - Journal of Information Systems & Operations Management (2019): 125-135