

Кордунова Юлія СергіївнаАд'юнкт, orcid.org/0000-0003-0151-8285

Львівський державний університет безпеки життєдіяльності, Львів

Смотр Ольга ОлексіївнаКандидат технічних наук, доцент кафедри, orcid.org/0000-0003-2767-5019

Львівський державний університет безпеки життєдіяльності, Львів

Кокотко Ігор ЯрославовичWeb UI Software Engineer, orcid.org/0000-0002-8665-5782

SoftServe, Львів

Малець Романна БогданівнаКандидат фізико-математичних наук, доцент кафедри, orcid.org/0000-0002-9272-0958

Львівський Національний університет імені Івана Франка, Львів

АНАЛІЗ ТРАДИЦІЙНОГО ТА ГНУЧКОГО ПІДХОДІВ ДО СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ДИНАМІЧНИХ УМОВАХ

***Анотація.** Описано принципи реалізації, переваги та недоліки традиційної (Waterfall) та гнучкої (Agile) методології управління IT-проєктом. Здійснено акцент на застосуванні цих методологій на реальних проєктах, що реалізуються в динамічному оточенні. Проаналізовано роль окремих процесів у формуванні проєкту з використанням різних методологій управління. Реалізовано візуалізацію процесу розроблення програмного забезпечення з використанням традиційної та гнучкої методології, що допомогло акцентувати специфіку їх застосування у динамічному оточенні. У ході аналізу зроблено акцент на актуальності цих методологій в умовах динамічного середовища. Проведено детальний аналіз «Маніфесту гнучкої розробки програмного забезпечення» та його основних положень при плануванні та реалізації проєктів з розроблення програмного забезпечення. Зважаючи на особливості розробки програмного забезпечення у динамічному оточенні та отримані результати аналізу, обґрунтовано неспроможність традиційних методологій (Waterfall) забезпечувати належний рівень ефективності на різних етапах життєвого циклу. Натомість окреслено основні переваги застосування методології Agile у розробці програмного забезпечення та її адаптивність до постійних змін до вимог, чисельності команди тощо.*

***Ключові слова:** програмне забезпечення; традиційні методології (Waterfall); гнучкі методології (Agile); трикутник проєкт-менеджменту; беклог; власник продукту; мінімально життєздатний продукт (МЖП)*

Постановка проблеми

Питання важливості методологій розроблення програмного забезпечення гостро назріло із шаленим темпом розвитку та створенням нових інформаційних продуктів у сучасному суспільстві. Кожного дня ініціалізуються нові проєкти, створюються програмні продукти та програми, які потребують належного управління. Станом на сьогодні маємо велике різноманіття методологій управління проєктами. Переважно виокремлюють два кардинально різні підходи до управління: традиційні (Waterfall) та гнучкі (Agile) методології управління проєктами. Вибір підходу, згідно з яким слід здійснювати управління визначеним проєктом, залишається за командою та замовником.

Аналіз останніх досліджень і публікацій

Аналіз останніх досліджень і публікацій свідчить, що питання вибору методології управління IT-проєктом є актуальним як для українських, так і зарубіжних вчених. Із появою «Маніфесту розробки програмного забезпечення» [1] все більше праць було присвячено Agile методології та порівнянню її із традиційними методологіями управління проєктами. Зокрема теоретичні основи Agile висвітлені у роботах Роба Коула, Едварда Скотчера, Роберта Мартіна. В Україні питанням управління IT-проєктами займалися такі вчені, як С. Д. Бушуєв, І. Ю. Лебедева, С. В. Леонова, І. М. Якубенко, І. І. Оберемок та інші.

Мета статті

Мета – підвищити рівень ефективності вибору методології розроблення програмного забезпечення в динамічному оточенні шляхом відтворення основних характеристик їх застосування на підставі аналізу традиційного (Waterfall) та гнучкого (Agile) підходів.

Виклад основного матеріалу

Waterfall належить до традиційних методологій управління проектами, яким притаманний чіткий і послідовний процес розроблення програмного забезпечення. Незважаючи на те, що на сьогодні така модель майже не застосовується у сфері реалізації ІТ-проектів, вона є дуже важливою, адже на ній базуються усі інші методології розроблення програмного забезпечення.

Традиційна методологія поділяє життєвий цикл на деякий набір фаз, кожна з яких розпочинається лише після завершення попередньої (рис. 1).



Рисунок 1 – Життєвий цикл моделі Waterfall

Такий підхід до розробки є доволі простим і прозорим, проте він занадто ідеалістичний і не практичний в умовах динамічного оточення. Модель Waterfall, зазвичай, використовують у проектах із чітко визначеними вимогами, що не змінюються протягом реалізації проекту. Здебільшого це проекти машинобудівної інфраструктури, або державні проекти, які вимагають чіткого документування, розрахунку бюджету та аналізу всіх можливих ризиків. Якщо брати до уваги ІТ-індустрію, то розроблення програмного продукту за моделлю Waterfall буде доречною, якщо мова йтиме про однотипні інформаційні системи (ІС), ІС із складними обчисленнями, ІС, що працюють у реальному часі, або ІС, до яких чітко та повною мірою сформульовано усі вимоги на стадії планування.

До переваг такої моделі належать:

- зрозуміла та чітка послідовність розроблення програмного забезпечення, що своєю чергою знижує поріг входження команд на проект;

- стабільність, оскільки на етапі затвердження вимог та документації проект стає незмінний, протягом усього процесу розроблення;

- можливість оцінити вартість та часові рамки ще до початку розроблення проекту;

- можливість легко відслідкувати процес впровадження проекту, матеріальні ресурси завдяки чіткій документації та суворій поетапності процесу розроблення.

Як результат отримуємо вчасно розроблений в межах бюджету цілісний програмний продукт (ПП) з документацією на нього. Такий ПП дуже зручний для впровадження та супроводу.

Проте, як вже зазначалось, цей підхід до розроблення програмного забезпечення в умовах сучасного динамічного середовища в більшості випадків є не достатньо ефективним. Найпоширенішими проблемами процесу розроблення на основі традиційних методів є:

- неможливість зміни вимог безпосередньо під час розроблення;

- відсутність чіткого розподілення обов'язків за виконану роботу та її результат;

- наявність безперервного потоку «дрібних» додаткових завдань, які відволікають розробників і менеджерів від основної роботи;

- як висновок – невідповідність часовим рамкам, збільшення бюджету, втрата якості.

Проблема традиційного підходу до реалізації проектів розроблення програмного забезпечення в тому, що виконання триває довше, ніж очікувалось, витрати виявляються більшими, ніж закладалось у бюджеті і часто не досягають очікуваних результатів. Традиційні підходи до проект-менеджменту беруть за основу співвідношення часу, витрат та обсягу робіт, що формують так званий «трикутник проект-менеджменту» (рис. 2), який у своїй книзі «Brilliant agile project-management» Роб Коул та Едвард Скотчер називають «Бермудський трикутник».

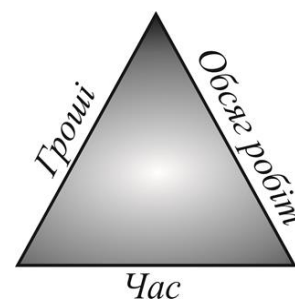


Рисунок 2 – Трикутник проект-менеджменту

Через жорсткі зв'язки, які лежать в основі цього трикутника, неможливо усунути одну з його сторін, не вплинувши на інші. Зміна будь-яких параметрів завжди матиме наслідки під час реалізації проекту. Здебільшого це відбувається тоді, коли вносяться зміни до проекту, скорочується час або бюджет.

Проект-менеджерам дуже важко зберігати ці три складові в рівновазі і при цьому забезпечувати всі бажання клієнта. У результаті ми отримуємо або неякісний продукт, випущений вчасно у межах свого бюджету, або дійсно вартісний результат, ціна якого значно перевищуватиме початкову, а на часові проміжки вже не звертатимуть увагу.

Під час розроблення продукту з використанням гнучкої методології важливим є лише одне – якість продукту. Методологія Agile відходить від традиційної одержимості термінами та бюджетами, зосереджуючись передусім на тому, чого хоче клієнт, або – чого він дійсно потребує [8, с. 247].

Рух Agile бере початок з концепції Lean – техніки «ощадливого виробництва», яка широко застосовується в автомобільній індустрії. Спершу методологію гнучкого управління використовували в галузі інформаційних технологій (IT). Тому не дивно, що саме у лютому 2001 р. сімнадцятьма фахівцями, які зібрались на гірськолижному курорті The Lodge at Snowbird у штаті Юта, щоб обговорити принципи розроблення програмного забезпечення, був опублікований «Маніфест гнучкої розробки програмного забезпечення», який раз і назавжди змінив підходи до процесу розроблення та створення нових проектів не тільки у галузі IT, а і у всіх інших сферах життєдіяльності.

«Маніфест гнучкої розробки програмного забезпечення» твердить, що [1]:

– люди та взаємодія важливіші за процеси та інструменти. Agile робить акцент на людській комунікації, командній роботі та розвитку розробників. Він говорить, що насамперед у команді має бути дух співпраці та взаємодопомоги. А дотримання жорстких правил і процесів швидше пасуватиме одинакам і диктаторам, які звикли працювати в одному стилі та не бажають змінюватись;

– працююче програмне забезпечення важливіше за вичерпну документацію. На думку Agile-спеціалістів, не варто витратити дорогоцінний час та гроші на написання документації. Набагато краще втілити його у розроблення готового функціоналу продукту. Це допоможе замовнику набагато швидше зрозуміти цінність і перспективи розвитку проекту;

– співпраця з клієнтом набагато важливіша за обговорення умов контракту. Гнучка методологія Agile приділяє велику увагу спілкуванню із замовником. Нерідко буває так, що клієнт сам до кінця не знає, які саме функції має реалізувати продукт та який термін і бюджет потрібні для його виконання. Дуже важко наперед передбачити всі умови створення проекту. Тому саме тісна співпраця замовника із командою допомагає випустити дійсно вартісний продукт, який задовольнятиме всі його потреби;

– готовність до змін важливіша за дотримання плану. Agile-методологія вважає зміни невід’ємною частиною розвитку хорошого проекту. Втілені під час розроблення ідеї можна реалізувати значно швидше, а отже, і тестувати їх можна набагато раніше.

Отже, під час створення проектів важливо зрозуміти одне – замовник не хоче кращого управління. Він хоче кращого кінцевого продукту. Методологія гнучкого управління Agile спрямована на це [8, с. 248]. Не важливо, які саме техніки та процеси ви використовуєте для досягнення кращого результату, не потрібно зосереджуватись на самих методах. Результат важливіший за шляхи, якими його досягнуто.

Розглянемо, чим відрізняється процес створення продукту на основі гнучких методологій від традиційних. Почнемо з того, що більшість гнучких методологій спрямовані на те, щоб мінімізувати ризики шляхом ітеративної розробки проекту (рис. 3).



Рисунок 3 – Життєвий цикл Agile-методології

Кожна ітерація це мініпроект, який включає в себе всі процеси розроблення починаючи від планування і завершуючи випуском готової функціональності. Зазвичай вони тривають два – три тижні (все залежить від того, який темп бере команда розробників) і закінчуються ретроспективою, тобто обговоренням того, що було зроблено та що можна зробити краще.

На стадії планування проекту Agile починає із визначення необхідного мінімуму й працює вже з ним. Цей мінімум так і називається – мінімально життєздатний продукт (minimum viable product, MVP), або мінімальний набір функціональності (minimum feature set, MFS). На практиці мінімально життєздатний продукт вже відповідає основним бізнес-вимогам проекту і може бути проданий. Такий підхід зменшує часові та матеріальні витрати,

необхідні на створення проєкту. Він може стати основою для більш складного і функціонального бізнес-рішення, до того ж його набагато краще і легше тестувати. А підсумки тестування виявляються більш продуктивними, адже ґрунтуються на реальних фактах [19, с. 166].

Не менш важливим є залучення розробника на всіх стадіях розроблення проєкту. В Agile прийнято у кожному команду включати одну людину зі сторони бізнесу. Таку людину зазвичай називають власником продукту. Власник продукту представляє інтереси бізнесу і кінцевого користувача, він точно знає що саме потрібно людям, а не як це реалізувати. Основна його функція – донести до команди завдання, які мають бути реалізовані з точки зору бізнесу, а команда, своєю чергою, реалізує це з технічної сторони. Для максимального ефекту замовник і команда розробників завжди мають працювати в парі. Тільки так можна створити дійсно якісний продукт, який задовольнятиме всі бажання клієнта.

Після того як визначене практичне бачення проєкту, команді потрібно якомога детальніше записати вимоги до продукту. Такий перелік вимог у Agile називають беклогом. На відміну від звичного для традиційних моделей технічного завдання, беклог формує список важливих ідей для бізнесу. В цьому підтверджується орієнтація гнучких методологій на результат, а не на процеси. Команда бачить беклог і вже після того вирішує, які дії (тобто технічні рішення) їй потрібно виконати для задоволення конкретної вимоги, проте виконання лише технічного завдання (формування якого властиве для традиційних моделей) не завжди означає одержання бажаного продукту.

Отже, до переваг використання Agile-методології належить:

- мінімізація ризиків завдяки гнучкій системі внесення змін;
- робота на результат (вкінці кожної ітерації замовник отримує готовий продукт);
- зменшення документації (за рахунок живого спілкування);
- можливість змінювати проєкт у ході розроблення.

Попри все, гнучкі методології розроблення програмного забезпечення мають свої недоліки. Зокрема це:

- складність підрахунку кінцевої суми та часового проміжку проєкту;
 - підвищені вимоги до кваліфікації і досвіду команди;
 - постійні зміни, які можуть призвести до того, що проєкт ніколи не дійде до фінальної версії.
- Порівняння традиційної методології (Waterfall) та гнучкої методології (Agile) наведено у таблиці.

Таблиця – Порівняння традиційної методології (Waterfall) та гнучкої методології (Agile)

Опис	Waterfall	Agile
Робота над проєктом чітко спланована та послідовно виконується	так	ні
Внесення змін до проєкту в процесі розроблення	ні	так
Сталий темп процесу розроблення	ні	так
Тестування ПП у процесі розроблення	ні	так
Управління директивне	так	ні
Контроль за роботою команди, що працює над розробкою	так	ні
У команді можлива взаємозаміна, розподілене лідерство	ні	так
Команда співпрацює в процесі розроблення із замовником	ні	так

Узагальнюючи вищевикладене можемо стверджувати, що при організації життєвого циклу проєкту згідно з гнучкими методологіями розробниками значно швидше буде здійснено представлення замовнику прототипу ПП, його тестування та адаптація даного ПП до мінливих потреб ринку та вимог замовника і, як результат, значно швидша доставка готового ПП на ринок.

Висновки

Зважаючи на особливості розроблення програмного забезпечення у динамічному оточенні та отриманні результату аналізу, зроблено висновки про неспроможність традиційних методологій (Waterfall) забезпечити належний рівень ефективності на різних етапах життєвого циклу. Модель Waterfall має низку недоліків, які роблять її повністю непридатною, або призводять до несвоєчасного виконання визначеного обсягу робіт, перевитрат бюджету та недотримання встановленого рівня якості продукту.

Натомість окреслено основні переваги застосування методології Agile у розробці програмного забезпечення та її адаптивність до постійних змін у вимогах, чисельності команди розробки, бюджету тощо. А все тому, що гнучкість – це основна перевага розробника на ринку. Лише ті команди, які можуть іти в ногу з часом, які працюють на результат та на задоволення будь-яких потреб замовника будуть мати місце на сучасному ринку праці. Саме такі команди і пропагують Agile-методологію.

Список літератури

1. Agile-маніфест розробки програмного забезпечення. URL: <https://agilemanifesto.org/iso/uk/manifesto.html>.
2. A guide to the Project Management Body of Knowledge. PMBOK guide SIXTH EDITION – USA: *Project Management Institute*, 2017.
3. Близнюкова І. О., Семко С. Г., Кійко С. Г. Огляд сучасних методологій управління командами ІТ-проектів. *Управління розвитком складних систем*. Київ, 2020. № 43. С. 60 – 66. <https://doi.org/10.32347/2412-9933.2020.43.60-66>.
4. Колянко О. В., Озимок Г. В., Використання жорсткої "Waterfall" та гнучкої "Agile" моделей управління проектами. *Вісник Львівського торговельно-економічного університету. Економічні науки*. Львів, 2017. Вип. 52. С. 177 – 182.
5. Вавіленкова А. І. Аналіз гнучких методологій розробки програмного забезпечення для реалізації у командних проєктах. *Вісник Національного технічного університету «ХПИ»*. Харків, 2021. № 1(7). С. 39 – 46. <https://doi.org/10.20998/2413-4295.2021.01.06>.
6. Кім О. О., Козлова В. В. Перспективи застосування методології Agile менеджменту в управлінні ІТ-проектами. *Соціальна економіка*. Харків. 2019. № 58. С. 95 – 99. <https://doi.org/10.26565/2524-2547-2019-58-12>
7. Кордунова Ю. С., Придатко О. В., Смотри О. О. Переваги використання Agile-методології під час розробки програмного забезпечення в умовах сучасного ринку. *Інформаційна безпека та інформаційні технології : зб. наук. праць ІV Всеукр. наук.-практ. конф. молодих учених, студентів і курсантів*. м. Львів 27 листопада 2020 р. Львів, 2020. С. 206 – 207.
8. Кордунова Ю. С., Смотри О. О. Сенс Agile-маніфесту для сучасного проєкт-менеджменту. *Проблеми та перспективи розвитку системи безпеки життєдіяльності: зб. наук. праць XVI Міжнар. наук.-практ. конф. молодих вчених, курсантів та студентів*. – Львів: ЛДУ БЖД, 2021. С. 247-248
9. Приймак В. Гнучкі моделі управління командною роботою інжинірингових проєктів. *Вісник Київського національного університету мени Тараса Шевченка. Економіка*. Київ, 2019. №6 (207). С. 21-27. <https://doi.org/10.17721/1728-2667.2019/207-6/3>.
10. Якубенко І. М. Agile-менеджмент, як дієве управління проектами для цілеспрямованих команд. *Економіка. Менеджмент. Бізнес*. 2017. №4(22). С. 167 – 172.
11. Муравецький С. А., Крамський С. О. Планування процесів забезпечення якості у великих та географічно розподілених гібридних ІТ-проєктах. *Вісник НТУ «ХПИ»*. Харків, 2016. №1(1173). С. 106 – 109. <https://doi.org/10.20998/2413-3000.2016.1173.21>.
12. Асєєва А. В., Кулаковська І. В. Аналіз проблем вибору технології для розробки програмного забезпечення. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*. Луцьк, 2019. № 37. С. 10 – 18.
13. Бушуєв С.Д., Бушуєва В. Б., Бойко О. О. Agile- трансформація підходів в управлінні будівельними проєктами, фазах ініціалізації та проєктування. *Управління розвитком складних систем*. Київ, 2020. №41. С. 14 – 20. <https://doi.org/10.32347/2412-9933.2020.41.15-20>.
14. Гидроец М. О., Гришанова Л. И. Методологии разработки программного продукта. *Системный анализ и логистика*. Санкт-Петербург, 2020. № 4 (26). С. 45 – 52. <https://doi.org/10.31799/2007-5687-2020-4-45-53>.
15. Vargaood S. O., Mohd H., Baharom F. A Comparison Study of Software Testing Activities in Agile Methods. *Knowledge Management International Conference (KMICe) Virtual Conference*. Malaysia, 2021, pp. 130–137.
16. Семенов С. Г., Халифе Кассем, Захарченко М. М. Усовершенствованный способ масштабирования гибкой методологии разработки программного обеспечения. *Вісник НТУ «ХПИ»*. Харків, 2017. Т. 1, № 1. С. 79 – 84. <https://doi.org/10.20998/2522-9052.2017.1.15>.
17. Шапошнікова О. П., Кірвас В. В. Застосування методології Agile в практиці проєктного навчання при підготовці ІТ спеціалістів. *Системи обробки інформації*. Харків. 2020. № 4(163). С. 94-100. <https://doi.org/10.30748/soi.2020.163.10>
18. Козир І. С. Фактори впровадження Agile-менеджменту в практику управління. *I International Scientific and Practical Conference «Problemas y perspectivas de la aplicación de la investigación científica innovadora»*. Кембридж. 2021. Т. 1. С. 78-79. <https://doi.org/10.36074/logos-19.03.2021.v1.26>.
19. Придатко О. В., Солотвінський І. В., Кокотко І. Я., Івановський М. Я. Модель портфельного управління проектами розвитку регіональних систем безпеки життєдіяльності. *Управління розвитком складних систем*. Київ. 2018. № 36. С. 42–50.
20. Кордунова Ю. С., Смотри О. О. Визначення ефективності використання Agile методології в сучасних організаціях. *Проблеми та перспективи забезпечення цивільного захисту: матеріали міжнародної науково-практичної конференції молодих учених*. Харків: НУЦЗУ, 2021. С. 166.
21. Islam G., Stoner T. A case study of agile software development for safety-Critical systems projects. *Reliability Engineering & System Safety*. Vol. 200. 2020. <https://doi.org/10.1016/j.ress.2020.106954>
22. Stellman A., Greene J. Learning Agile: Understanding Scrum, XP, Lean, and Kanban. 1st Edition, USA: O'Reilly Media, 2013. 420 с.
23. Stioca M., Ghlic-Micu B., Mircea M., Uscatu C. Analyzing Agile Development – from Waterfall Style to Scrumban. *Informatica Economică*. 2016. №4. С. 5–14. <https://doi.org/10.12948/issn14531305/20.4.2016.01>
24. Cole R., Scotcher E. Brilliant Agile Project Management: A Practical Guide to Using Agile, Scrum and Kanban. Edinburg: Pearson, 2015. 187 p.
25. Papadopoulos G. Moving from traditional to agile software development methodologies also on large, distributed projects. *Procedia – Social and Behavioral Sciences*. 2015. № 175. pp. 455 – 463. <https://doi.org/10.1016/j.sbspro.2015.01.1223>

Стаття надійшла до редколегії 23.05.2021

Kordunova YuliiaDoctoral Student, orcid.org/0000-0003-0151-8285

Lviv State University of Life Safety, Lviv

Smotr OlgaPh.D, Senior Lecturer of the Department, orcid.org/0000-0003-2767-5019

Lviv State University of Life Safety, Lviv

Kokotko IgorWeb UI Software Engineer, orcid.org/0000-0002-8665-5782

SoftServe, Lviv

Malets RomannaPhD (Physics and Mathematics), Senior Lecturer of the Department, orcid.org/0000-0002-9272-0958

Ivan Franko National University of Lviv, Lviv

**ANALYSIS OF THE TRADITIONAL AND FLEXIBLE APPROACHES
TO CREATING SOFTWARE IN DYNAMIC CONDITIONS**

Abstract. The success of the project depends on the quality of management, in particular, the management methodology chosen for the project. When it comes to developing new software, the development team and the customer are faced with the methodology choice and approach to the development of this software. The purpose of this work is to explore and analyze two completely different approaches to software development – the traditional (Waterfall) and the flexible (Agile). In the article are described implementation principles, advantages, and disadvantages of traditional (Waterfall) and flexible (Agile) methodology. It is made an emphasis on using these methodologies on real projects are made. Also, we analyzed the so-called “Project Management Triangle” and its role in project formation using different management methodologies. A visual representation of the software development process using these two methodologies was embedded to better understanding the specifics of each. During the analysis, conclusions were made regarding the relevance of methodologies in a modern, dynamic environment. Detailed description and analyze of Manifesto of Flexible Software Development, which changed people's perception of the process of developing and creating new projects not only in the field of IT but also in all other spheres of life, were conducted. As a result, were made conclusions about the inability of traditional methodologies (including the Waterfall model) to meet modern market requirements and was preferred Agile methodology, which enables customer requirements to be met in a dynamic environment, ready to changes and promote them.

Keywords: software; traditional methodologies (Waterfall); flexible methodologies (Agile); Project Management Triangle; backlog; product owner, minimum viable product (MVP)

References

1. Kordunova, Yu. S. & Smotr, O. O. (2021). The meaning of Agile-manifest for modern project management. *Problems and prospects of life safety: collection of scientific papers XVI International Scientific and Practical Conference of Young Scholars, Cadets and Students*, pp 247-248. (in Ukr.).
2. Pryimak, V. (2019). Agile management of teamwork of engineering projects. *Bulletin of Taras Shevchenko National University of Kyiv*, Kyiv, 6(207), 21-27 (in Ukr.). <https://doi.org/10.17721/1728-2667.2019/207-6/3>.
3. Yakubenko, I. (2017). Agile-management as an effective project management for targeted teams. *Economics. Management. Business*, 4(22), 167–171 (in Ukr.).
4. Muravetskiy, S. A. & Kramskiy, S. O. (2016). Planning Quality Assurance processes in a large scale geographically spread hybrid software development project. *Bulletin of National Technical University "KhPI"*, 1(1173), 106-109 (in Ukr.). <https://doi.org/10.20998/2413-3000.2016.1173.21>
5. Asieieva, A. V. & Kulakovska, I. V. (2019). Analysis of technology selection problems for software development. *Computer-Integrated Technologies: Education, Science, Production*, 37, 10-18 (in Ukr.).
6. Bushuyev, S., Bushuyev, D., Bushueva, V. & Boyko, O. (2020). Agile transformation of approaches in the management of building projects of the initiation and design phases. *Management of Development of Complex Systems*, 41, 14–20. (in Ukr.). <https://doi.org/10.32347/2412-9933.2020.41.15-20>.
7. Gidroets, M. O. & Grishanova L. I. (2020). Software development methodology. *System analysis and logistics*, 4(26), 45–52. (In Russ.) <https://doi.org/10.31799/2007-5687-2020-4-45-53>.
8. Barraood, S. O., Mohd, H. & Baharom, F. A. (2021). Comparison Study of Software Testing Activities in Agile Methods. *Knowledge Management International Conference (KMICe) Virtual Conference*, pp. 130–137.
9. Semenov, S., Kassem, Khalifeh and Zakharchenko, M. (2017). Advanced method of scaling the flexible methodology of software development. *Bulletin of National Technical University "KhPI"*, 1, 1, 79-84 (in Russ.). <https://doi.org/10.20998/2522-9052.2017.1.15>.
10. Shaposhnikova, O. & Kirvas, V. (2020). Application of the Agile methodology in the practice of project-based learning in the training of IT specialists. *Information Processing Systems*, 4(163), 94-100. (In Ukr.) <https://doi.org/10.30748/soi.2020.163.10>.

11. Prydatko, O., Solotvynskyi, I., Kokotko, I. & Ivanovskyi, M. (2018). Model of portfolio management by development projects of regional life safety systems. *Management of Development of Complex Systems*, 36, 42–50. (in Ukr.).
 12. Kozyr, I. (2021). Factors of Agile management implementation in management practice. *I International Scientific and Practical Conference «Problemas y perspectivas de la aplicación de la investigación científica innovadora»*, 1, pp. 78-79 (In Russ.) <https://doi.org/10.36074/logos-19.03.2021.v1.26>.
 13. Kordunova, Yu. S. & Smotr, O. O. (2021). Determining the effectiveness of using Agile methodology in modern organizations. *Problems and prospects of civil protection: materials of the international scientific-practical conference of young scientists*, 166. (In Ukr.).
 14. Islam, G. & Stoner, T. (2020). A case study of agile software development for safety-Critical systems projects. *Reliability Engineering & System Safety*, 200. <https://doi.org/10.1016/j.ress.2020.106954>.
 15. Stellman, A. & Greene, J. (2013). *Learning Agile: Understanding Scrum, XP, Lean, and Kanban 1st Edition*, USA: O'Reilly Media.
 16. Stioca, M., Ghlic-Micu, B., Mircea, M. & Uscatu, C. (2016). Analyzing Agile Development – from Waterfall Style to Scrumban. *Informatica Economică*, 4, 5–14. <https://doi.org/10.12948/issn14531305/20.4.2016.01>.
 17. Cole, R. & Scotcher, E. (2015). *Brilliant Agile Project Management: A Practical Guide to Using Agile, Scrum and Kanban*. Edinburg: Pearson.
 18. Papadopoulos, G. (2015). Moving from traditional to agile software development methodologies also on large, distributed projects. *Procedia – Social and Behavioral Sciences*, 175, 455–463. <https://doi.org/10.1016/j.sbspro.2015.01.1223>.
-

Посилання на публікацію

- APA Kordunova, Yu., Smotr, O., Kokotko, I. & Malets, R. (2021). Analysis of the traditional and flexible approaches to creating software in dynamic conditions. *Management of Development of Complex Systems*, 47, 71–77, [dx.doi.org\10.32347/2412-9933.2021.47.71-77](https://doi.org/10.32347/2412-9933.2021.47.71-77).
- ДСТУ Кордунова Ю. С., Смотр О. О., Кокотко І. Я., Малець Р. Б. Аналіз традиційного та гнучкого підходів до створення програмного забезпечення в динамічних умовах. *Управління розвитком складних систем*. Київ, 2021. № 47. С. 71 – 77, [dx.doi.org\10.32347/2412-9933.2021.47.71-77](https://doi.org/10.32347/2412-9933.2021.47.71-77).