

Кіндрат О.В.

*к. е. н., доцент кафедри інформаційних технологій у менеджменті,
факультет економіки та менеджменту
Львівський національний університет
ветеринарної медицини та біотехнологій імені С.З. Гжицького
<https://orcid.org/0000-0003-0820-2806>*

Дутка Г.І.

*асистент,
кафедра інформаційних технологій у менеджменті,
факультет економіки та менеджменту
Львівський національний університет
ветеринарної медицини та біотехнологій імені С.З. Гжицького
<https://orcid.org/0000-0001-5834-9043>*

AGILE-МЕТОДИ ДЛЯ ЕФЕКТИВНОЇ ТА ПРОДУКТИВНОЇ ІМПЛЕМЕНТАЦІЇ ІТ-ПРОДУКТУ

**JEL Classification: M1, M2
SECTION «ECONOMICS»: Економіка**

Анотація. Стаття акцентує увагу на найвагоміших перевагах гнучкого підходу у методах розробки програмного забезпечення. Розглядаючи цінності, засади та практики відомих методологій, що підтримують принципи гнучкої розробки програмного забезпечення, у роботі визначено поняття Agile; окреслено власне бачення, чим визначається і вимірюється гнучкість; визначено, які фактори у процесі розробки програмного забезпечення можна контролювати завдяки гнучкості. Крім того, дається характеристика відомих методологіям, що існують та застосовуються у сфері ІТ-менеджменту у переході до Agile. У ході нашої роботи ми дійшли висновку, що команди-розробники, при виборі однієї з гнучких методологій для створення програмного забезпечення, активно оцінюють та контролюють динамічну напруженість та компроміси між гнучкістю розробки програмного забезпечення та ефективністю її розробки, відповідаючи на запит зміни вимог. Отже, при аналізі методологій гнучкої розробки програмного забезпечення, наголошено на їх сильних та слабких сторонах.

Ключові слова: Agile, agile-методи, методології гнучкої розробки програмного забезпечення, гнучкість, ІТ-менеджмент, компанія-розробник, Scrum, eXtreme Programming, DSDM, Adaptive software development, Feature Driven Development, Crystal, Lean Software Development.

Annotation. One of the most significant advantages of Agile software development is the focus. Considering the values, principles and practices of known methodologies that support the principles of Agile software development, the paper defines the concept of Agile; outlines one's vision which determines and measures Agile; identified which factors in the software development process can be controlled through Agile principles. We decrypted the known methodologies, which are prevalent in IT management during the transition to agility. In the course of our work, we concluded that development teams, when choosing one of the agile methodologies for software development, actively assess and monitor the dynamic tension and trade-offs between software development agility and efficiency of its development, responding

to requests for changes in requirements. We analysed different methodologies of agile software development as Scrum, eXtreme Programming, DSDM, Adaptive software development, Feature Driven Development, Crystal, and Lean Software Development, emphasising their strengths and weaknesses. We see that different methodologies require somewhat different approaches in management culture and software development. For a software company, preparing to work on a project is essential before determining which methodology they will choose to achieve the efficiency and productivity of software development. These efficiency and productivity factors are timely completion of the project, budget development, and the software's successful functionality. Among the Agile methodologies, the most popular in the IT field are those methodologies that allow them to be used for the process of efficient and productive development of complex and reliable software systems. However, the choice of methodology remains unresolved. It is exciting and promising to make a more thorough analysis of each of the methodologies in comparing the choice of real projects that determine the new perspectives of this study.

Key words: Agile, agile-methods, methodologies of flexible software development, agility, IT-management, IT company, Scrum, eXtreme Programming, DSDM, Adaptive software development, Feature Driven Development, Crystal, Lean Software Development.

Вступ

Через зростаючу потребу в управлінні системами за більш короткі терміни і з меншими витратами та задовільною якістю, інженерія програмного забезпечення зазнала значних змін у власних методах і практиках. Створені через потребу гнучкості, agile-методи значно відійшли від традиційних планових підходів, зосередившись на наданні окремих частин продукту, на відміну від інших, що пред'являли готову програму лише у кінці процесу розробки.

Термін Agile з'явився у 2001 році, коли 17 представників компаній-розробників, що вже використовували технології-попередники у своїй роботі: Extreme programming, Scrum, DSDM, Adaptive software development, Crystal Clear, Feature driven development, Pragmatic Programming і розробили маніфест гнучкої розробки [8]. Вже станом на 2008 рік за результатами одного з опитувань стало зрозуміло, що більше половини відібраних команд-розробників програмних продуктів використовують agile-методи [14, с. 2], бо проекти, впроваджені за допомогою гнучких підходів реалізуються набагато швидше, ніж проекти за іншими типами моделей [7]. Ще новіші дослідження лише підтверджують тенденцію до зростання ступеня гнучкості [6].

Сьогодні практично не залишилося сумнівів, що agile-методи – ефективний спосіб створення програмного забезпечення, а команди, які використовують цю технологію, однозначно досягають покращення, а іноді й галопоподібних успіхів у своїх результатах.

Особливістю гнучких методів програмного забезпечення є те, що вони стоять на засадах постійного пошуку досконаліших методів розроблення програмного забезпечення, досягаючи успіху. Доцільністю використання гнучких методів є те, що на даний час відсутні дані про провальні agile-проекти, а потужні компанії-розробники продовжують активно їх запроваджувати у свою діяльність. Питання їх впровадження сьогодні є як ніколи актуальним і злободенним, адже зростають і вимоги до якості розроблюваного програмного забезпечення, що вимагає вдосконалення процесів їх розробки.

Народившись у контексті розробки програмного забезпечення, зараз agile-методи задіяні до майже будь-якого типу проектів у різних сферах діяльності, допомагаючи багатьом командам боротися з непередбачуваністю в рамках проекту шляхом поступових ітераційних циклів. Основною перевагою Agile Software Development є здатність допомагати командам розвиватися динамічно, зосереджуючись при цьому на забезпеченні високої якості продукту. Крім того, дуже важливою є співпраця, оскільки фахівцям потрібно працювати разом і розуміти свою конкретну роль у цьому процесі. Іншим важливим фактом є збереження високої якості, оскільки тестування

проводиться протягом усього процесу розробки, що дає достатньо часу, щоб зіткнутися та встановити помилки, які з'являються у кодї, та вчасно подати ці проблеми, щоб команда розробників виправила їх. Тоді команда досягає ефективності.

Фактично, ми перерахували найвагоміші переваги гнучкого підходу у методах розробки програмного забезпечення. Але, незважаючи на те, що окремі методології дійсно ефективні, популярні й використовуються дедалі більше, кожна з них має свої сильні та слабкі сторони, що спонукає нас розглянути їх ретельніше. Отже, метою нашої статті виступає проблема використання agile- методів у сфері IT-менеджменту, де основна мета полягає у тому, щоб постачати програмне забезпечення, а друга, не менш важлива, — знайти, налаштуватись на необхідну методологію. І хоч суть та принципи гнучких підходів управління проектами неодноразово розглядалися у науковій літературі, в результаті, багато організацій, застосовуючи їх, так до кінця і не розуміють, що означає використання гнучких методологій. Предметом статті є agile-методи (методології гнучкої розробки програмного забезпечення), об'єктом — цінності, засади та практики відомих методологій, що підтримують принципи гнучкої розробки програмного забезпечення, аналіз їх сильних та слабких сторін.

Виходячи із поставленої мети, перед нами стоїть декілька науково-дослідних завдань:

- визначити що таке гнучкість,
- окреслити, як визначається і вимірюється гнучкість;
- зазначити, які фактори можна контролювати завдяки гнучкості;
- охарактеризувати відомі методології, що існують та застосовуються у переході до гнучкості;
- дати аналіз найчастіше використовуваним, визначити їх сильні та слабкі сторони.

Інтернет-економіка суттєво змінила правила програмної інженерії. Традиційні методології розвитку були надто громіздкими, щоб відповідати швидкоплинним вимогам та коротким циклам продукції, яких вимагав бізнес. Сьогодні зростання потреби у гнучкості відображається у всіх секторах економіки, а дослідження Agile Software Development упродовж двадцяти років стали значною субдисципліною галузі розробки програмного забезпечення.

У квітні 2018 р. [14, с. 3] пошук за ключовими словами «Agile Software Development» в Google Scholar за період до 2001 р. давав трохи більше 13 000 результатів. Цей самий пошук привів до більш ніж 428 000 результатів за весь час досліджень сьогодні.

Ось теми, над якими активно працюють науковці світу, розглядаючи лише загальні питання гнучкої розробки програмного забезпечення: Маніфест і методика гнучкої розробки програмного забезпечення, інженерне забезпечення вимог до такого типу розробки програмного забезпечення, розвиток і оцінка зусиль у сфері гнучкої розробки програмного забезпечення. Найбільш популярними серед аналізованих методологій є Scrum [15] та Extreme programming [13].

Найновіші дослідження питання вивчення agile-методи у світі, датовані 2001 роком, становлять більш ніж 15 000 результатів, де основними напрямками вивчення виступають: порівняння методів широкомасштабної гнучкої розробки програмного забезпечення [3], стратегії управління ними, ролі керівника проєктів і синтезування досліджень з управління знаннями для гнучкої розробки програмного забезпечення. Велике значення приділяється загальнотеоретичним розвідкам, таким як: питання, виклики, теоретичне ядро гнучкої розробки програмного забезпечення [2], дослідження нефункціональних вимог [16], проблеми безпеки, перебої, що виникають в роботі команд Agile тощо. Розповсюдженими є дослідження щодо застосування agile-методологій у різних сферах промисловості, бізнесу, банківському секторі тощо.

Найцікавішими є питання, пов'язані з перспективами використання цих методологій. Так, в огляді, написаному Грегорі Коен, зазначено, що у майбутньому гнучкі методи можуть бути консолідовані «так само, як об'єднані в об'єктно-орієнтовані методи» [4]. Більш того, вчені вважають, що гнучкі методи «співіснуюватимуть» разом з традиційними, маючи симбіотичні

стосунки, де різні види факторів, такі як кількість людей, що працюють над проектом, інновації, визначатимуть, який процес обрати. Інше дослідження говорить про програмні процеси, що використовуються під час спроби впровадити або покращити гнучкість у компанії-розробнику, вже мутують на практиці у гібриди, і скоро ми зрозуміємо, чи ці технології все ще залишаються гнучкими? [6].

В Україні питома вага досліджень зосереджена здебільшого на огляді цих методологій [17], виборі методів розробки програмного забезпечення та їх оптимізації [9], вивченні світового досвіду запровадження agile-методологій [10]. Значна частина робіт присвячена аналізу підвищення конкурентоспроможності підприємств на основі впровадження гнучких методологій, приділяючи увагу впровадженню методів гнучкого програмного забезпечення у різних сферах промисловості, бізнесу, державного сектора, зокрема: організаціях транспортно-експедиторських компаній, кіноіндустрії тощо.

Тим не менше, серед наукових робіт найбільш прикладним напрямком досліджень досі залишається аналіз найпопулярніших методів гнучкої розробки програмного забезпечення. Сьогодні їх налічується чималий перелік, ось лише список найбільш задіяних: Scrum, eXtreme Programming (XP), Feature Driven Development (FDD), Method Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD), Crystal, Lean Software Development (LSD). У порівнянні з традиційними, вони мають однакові принципи та цінності, але різну практику [11, с. 73]. Scrum та eXtreme Programming приділено левову частку досліджень, тоді як іншим не надано такої уваги, що змушує нас звернути увагу на кожен із перелічених методологій та дати їм порівняльну характеристику.

Результати дослідження

Глобальна економіка розвивається швидкими темпами, насамперед, за допомогою інноваційних проектів, які, як правило, мають високий рівень невизначеності. Швидко реагувати на зміни та ризики, що виникають саме під час розробки ІТ-проектів, та безпосередньо адаптувати сам проект під потреби ринку або середовища, в якому відбувається його розробка, однозначно дозволяє використання гнучких чи то адаптивних методологій. Чи не найбільше цього потребує ІТ-галузь, підходи якої спрямовані на динамічне формування вимог та забезпечення їхньої реалізації за результатами постійної взаємодії всередині самоорієнтованих команд.

У своїй книзі Tore Dуба зазначив, «ми можемо розглядати гнучкі методи, як реакцію на планові або традиційні методи, які підкреслюють раціоналізований, інженерно-обґрунтований підхід, що включає широке планування, кодифіковані процеси та суворе повторне використання. Натомість, гнучкі методи наближаються до ідеї непередбачуваного світу, визнаючи ціннісні відносини між людьми, які можуть зробити внесок у розробку програмного забезпечення» [12]. Адже Agile – це тип мислення, в основу якого покладено цінності [8]. Agile визначає і принципи [1], якими керуються успішні команди, даючи вказівки, як створювати зміни та реагувати на них, а також як боротися з невизначеністю.

Хоч Agile вживають як загальний термін для переліку фреймворків та практик, що ґрунтуються на принципі гнучких методів розробки програмного забезпечення, це більше філософія, спосіб підходу до розробки програмного забезпечення. Якщо дати загальне визначення, ми можемо сказати, що Agile – це набір цінностей та принципів, які передбачають постачання якнайкращого програмного продукту клієнту з використанням адаптивного, поступового та ітеративного способу роботи за допомогою участі крос-функціональних та організованих команд.

Отже методологія Agile – це умови, яких команда вирішує дотримуватись таким чином, щоб вона відповідала цінностям та принципам Agile. Цінності та принципи Agile трансформовано у низку засад, що використовуються розробниками ІТ-продуктів [5; 6]. Взнявши за основу ці принципи, теоретично можна створити будь-який проект, який принесе якісні результати. Однак, це зовсім не так. Дослідження, проведене в Оксфордському університеті, показало, що ІТ-компанії в

20 разів більше схильні до ризику зазнати невдачі, ніж інші бізнес-проекти. Тому надважливим для компаній-розробників програмного забезпечення є передчасне виявлення й управління ризиками, причини яких знаходяться в невизначеності.

З огляду на це IT-компанії традиційно прагнуть застосовувати найсучасніші, найбільш ефективні технології у своїй роботі, обрав найбільш вдалу методологію для управління конкретного проєкту [5, с. 126]. Розглянемо найбільш використовувані сьогодні:

Scrum – гнучкий метод, що встановлює правила керування процесом розробки та дозволяє використовувати вже існуючі практики кодування, коректуючи вимоги чи вносячи зміни.

eXtreme Programming (XP) – популярна серед так-званих гнучких методологій тому, що здатна чутливо реагувати на зміни. Як вид методології дозволяє збільшити продуктивність праці та покращити можливості виконання вимог замовника.

Feature Driven Development (FDD) – функціонально-орієнтована розробка, що являє собою спробу об'єднати найбільш визнані в індустрії програмного забезпечення методики.

Method Dynamic Systems Development Method (DSDM) – заснований на концепції швидкої розробки додатків (Rapid Application Development, RAD), становить ітеративний і інкрементний підхід, який надає особливого значення участі у процесі користувача.

Adaptive Software Development (ASD) – методологія, побудована на концептуальній базі теорії складних адаптивних систем. Вона розрахована на використання в екстремальних проєктах, в яких переважають швидкий темп розробок, непередбачуваність й часті зміни.

Crystal — це не просто методологія, це сімейство методологій, де важливою особливістю виступає власне безперервне настроювання методології. Це дозволяє поступово адаптувати її для конкретної команди і визначеного проєкту.

Lean Software Development (LSD) – є доволі абстрактною методологією, проте ефективним підходом до управління, що скерований на оптимізацію бізнес-процесів за рахунок орієнтації на потреби замовника. Робота розбивається на невеликі частини, та розробка кожної частини відбувається незалежно та окремо.

Як ми вже зазначали, що IT-галузь потребує гнучких методологій для створення й впровадження нових продуктів, орієнтуючись саме на динамічні зміни вимог замовника, опираючись на можливість їх імплементації на основі постійної взаємодії незалежних робочих команд. Здатність команди відносно швидко та ефективно реагувати на зміни вимог користувачів визначається як її ефективність [5, с. 143]. Отже, ефективністю компанії-розробника програмного забезпечення може бути відповідь на питання: наскільки мінімальним може бути час, витрати, персонал та ресурси команди, що вимагають відреагувати на певну вимогу та включити їх у свої зміни. Аналогічно, ми можемо визначити, що ефективність вибору гнучкої методології стосовно створення програмного продукту — це здатність команди програмного забезпечення ефективно реагувати та включати вимоги та зміни користувачів протягом життєвого циклу проєкту. На нашу думку, саме від того, якою є незалежність і різноманітність команди програмного забезпечення, залежить ефективність її реагування. Натомість, ефективність реагування в свою чергу впливає на продуктивність розробки програмного забезпечення: своєчасне завершення, освоєння бюджету і вдала функціональність програмного забезпечення.

Розглянемо найбільш адаптовані до цього методи гнучкої розробки програмного забезпечення, відносно їх переваг у ефективності реагування на зміни.

Scrum – це гнучка система управління програмним забезпеченням, що сприяє ітераційному та поступовому розвитку. Незалежність її команд у ході роботи над програмним забезпеченням полягає у тому, що кожна з команд визначає особливості кожного спринту з еволюції накопичених продуктів; створює приріст потенційно доступного програмного забезпечення під час кожного спринту. Ефективність методології полягає у тому, що багатофункціональні команди самоорганізуються на різних етапах/спринтах; команди самі визначають, наскільки багато функцій у резерві вони хочуть взяти на себе під час наступного спринту.

XP методологія з самого початку створювалась як орієнтована власне на розробку програмного забезпечення, а її назва виникла від ідеї застосувати найбільш корисні методи і практики розробки програмного забезпечення. Отже, її переваги у специфіці розробки програмного забезпечення полягають у тому, що: найвищим пріоритетом є постійне задоволення мінливих потреб клієнтів; відбуваються швидкі огляди змін користувачів та відгуки на них. Ефективність забезпечується поєднанням повноважень команди і контролю відповідальності за виконання завдань; можливістю парного програмування – два розробники доповнюють навички та роботу один одного.

Функціонально-орієнтована методологія Feature Driven Development, зосереджена на найважливішій для замовника опції – функціонал розроблюваного програмного забезпечення. Тому ефективність роботи її команд полягають у тому, що всі ітераційні цикли орієнтовані на клієнта/функцію; регулярна відбувається інспекція з метою забезпечення оновлення систем. Продуктивність роботи забезпечується ефективністю маленьких, динамічно сформованих, автономних команд; а також тим, що до кожного дизайнерського рішення завжди застосовуються багатофункціональні точки зору.

Method Dynamic Systems Development Method – ітеративний і інкрементний підхід, що надає особливого значення участі у процесі користувача. Відповідно, у роботі команд визначальним є: ітераційний, поступовий розвиток, що керується відгуками користувачів; менш важливим фактором виступає створення ідеальної системи, ніж створення системи, яка відповідає поточним потребам бізнесу. Ефективність команд досягається у тому, що: команди мають повноваження приймати проєктні рішення, не чекаючи затвердження вищого рівня; відбувається постійна взаємодія та співпраця між усіма зацікавленими сторонами проєкту.

Результати цього аналізу свідчать про те, що команди-розробники, при виборі однієї з гнучких методологій для створення програмного забезпечення, динамічно оцінюють та контролюють складну напруженість та компроміси між гнучкістю розробки програмного забезпечення та ефективністю розробки програмного забезпечення. Відповідаючи на запит зміни вимог, вони оцінюють: вплив на проєкт, вплив на ризики часу/витрат/обсягу та технічні труднощі.

Враховуючи складні взаємозв'язки між напруженістю реагування, ефективністю реагування, автономністю команди, різноманітністю команди та продуктивністю при розробці програмного забезпечення, команди програмного забезпечення стикаються зі складними проблемами у виявленні та досягненні правильного балансу між двома вимірами гнучкості та продуктивності розробки програмного забезпечення. Для того, щоб проєкт ставу успішний, компаніям-розробникам програмного забезпечення, насамперед, доводиться визначати цілі продуктивності, час, вартість та функціональні можливості, які визначатимуть, наскільки необхідний вимір гнучкості, а отже, який вид agile-методів підходить більше.

Висновки

Світ інженерії програмного забезпечення постійно відкриває нові шляхи, які призводять до підвищення ефективності роботи ІТ-компаній і в той же час дають клієнтам задоволення користуватись чудово та якісно розробленим функціональним програмним продуктом. З появою гнучких методологій у розробці програмного забезпечення ці цілі розглядаються ІТ-компаніями все серйозніше. Натомість, бізнес-середовище, яке вимагає гнучких змін, не завжди розуміє, що за цим стоїть. Саме тому, у своїй роботі ми ставили за мету визначити, що таке гнучкість, а також окреслити, як визначається і вимірюється гнучкість.

Гнучкість (Agile) — це філософія, набір вказівок, яких дотримуються команди-розробники програмного забезпечення, щоб сягати суттєвих покращень в процесі розробки, досягнувши максимальної якості продукту і мінімізувавши витрати проєкту. Ці покращення — це те, чим вимірюється гнучкість і чим керується команда у процесі розробки програмного забезпечення: більш стабільні вимоги, більш раннє виявлення збоїв, менший час тестування, збільшення зв'язку та збільшення адаптаційної спроможності.

У ході нашої роботи ми дійшли висновку, що команди-розробники, при виборі однієї з гнучких методологій для створення програмного забезпечення, активно оцінюють та контролюють динамічну напруженість та компроміси між гнучкістю розробки програмного забезпечення та ефективністю її розробки, відповідаючи на запит зміни вимог. Вони оцінюють та приймають рішення щодо впливу змін: на проект у цілому, на ризики часу, ризики додаткових витрат, ризики зміни обсягу робіт та технічні труднощі. Таким чином, ми визначили ті фактори, які ІТ-компанії можуть контролювати завдяки методології гнучкості.

Водночас для компанії-розробника програмних продуктів важливим є процес підготування до роботи над проектом, перед тим, як визначити, яку методологію вони оберуть для досягнення ефективності та продуктивності розробки програмного забезпечення. Цими факторами ефективності та продуктивності є: своєчасне завершення проекту, освоєння бюджету і вдала функціональність програмного забезпечення.

Охарактеризувавши відомі agile-методології, що застосовуються у практиці, зауважено, що всі вони якнайкраще відповідають загальній філософії гнучкості. На основі порівняльного аналізу, нами зауважено деякі особливості кожної з методологій у плані досягнення ефективності та продуктивності. Адаже всі методології орієнтовані на: покращення якості продукції, на цінність бізнесу, цінність користувачів, готовність до змін, високий рівень прозорості та залучення замовника.

Однак, як бачимо, різні методології вимагають дещо відмінних підходів у культурі управління та програмного забезпечення розвитку, характеризуючи їх суттєві переваги як сильні сторони, визначаючи слабкі.

Scrum — використання цієї методології дає можливість виявляти та усувати відхилення від бажаного результату на більш ранніх етапах готовності програмного продукту. Вона якнайкраще підходить для вироблення ефективної управлінської культури постійного удосконалення. Її призначення — управління проектами з розробки програмного забезпечення. Може також використовуватися в роботі команд обслуговувань програмного забезпечення або як підхід управління розробкою і супроводом програм. Недоліки методу — присутність конкретних ролей та процесів (щоденні зустрічі, спринт-огляди), чітка визначеність спринтів у часі, неможливість додавати або скасовувати роботу під час спринту.

eXtreme Programming (XP) — цінності та методи цієї методології будуються навколо створення сприятливих умов для розвитку та формування звичок, які допомагають розробнику писати простий і легко змінюваний код. Іншими можливостями екстремального програмування є: парне програмування, здатність до обширної перевірки сирцевого коду, модульне тестування всього коду, уникання створення функціональності до того моменту необхідності, простота та ясність коду, можливість очікування на зміну вимог, досить часте спілкування з замовником і учасниками команд, — що розглядається як сильні сторони, так і сторони, що ускладнюють роботу з цією методологією.

Feature Driven Development (FDD) — всі ітераційні цикли орієнтовані на клієнта чи функцію програми, тому ця методологія є зручною для розробки тих програмних продуктів, умови реалізації яких дуже чітко визначені замовником і враховуються на початку розробки загальної моделі. Деталізація можлива лише на кожному інкрементному та ітераційному циклі. Негативним є момент, що кожен програміст володіє лише частиною коду, яку розробляє.

Method Dynamic Systems Development Method (DSDM) — ця методика вимагає створення прототипів системи під час розробки на ранніх етапах, дозволяючи майбутнім користувачам протестувати її і виявити всі недоліки. Один з ключових факторів успіху методу DSDM — залучення користувача. Метод DSDM наполягає на проведенні тестування на кожній ітерації, що може не всіх задовольняти.

Враховуючи вищевикладене, зазначимо, що серед agile-методів найбільшої популярності у ІТ-сфері набули ті методології, що дозволяють застосовувати їх для процесу ефективного і

продуктивного розроблення складних і надійних програмних систем, де водночас Agile означає ефективність і маневреність.

Необхідність вибору різних способів роботи полягають не у маніфесті, а у гнучкості розуму. Тому вибір методологій гнучкості проекту для шести осіб завжди відрізнятиметься від умов роботи іншого проекту. Перспективи подальших досліджень полягають у тому, що, приймаючи зміни програми, які вимагаються шляхом зміни зовнішніх обставин як невід'ємної частини програмування, програмісти повинні побудувати теорію того, як відповідні питання можуть підтримуватися виконанням програми. А щодо вибору методології, як зазначив Алістер Кокберн: «..методологія — це особиста річ... команді доведеться адаптуватися як до себе, так і себе до методології спільної роботи...» [5, 96].

Перспективні напрямки подальших досліджень

Питання вибору методології залишається до кінця не розв'язаним. Цікавим і перспективним є більш ґрунтовний аналіз кожної з методологій у зіставленні вибору щодо реальних проєктів, що визначають нові перспективи даного дослідження.

Список використаних джерел

1. 12 принципів, що базуються на маніфесті Agile. *Agilealliance*. URL: <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/> (дата звернення: 22.02.2021).
2. Бахам К., Гіршхайм Р. Питання, виклики та запропоноване теоретичне ядро гнучкої розробки програмного забезпечення. *Журнал інформаційних систем*. 07 березня 2021 року. DOI: <https://doi.org/10.1111/isj.12336>
3. Едісон Г., Ван С., Конбой К та ін. Порівняння методів широкомасштабної гнучкої розробки програмного забезпечення: систематичний огляд літератури *IEEE Транзакції з програмної інженерії*. 26 Березня 2021. doi: 10.1109/TSE.2021.3069039
4. Коен Г. Гнучкість у майстерності для менеджерів з продукту: посібник зі створення виграшних продуктів разом з гнучкими командами розробників. Купертіно, Каліфорнія: Super Star Press, 2010.
5. Кокберн А. Гнучка розробка програмного забезпечення. Січень 2000. 221с.
6. Курман М., Телль П., Хебіг Р. та ін. Що робить гнучку розробку програмного забезпечення гнучкою. *IEEE Операції з програмної інженерії*. 26 липня 2021. DOI: 10.1109/TSE.2021.3099532
7. Ледбрук Л. Управління проєктом водоспадної моделі: огляд. 13 липня, 2012. URL: <http://projectcommunityonline.com/waterfall-project-management-an-overview.html> (дата звернення: 22.02.2021).
8. Маніфест Agile. *Agilealliance*. URL: <https://www.agilealliance.org/agile101/the-agile-manifesto/> (дата звернення: 22.02.2021).
9. Повна С. В. Світовий досвід запровадження гнучких методологій управління конкурентоспроможністю підприємств та організацій. *Проблеми і перспективи економіки та управління*. № 1(21), 2020. URL: <http://ppeu.stu.cn.ua/article/view/211371> (дата звернення: 22.02.2021).
10. Рантюк І. І. Огляд гнучких методологій в управлінні ІТ проєктами. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2021/01/141.pdf> (дата звернення: 22.02.2021).
11. Стойка М., Мірча М., Гіліч-Міку Б. Розробка програмного забезпечення: гнучка проти традиційної. *Informatica Economică*. Т. 17 (4), 2013. С. 64-76. DOI:10.12948/issn14531305/17.4.2013.06
12. Торе Д. Імпровізація в малих програмних організаціях. *IEEE Програмне забезпечення*. 17 (5). 2000, С. 82-87. DOI: 10.1109/52.877872.

13. Фойтік Р. Екстремальне програмування у розробці специфічного програмного забезпечення. *Procedia Комп'ютерні науки*. Том 3, 2011. С. 1464-1468. DOI: <https://doi.org/10.1016/j.procs.2011.01.032>
14. Хода Р., Салле Н., Гранді Д. Становлення та еволюція гнучкої розробки програмного забезпечення. *IEEE Програмне забезпечення*. Липень, 2018. С. 1-7. DOI:10.1109/MS.2018.290111318
15. Цзююнь Чо. Проблеми та виклики розвитку програмного забезпечення Agile зі Scrum. *Питання інформаційних систем*. Вип. IX. № 2. 2008. С. 188-195. URL: https://acis.org/iis/2008/S2008_950.pdf (дата звернення: 22.02.2021).
16. Яржебович О., Вейхброт П. Якісне дослідження нефункціональних вимог до гнучкої розробки програмного забезпечення. *Доступ IEEE*. Том: 9. 08 березня 2021. С. 40458 – 40475. DOI: 10.1109/ACCESS.2021.3064424
17. Яшина К. В., Ялова К. М., Сугаль Є. О. Огляд гнучких методологій розробки програмного забезпечення. *Збірник наукових праць Дніпровського державного технічного університету (технічні науки)*. Том 1, № 30 (2017). С. 153-156. URL: <http://sj.dstu.dp.ua/article/view/145898> (дата звернення: 22.02.2021).