

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДОЛОГІЙ РОЗРОБКИ ІТ ПРОДУКТУ

¹Вінницький національний технічний університет;

Анотація

Розглянуто аналіз відомих методологій розробки ІТ продукту. Запропоновано алгоритм вибору методології.

Ключові слова: водоспадна; парадигма програмування; структурний аналіз; функціональна специфікація.

Abstract *The analysis of known methodologies of IT product development is considered. The algorithm selection methodology.*

Keywords: waterfall; programming paradigm; structural analysis; functional specification.

Вступ

Методологія розробки програмного забезпечення – сукупність методів, які використовуються на різних стадіях життєвого циклу розробки програмного забезпечення, що мають спільний філософський підхід та дозволяють забезпечити найкращу ефективність процесів розробки. Кожна методологія характеризується: філософським підходом або основними принципами. Ці принципи, від яких залежить ефективність всієї методології, зазвичай можна коротко сформулювати і легко пояснити узгодженою множиною моделей та методів, які реалізують визначену методологію. В більш вузькому значенні, у випадку, коли методологія застосовується на стадії програмування (конструювання), її зазвичай називають парадигмою програмування. Поширені методології програмування: водоспадна модель (waterfall), макетування (prototyping), ітеративна та інкрементна розробка (iterative and incremental development), (spiral development), спіральна модель (spiral model), швидка розробка програмного забезпечення (rapid application development), екстремальне програмування (extreme programming), різні види методологій гнучкої розробки (agile methodology) [1].

Можна простежити три шляхи виникнення методологій. По-перше, вони можуть бути результатом практичного досвіду. По-друге, методології можуть походити від однієї з чотирьох моделей алгоритму: абстрактна машина Тюрінга (імперативне програмування), рекурсивні функції Гільберта і Аккермана (структурне програмування), лямбда-числення Черча (функціональне програмування), нормальні алгоритми Маркова (логічне програмування). По-третє, методології можна пояснити через візуалізацію структури та моделей відповідно до мов моделювання та програмування. Складовими частинами можуть бути структура даних, структура управління і логіка. Кожна з моделей визначає частину методології. Вибір методології відповідає особливостям проекту, вимогам замовника, враховує досвід розробників та складність розробки та її запровадження.

Результати дослідження

Визначимо основні поняття, які необхідно врахувати для вибору методології розробки. Модель життєвого циклу – загальний опис того, як відбувається процес розробки. Методологія – більш деталізований набір правил, практик і принципів як спосіб реалізації тією чи іншою моделі. Існують різноманітні реалізації гнучких методологій. Наприклад, методологія Скрам реалізує ітеративну модель розробки.

Фреймворк процесів – загальна модель та програмна реалізація процесу розробки. Приклади: RUP, EssUp.

Каскадна модель (або модель водоспаду, Waterfall) – характеризується тим, що етапи розробки, такі як: аналіз, проектування, реалізація, тестування, йдуть один за одним. Дозволяє

швидко створювати систему, без додаткових витрат на організацію процесу розробки. Однак вона працює тільки тоді, коли вимоги стабільні і не змінюються в ході розробки.

V-Model - придумана в Німеччині і США модель, як спосіб поліпшити каскадну модель для застосування в державних проєктах. V-Model містить чітко фіксовані вимоги, вартість і час. Відмінність в тому, що етап аналізу та проєктування пов'язаний з етапом тестування. Наприклад, під час аналізу вимог одночасно вивчаються підходи до тестування, піна етапі проєктування архітектури системи розробляються високорівневі плани і сценарії тестування, в період проєктування компонентів системи вивчаються способи тестування компонентів та їх взаємодії, створюються сценарії тестування, утиліти, що допомагають в тестуванні, інструкції, скрипти тощо. Все це допомагає краще зрозуміти вимоги і спроєктувати систему. Однак тут, також як і в каскадній моделі, небажано, щоб вимоги змінювалися під час розробки.

Спіральна модель (Spiral) – орієнтована на проєкти, в яких є серйозні ризики. Розробка представляється у вигляді спіралі. Кожен виток спіралі – ітерація. Виток спіралі складається з чотирьох етапів: планування, аналіз ризиків, розробка, оцінювання замовником. В кінці кожної ітерації вирішується, чи варто продовжувати проєкт. Характерною рисою є те, що на етапі аналізу ризиків створюються прототипи, концепти, моделі які покликані дозволити ризик на ранній стадії. Чим далі рух по спіралі, тим більше розробки продукту і менше прототипів і концептів. Типове застосування такої моделі – дослідні проєкти. Модель передбачає багато витрат на кожному етапі і не виправдана в системах, де ризики незначні [2].

Ітеративна модель – орієнтована на проєкти, де вимоги можуть змінюватися по ходу розробки. Проєкт складається з ітерацій (від 1-2 до 6 тижнів). Кожна ітерація може включати в себе етап аналізу, проєктування, реалізації, тестування. Має більший обсяг витрат на організацію процесу, ніж каскадна модель, проте вартість виправлення помилки в залежності від тривалості проєкту не така висока. Різноманітні реалізації ітеративної моделі – Scrum, XP, використання Kanban – мають свої особливості.

Методологія Скрам (Scrum) одна з популярних реалізацій ітеративної моделі, в якій ітерація називається спринтом. Команда складається з 3 ролей: власник продукту (представник замовника), скрам-майстер (стежить за проходженням процесу), інші члени команди. Спринт починається з мітингу планування, коли команда відбирає і розподіляє завдання на ітерацію, формуючи беклог спринту. Спринт закінчується аналітичним оглядом спринту, де проводиться демонстрація продукту і мітингом ретроспективи спринту, на якому обговорюються поліпшення. Щодня проводяться 15-хвилинні скрам-зустрічі, на яких швидко аналізуються такі питання: який результат, що робиш, чи є проблеми та які вони. Можливі проблемні питання швидко вирішуються або/і для рішення проблеми призначається спеціальний час або вносяться зміни в план спринтів, загальний план розробки тощо.

Методологія екстремального програмування (XP) - складається з 12 практик: парне програмування, розробка через тестування, рефакторинг, формування архітектури, колективне володіння кодом, безперервна інтеграція, замовник в команді, періодичні релізи, гра в планування, 40-годинний робочий тиждень, стандарти кодування, метафора системи. Обов'язкове використання всіх 12 практик.

Методологія Канбан (Kanban) – використання конвеєру завдань. Має всього 3 правила: візуалізація процесу розробки за допомогою канбан-дошки, обмеження на кількість завдань на кожному етапі, постійне вимірювання продуктивності команди і поліпшення. Ця методологія часто використовується в інших як додатковий інструмент візуалізації.

Методологія RAD (Rapid Application Development) – орієнтована на швидку розробку програми, ітеративно, з максимально простою архітектурою, мінімальними витратами на процес, максимально використовуючи готові компоненти і потужні інструменти. Має обмеження на тривалість проєкту - 60-90 днів. Методологія передбачає використання готових компонентів.

Протистояння Agile і Waterfall не так теоретичне, скільки практичне. Вибір невідповідної методології, в кращому випадку істотно загальмує розвиток, проєкту в гіршому - відправить в список «ТОП-провалів року» [2]. На рис. 1. Представлено алгоритм вибору методології розробки.

Гнучка і каскадна модель розробки проєкту (Agile і Waterfall відповідно) – одна з найбільш популярних серед інших методологій управління. Agile не можна назвати просто моделлю – це загальний підхід до формування гнучкої методології – система ідей і принципів «гнучкого» управління проєктами, на основі яких розроблені популярні методи Scrum, Kanban та інші.

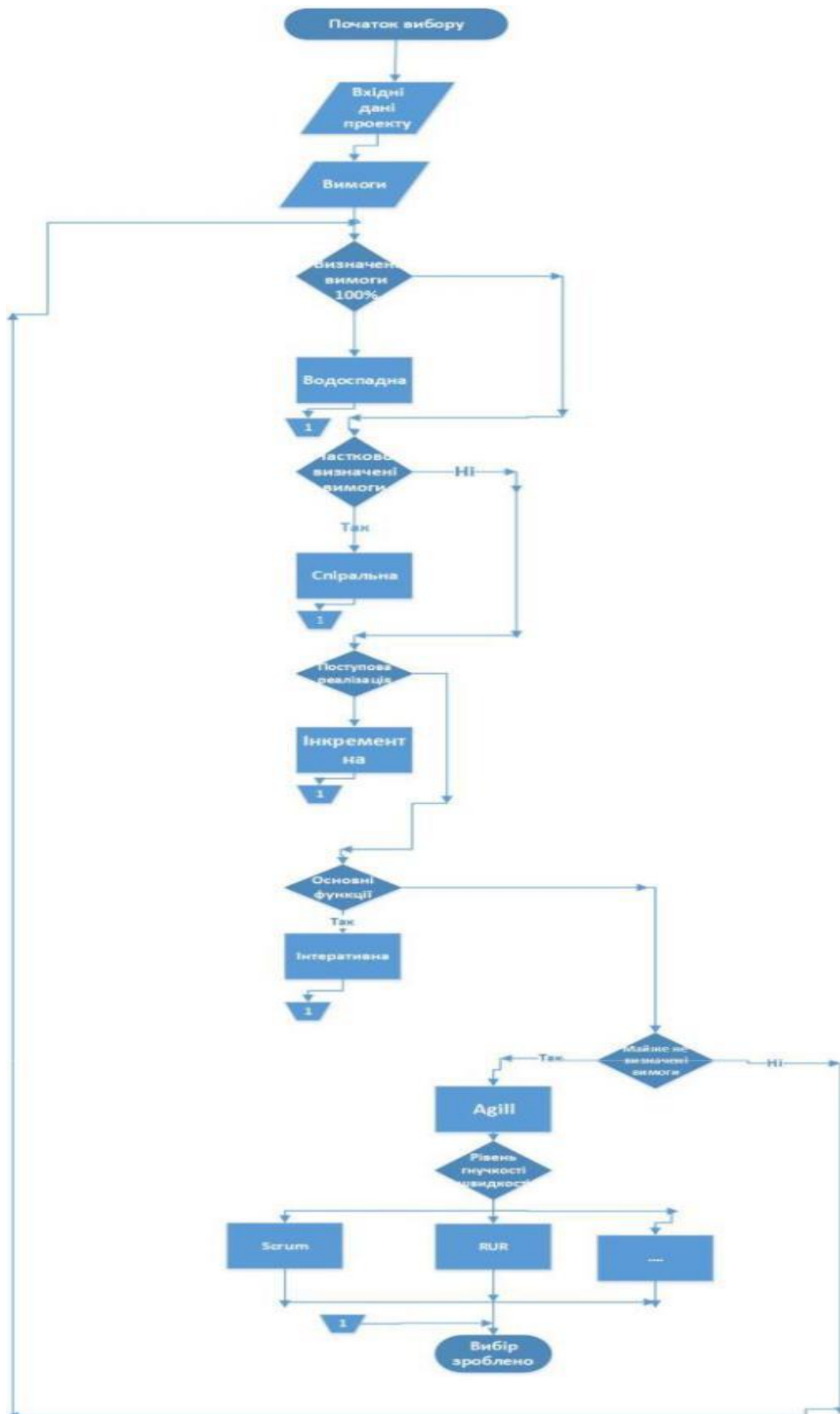


Рисунок 1 – Алгоритм вибору методологій

Ключовий принцип – розробка через короткі ітерації (цикли), в кінці кожного з яких замовник (користувач) отримує робочий код або продукт. Waterfall – методологія послідовного управління проектами, яка реалізує перехід з одного етапу на інший без пропусків і повернень на попередні стадії. Необхідно відмітити, що гнучка методологія передбачає використання часткової водоспадної моделі на кожному етапі розробки при умові визначених вимог[2].

Приймаючи рішення на користь того чи іншого методу, головним завданням є створення не тільки якісного продукту, але і можливість його ефективного практичного запровадження.

Гнучка модель вимагає високого професіоналізму від виконавця і ідеально підходить для стартапів, пілотних проектів та проектах, впровадження яких передбачає адаптивні процеси до різноманітних умов використання. Водоспадна модель найчастіше використовується у випадках чіткого визначення вимог як до розробки, так і до впровадження та висококваліфікованих користувачів. Деякі розробники вважають, що в межах одного проекту можна оптимально поєднати Agile і Waterfall для різних модулів системи.

Висновки

Вибір методики управління проектами є життєво важливим для реалізації успішного програмного продукту та його запровадження. Аналіз відомих підходів базується на критеріях вибору, що враховують особливості розробки від ідеї до впровадження, визначення вимог проекту, складності модулів, можливостей адаптації, потреб розвитку програми, рівня кваліфікації розробників, користувачів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. 6 most-common-software-development-methodologies. Режим доступу до ресурсу: <https://www.upwork.com/resources/most-common-software-development>
2. Agile или Waterfall-какой вариант соответствует вашему бизнесу? Режим доступу до ресурсу: <https://worksection.com/blog/waterfall-vs-agile.html>.
3. Сравнение методологий веб-разработки. Режим доступу до ресурсу: <https://vc.ru/flood/42084-agile-ili-waterfall-sravnienie-metodologiy-veb-razrabotki>.

Коваленко Олена Олексіївна – канд. техн. наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, Вінниця, e-mail: ok@vntu.edu.ua

Kovalenko Olena – Cand. Sc. (phd), Assistant Professor of Department Software, Vinnytsia National Technical University, Vinnytsia, e-mail: ok@vntu.edu.ua

Денисюк Алла Василівна – асистент кафедри програмного забезпечення, Вінницький національний технічний університет, Вінниця, e-mail: dealla@vntu.edu.ua

Denusjuk Alla – Assistant of Department Software, Vinnytsia National Technical University, Vinnytsia, e-mail: dealla@vntu.edu.ua

Бажан Вікторія Михайлівна – студентка групи 2ПІ-196, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, email: bazhan.viktoriya@icloud.com

Bazhan Victoria – Department Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email: bazhan.viktoriya@icloud.com.