

УДК 004.5 (045)

В. О. Яковенко, Ю. В. Ульяновська, Т. Ю. Яковенко, Т. А. Чупілко

АДАПТАЦІЯ ПРИНЦИПІВ AGILE МЕТОДОЛОГІЇ ДЛЯ УПРАВЛІННЯ ПРОЄКТОМ РОЗРОБКИ ПРОГРАМНОГО ЗАСТОНОКУ

Університет митної справи та фінансів, Дніпро

Анотація. В роботі описано використання деяких принципів Agile методології та SCRUM фреймворку для управління реальним проєктом, який має свою специфіку та особливості. Розглянуті процеси, що мають зв'язок з проєктним менеджментом: проєктування логіки та архітектури додатку, організація роботи з кодом та ресурсами додатку командою з розробки. Досліджуваний проєкт є навчальним 3D додатком в ігровій формі, що являє симулятор роботи митного поста. Даний проєкт побудований на базі Unreal Engine 4 та використовує Git як систему контролю версій для командної роботи. Визначено, що успішність застосування принципів гнучкої методології істотно залежить від коректності виконання початкового етапу (проєктування архітектури та логіки додатку) та адекватного до принципів методології розподілу задач на підзадачі. Розроблена архітектура повинна бути гнучкою. Вона має передбачати можливість змін або розширення функціональності розроблюваного програмного продукту без внесення змін до вже розробленої частини програмного продукту та без радикальних змін самої архітектури. Результати даного дослідження можуть бути використані для розробки методології проєктування архітектури та логіки програмного забезпечення, що розроблятиметься за використання Agile методології. Застосування гнучкої методології для управління проєктами розробки програмних продуктів порівняно з методологією водоспаду дозволяє скоротити час розробки до 10-12 разів та витрати на виправлення помилок (багів) до 20 разів.

Ключові слова: методологія Agile, управління проєктом, розробка додатку, адаптація SCRUM фреймворку.

Анотация. В работе описано использование некоторых принципов Agile методологии и SCRUM фреймворка для управления реальным проектом, который имеет свою специфику и особенности. Рассмотрены процессы, имеющие связь с проектным менеджментом: проектирование логики и архитектуры приложения, организация работы с кодом и ресурсами приложения командой по разработке. Исследуемый проект является учебным 3D приложением в игровой форме и представляет собой симулятор работы таможенного поста. Данный проект построен на базе Unreal Engine 4 и использует Git как систему контроля версий для командной работы. Agile методология имеет весомые преимущества в современных условиях постоянных и быстрых изменений технологий и потребностей пользователей относительно управления проектами разработки программных продуктов. Но ее применение требует адаптации к особенностям проекта, что определяет актуальность данного исследования. Определено, что успешность применения принципов гибкой методологии значимо зависит от корректности выполнения начального этапа (проектирование архитектуры и логики приложения) и адекватного принципам методологии распределения задач на подзадачи. Разработанная архитектура должна быть гибкой. Она должна предусматривать возможность изменения или расширения функциональности разрабатываемого программного продукта без внесения изменений в готовую часть программного продукта и без радикальных изменений самой архитектуры. Результаты данного исследования могут быть использованы для разработки методологии проектирования архитектуры и логики программного обеспечения, которое будет разрабатываться с использованием Agile методологии. Применение гибкой методологии для управления проектами разработки программных продуктов по сравнению с методологией водопада позволяет сократить время разработки до 10-12 раз и затраты на исправление ошибок (багов) до 20 раз.

Ключевые слова: методологии Agile, управление проектом, разработка приложения, адаптация SCRUM фреймворка.

Abstract. The article describes the use of some principles of Agile methodology and SCRUM framework for managing a real project, which has its own specifics and features. The processes related to project management are considered: designing the logic and architecture of the application, organizing work with the code and resources of the application by the development team. The investigated project is a 3D educational application in a game form and is a simulator of the work of a customs post. This project is built on the basis of Unreal Engine 4 and uses Git as a version control system for team work.

Agile methodology has significant advantages in today's conditions of constant and rapid changes in technology and user needs regarding project management of software development. But its application requires adaptation to the specifics of the project, which determines the relevance of this study.

It has been determined that the success of the application of the principles of the agile methodology significantly depends on the correctness of the initial stage (design of the architecture and application logic) and the methodology that is adequate to the principles of distributing tasks into subtasks. The designed architecture must be flexible. It should provide for the possibility of changing or expanding the functionality of the developed software product without making changes to the finished part of the software product and without radical changes in the architecture itself.

The results of this study can be used to develop a methodology for the design of architecture and software logic, which will be developed using the Agile methodology. The use of an agile methodology for managing software development projects in comparison with the waterfall methodology can reduce development time by up to 10-12 times and the cost of fixing errors (bugs) by up to 20 times.

Key words: agile methodology, project management, application development, adaptation of SCRUM framework.

DOI: <https://doi.org/10.31649/1999-9941-2021-52-3-44-52>.

Вступ

Стрімкий розвиток інформаційних технологій, масове впровадження автоматизацій процесів спричинили значне зростання попиту на розробку програмних продуктів. Конкурентна боротьба серед компаній вимагає чіткого дотримання термінів розробки. На противагу цьому, зростання складності процесів, що підлягають алгоритмізації, необхідність пошуку інноваційних шляхів вирішення поставлених задач, оптимізації розроблюваних програмних продуктів, тощо часто спричиняють затримки термінів виконання. Зазначене протиріччя між необхідністю дотримання дисципліни та створення комфортних умов для інновацій викликало появу нових методів управління проєктами з розробки програмного забезпечення, коли традиційні методи менеджменту стали неефективними.

Необхідність зміни методології була спричинена початком розробки інтернет-додатків. З'явилася необхідність виводити на ринок якнайшвидше веб-сайти, додатки та нові можливості. Швидко змінювалися інструменти та платформи розробки. Через ці зміни з'явилися спроби пошуку більш ефективної методології управління проектами розробки програмного забезпечення. Конкуренція на ринку викликала потребу враховувати відгуки користувачів, що було неможливо за використання старої методології.

Agile-розробка програмного забезпечення заохочує гнучке реагування на зміни та приділяє особливу увагу чотирьом основним цінностям: люди та командна взаємодія має перевагу над процесами та інструментами; працездатне програмне забезпечення має перевагу над вичерпною документацією; співпраця із замовником має перевагу перед обговоренням умов контракту; готовність до змін важливіша за дотримання плану [1].

Такий підхід до управління проектами з розробки програмного забезпечення дозволяє швидко реагувати на зміни та завершувати розробку у найкоротші терміни, що надзвичайно актуально в сучасних умовах постійних та швидких змін технологій та потреб користувачів. Проте застосування гнучкої методології до управління проектами розробки програмного забезпечення потребує її адаптації до особливостей проекту. Актуальним у зв'язку з цим є дослідження принципів адаптації нових методів до управління проектами з розробки програмного забезпечення з урахуванням їх особливостей.

Актуальність

Agile-методологія – це практика, що сприяє неперервній ітерації розробки та тестування протягом всього життєвого циклу проекту з розробки програмного забезпечення. Відповідно до цієї методології, розробка програмного продукту, тестування та доопрацювання виконуються одночасно на відміну від водоспадної методології, де ці етапи відбувалися послідовно.

У роботі [1] проаналізовані фактори, що є визначальними для успішності у сучасному світі інформаційних технологій. Було визначено, що успіх розроблюваного проекту, а, відповідно, і бізнесу, пов'язаного з ним, значною мірою залежить від способу управління проектом, ефективної організації робочих процесів, використання сучасних інструментів розробки. На думку авторів роботи [1], одним з визначальних факторів є методологія управління проектом. Аналогічну думку висловлюють автор роботи [2], де зазначено, що саме вірно обрана методологія управління проектом дозволяє успішно реалізувати проект, уникаючи негативного впливу інших факторів. У роботі [3] висловлена думка, що вірно обрана методологія управління процесом реалізації проекту дозволяє з максимальною ефективністю організувати роботу команди. Завдяки цьому можна врахувати всі особливості проекту як з метою уникнення негативних наслідків, так і з метою отримання всіх можливих переваг. Але залишилося невирішеним питання, чим саме слід керуватися при виборі методології проекту.

В роботі [4] зазначено, що всі проекти різні, кожен має свої особливості і тому не існує єдиної ідеальної системи управління проектами, що підходить для кожного з видів проектів. В роботі [5] проблема вибору методології проаналізована з позиції кадрів. Висловлено думку, що не існує системи управління, яка б підходила кожному керівнику і була зручна для всіх членів команди.

За час існування проектного управління було створено чимало ефективних підходів, методів і стандартів, які можна взяти на озброєння. Існують різні моделі та методології для організації життєвого циклу розробки продукту (Software Development Life Circle – SDLC), управління проектом та процесами, які в ньому проходять. Відповідно до даних роботи [6], найбільш поширеною практикою є застосування принципів Agile методології, її фреймворків, таких як SCRUM або Kanban. Інформація наведена у щорічному звіті «State of Agile» підтверджує факт росту використання Agile методології у різних регіонах світу, різними типами організацій, різними видами індустрій [7]. Автори роботи [8] зазначають, що темпи розповсюдження Agile методології пояснюються тим, що вона може бути легко адаптована до проектів різних галузей. Крім того, Agile методологія дозволяє покращити цілий ряд характеристик: зменшити ризики; ефективніше використовувати бюджет; отримувати результати за короткі строки; мати постійну можливість вносити зміни та оновлювати вимоги; отримувати більш якісні і протестовані продукти; неперервно покращувати робочі процеси; адаптуватися до змін, мати прозорий робочий процес; організовано зберігати і використовувати ресурси проекту; налаштувати ефективну командну роботу та комунікацію.

Часто проекти можуть мати свою унікальну специфіку, тому застосування Agile фреймворків у цьому вигляді може бути неефективним або неможливим для них. Agile фреймворки пропонують лише рекомендовану схему побудови управління проектом і організації робочих процесів.

В той же час, однією з ключових властивостей Agile методології є адаптивність. Отримати ефективну схему управління проектом і організації робочих процесів, можна взяти за основу один з Agile фреймворків з його принципами та здійснити ряд змін та покращень з урахуванням специфіки проекту. Таким чином, результатом стане гнучка схема управління, що враховує нюанси предметної області проекту, яка дозволить ефективно керувати процесами, отримувати максимальні результати, користуватись перевагами, які приносить Agile методологія.

Agile методологія може застосовуватись для організації проєктів у різних індустріях. Так, принципи Agile методології можуть бути використані для менеджменту науковими проєктами. У праці [6] описуються умови за яких SCRUM фреймворк може бути успішно реалізований для управління проєктами даного напрямку. Результати даного дослідження ґрунтуються на даних, отриманих під час опитувань та інтерв'ю науковців, які спробували використовувати Scrum для організації своїх робочих процесів. Також Agile методологію використовують в галузі з розробки ігрових додатків, беручи до уваги схожі та відмінні характеристики, порівнюючи розробку ігрових додатків та програмного забезпечення [8]. Під час розробки ігрового додатку з використанням Agile як методології розробки важливими є планування ітерацій (sprints planning), відгуки та зворотній зв'язок (feedbacks). Відповідно до праці [9], підвищення якості кінцевого продукту може бути досягнуте за рахунок правильно організованих процесів з використанням Agile методології та SCRUM фреймворку.

Таким чином, актуальною задачею є дослідження особливостей адаптації принципів Agile методології для управління проєктами.

Мета

Метою роботи є дослідження особливостей адаптації принципів Agile методології для управління проєктами. Це дасть можливість розробити методологію адаптації принципів Agile до особливостей проєктів.

Задачі

1. Дослідити досвід використання деяких принципів Agile методології та SCRUM фреймворку для управління реальним проєктом створення додатку.
2. Проаналізувати особливості організації процесів, що тісно пов'язані з проєктним менеджментом за використання принципів Agile методології/

Розв'язання задач

Використання принципів Agile методології та SCRUM фреймворку для управління проєктом створення додатку

Досліджуваний проєкт є навчальним 3D додатком в ігровій формі, що являє симулятор роботи митного поста. Користувач зможе поринути в процеси, які відбуваються під час митного оформлення та контролю в ігровій формі та з якісною графікою (рис 1). Корисним даний симулятор буде як для студентів, так і для працівників митної служби. Використовуючи даний ігровий симулятор вони зможуть отримати досвід і розуміння роботи митного поста.



Рисунок 1 – Демонстрація ігрового додатку. Митний ангар в якому проводиться митний огляд транспортних засобів

Розглянемо управління проєктом і організацію робочих процесів. За основу було взято SCRUM фреймворк, але в спрощеній формі відповідно до потреб і специфіки проєкту. Структура SCRUM фреймворку наголошує, що для успішного використання фреймворку повинні бути наявні певні ролі, артефакти, події [11, 12]. Команда з розробки (Development Team) складалась з 6 учасників і була крос-функціональною, як і вимагає SCRUM. Під крос-функціональністю мається на увазі спроможність членів команди виконувати різні види роботи: створення дизайну, участь у плануванні, виконання задач з розробки та тестування, тощо. Ролі власника продукту (Product Owner) та SCRUM майстра (Scrum Master) виконували відповідні учасники команди з розробки.

Під час робочого процесу проводились такі події та заходи, як: Спринт (Sprint) або ітерація (Iteration), уточнення вимог (Refinement), щоденна SCRUM зустріч (Daily Scrum Meeting) – проводилась факультативно і зазвичай в онлайн режимі, тому що роботою займалась розподілена за різними локаціями команда, SCRUM огляд (SCRUM Review), ретроспектива (Retrospective). Важливим артефактом, який не був описаний вище, є SCRUM дошка (SCRUM Dashboard). Ключові властивості, які має SCRUM Dashboard, є такими: зберігає і відображає «Product Backlog»; зберігає і відображає «Sprint Backlogs»; відображає виконавця, статус кожної з задач; доступна і прозора репрезентація робочих процесів для власника продукту, замовників, SCRUM майстра і команди з розробки.

Розглянемо організацію робочого процесу та схему роботи «SCRUM Dashboard» в описуваному проєкті. Схема, зображена на рис. 2, відображає типи задач (Work items), які розміщуються на дошці, зв'язки між ними, а також додаткові статуси, які можуть мати задачі. В проєкті були використані такі типи задач: ідея (Idea), історія (Story), задача (Task), Баг (Bug).

Додаткові статуси задач: критична задача (Critical), перешкода (Blocker).

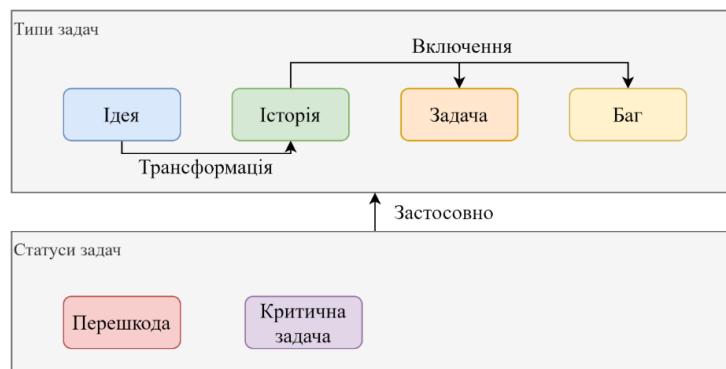


Рисунок 2 – Типи задач на SCRUM дошці, зв'язки і додаткові статуси

Схема на рис. 3 описує елементи, які включаються до різних типів задач. До задач можуть бути прикріплені додаткові файли, необхідні для виконання задач. Також можуть вказуватися елементи системи контролю версій Git. «Story» включає в себе лист критеріїв (Acceptance Criteria), за виконання і дотримання яких вона може вважатися завершеною. «Tasks», «Bugs» включають перелік під-задач, які є пунктами, що визначають у зручному форматі кроки, котрі потрібно пройти для того, щоб завершити задачу.



Рисунок 3 – Елементи, які включають у себе різні типи задач (Work Items)

На схемі, зображеній на рис. 4, відображено принцип роботи «SCRUM Dashboard» та організації робочого процесу.

Нові «Ideas», «Stories» потрапляють у колонку «Backlog». Під час планування визначаються «Stories», які взяті у наступний «Sprint». «Stories» декомпонуються на «Tasks», уточнюються, закріплюються за виконавцем і потрапляють до колонки «Current Sprint». Коли виконавець бере задачу у роботу, вона переміщується до колонки «In-Action». Коли задача виконана, вона переходить до колонки «Review». Після того, як лідер (Tech Lead) команди перевірить виконану роботу, задача потрапляє в ко-

лонку «Done» – у випадку, якщо не знайдено помилок або дефектів. Якщо знайдено помилки або дефекти, задача переміщується до колонки «Test or Fix». Після роботи над усуненням помилок відбувається повторний огляд лідером команди. Більш серйозна за масштабом помилка або помилка, виявлена в будь-якій частині функціоналу поза фазою «Review», розміщується як «Bug» відповідної «Story». У кінці кожної ітерації після «Sprint Review», що включає демонстрацію, виконані протягом ітерації задачі переміщуються і зберігаються у окремому листі на дошці. Ці листи можуть бути приховані і відображені знову за потреби.

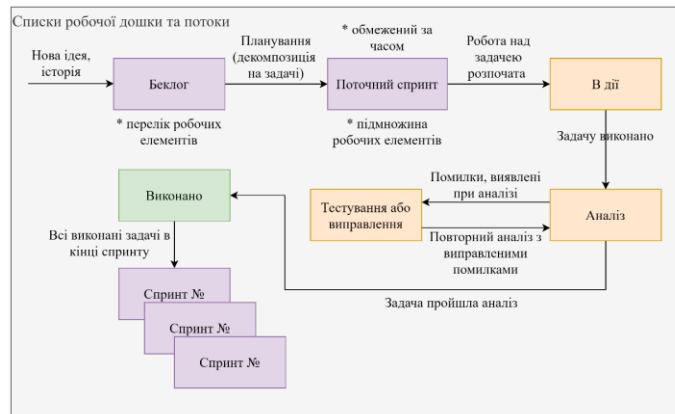


Рисунок 4 – Діаграма робочого процесу і принципу роботи SCRUM дошки (Workflow diagram)

В якості інструменту для організації робочого процесу відповідно до Agile SCRUM фреймворку використовувався сервіс «Trello». Зазначений сервіс має зручну дошку, дозволяє визначати вимоги до задач, визначати перелік можливих статусів задач, змінювати їх статус, прикріплювати результат виконання задачі, залишати коментарі для виконавців щодо виявлених помилок, тощо.

Правильна організація зберігання коду і ресурсів додатку, а також робота з ними командою з розробки є важливим аспектом робочого процесу і забезпеченням прозорого відображення результатів розробки. Використання системи контролю версій тісно пов'язано з описаними Agile принципами. В даному проекті в якості системи контролю версій використовувався Git та сервіс GitHub для зберігання репозиторію ігрового додатку. Дані інструменти є безкоштовними, доступними, добре документованими, допомагають організувати командну роботу над кодом та ресурсами додатку, а також можуть бути інтегровані з Unreal Engine та Trello.

Ключовими елементами Git репозиторію є гілка (branch) та коміт (commit). Гілки створюються в ієрархічному порядку одна від одної і зберігають історію змін у вигляді комітів. Коміт являє собою зафіксовані зміни у коді. При створенні гілки, нова гілка включає повну історію змін гілки, з якої вона була створена. Код з однієї гілки може бути влитий до іншої на певному етапі розробки. Тобто технічна реалізація кожної задачі ведеться на незалежній гілці, одним з учасників команди з розробки. Коли технічна реалізація завершена, протестована і оглянута, гілка вливається до батьківської гілки.

Командна робота з використанням Git потребує встановлення єдиної схеми і правил користування системою контролю версій. Тому перед початком застосування Git було сформовано дві схеми, які описують структуру і організацію репозиторію. Схема GitFlow, зображена на рис. 5, відображає організацію гілок репозиторію, на яких ведеться розробка: master, dev, release, feature.

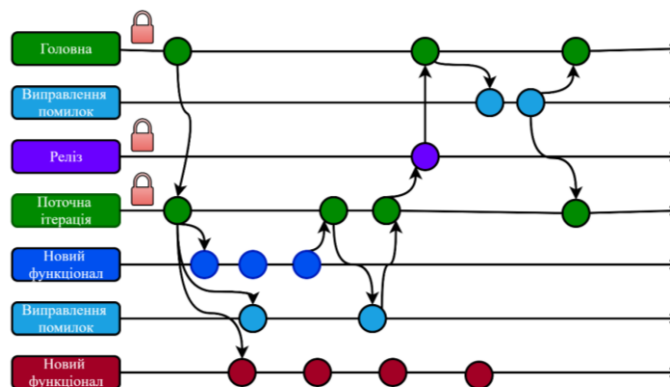


Рисунок 5 – Схема GitFlow

На рис. 6 відображено, за якою структурою організовано зберігання гілок репозиторію, а також правила іменування гілок. Наприклад, feature-гілки зберігаються у виділеній директорії, а до їх назви включається індекс «Story» або «Task» з Scrum-дошки і безпосередньо коротка назва, яка описує розроблений функціонал на даній гілці (наприклад features/s01-t1-game-score-implementation). В свою чергу формування назви «release» гілки подібно до прикладу формування feature-гілки, але передбачає також включення індексу релізу (наприклад releases/1.0.0-two-game-modes).

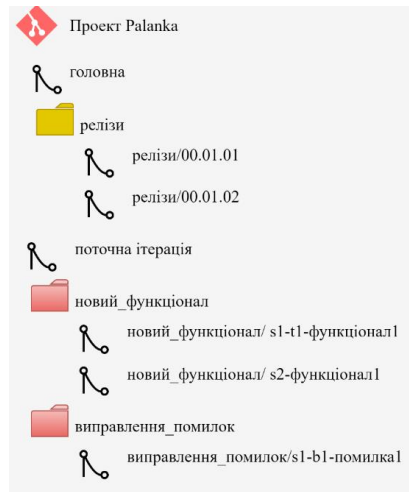


Рисунок 6 – Схема GitFlow

Також слід підкреслити переваги інтеграції GitHub і Trello. До карток задач в Trello можуть бути включені посилання на гілки, коміти, а також посилання на події вливання однієї гілки до іншої (pull requests). Досягається описана інтеграція шляхом встановлення розширення для Trello. За рахунок розширень функціональність і можливості використання даного інструменту значно зростають. Інтеграція системи контролю версій і Scrum дошки створюють єдину екосистему, коли ключові артефакти розробки зберігаються в одному місці, завжди зафіксовані і доступні. Це, безперечно, є важливим для ефективної командної роботи і управління робочими процесами, адже і учасники команди з розробки, і замовники отримують чітке розуміння статусу виконання робіт і прогресу розробки продукту.

Результати аналізу особливостей організації процесів за використання принципів Agile методології

Розробка будь-якого програмного продукту починається з проектування архітектури та логіки додатку. Використання принципів Agile методології означає, що процес розробки є гнучким та адаптивним. Вимоги до розроблюваного додатку можуть змінюватись в процесі роботи, архітектура може бути переглянута, розширена або оновлена. Перший крок проектування – збирання вимог від замовників ігрового додатку і формування документу, який включає ці вимоги. Другий крок – аналіз даних вимог і створення діаграм, які описують логіку розроблюваного ігрового додатку. Третій крок передбачає виділення задач для команди з розробки на основі результатів попередніх двох кроків. Четвертим кроком є впровадження принципів Agile методології, а саме SCRUM фреймворку, для організації процесу розробки. Описані кроки відображені на рис. 7.

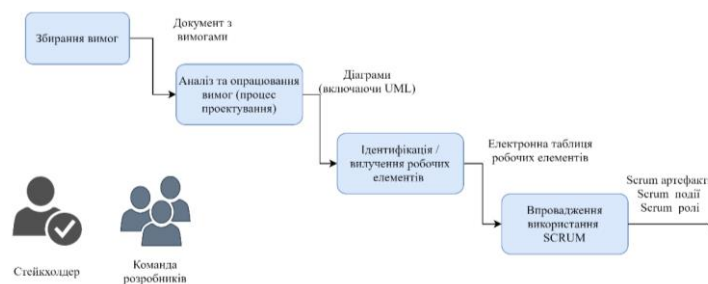


Рис. 7. Збір вимог, процес проектування і введення SCRUM фреймворку

Розглянемо UML діаграми створені в процесі проектування. На рис. 8 відображена діаграма варіантів використання розроблюваного додатку. Вона відображає цільових користувачів і варіанти викорис-

тання ігрового додатку. До цільових користувачів додатку відносяться студенти, які навчаються митній справі та співробітники митниці, які не працюють безпосередньо на митних постах і пунктах пропуску.

Заплановані варіанти використання додатку: ознайомлення зі змістом митної декларації, її реквізитами, значеннями, які можуть в них вноситись; ознайомлення з інфраструктурою митного посту, шляхом відтворення митного посту за допомогою засобів 3D моделювання; ознайомлення з режимами митного огляду транспортних засобів та їх етапами; демонстрація використання системи «Rapiscan» для перевірки вантажу транспортних засобів [10]; оцінка знань користувачів додатку після ознайомлення з матеріалом.

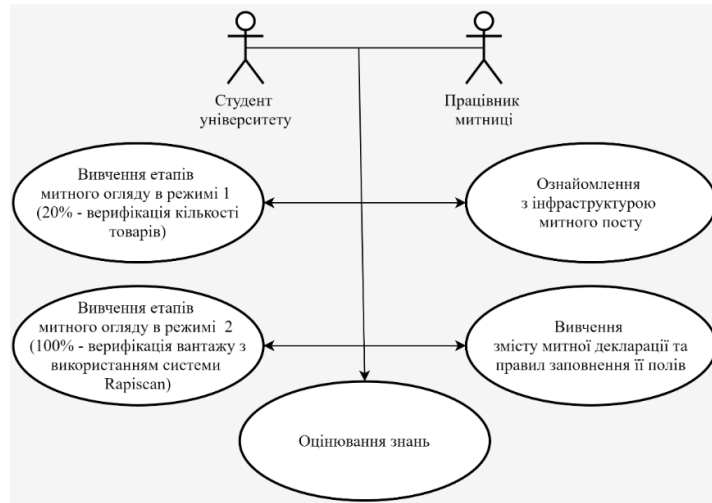


Рисунок 8 – UML діаграма варіантів використання

UML діаграми, представлені на рис. 9 описують логіку ігрових режимів, що розроблювались. На цих діаграмах (рис. 9) представлені кроки двох видів митного огляду транспортних засобів.

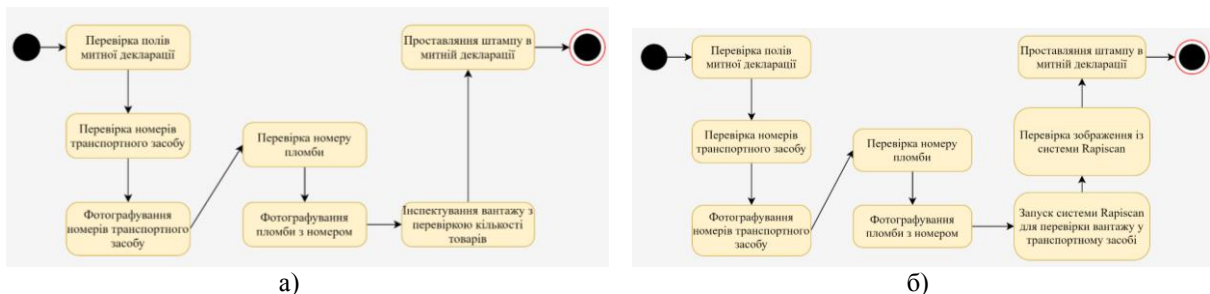


Рисунок 9 – Діаграма станів для ігрового режиму: а) №1 (безпосередня перевірка вантажу транспортного засобу за кількістю товарів), б) №2 (перевірка вантажу транспортного засобу з використанням системи «Rapiscan»)

Кожен з режимів включає наступні кроки: перевірка заповнення митної декларації; перевірка номеру транспортного засобу; фотофіксація номеру транспортного засобу; перевірка номеру пломби на транспортному засобі; фотофіксація номеру пломби на транспортному засобі; внесення штаму до митної декларації.

Ігровий режим, відображений на рис. 4, включає перевірку вантажу транспортного засобу за номерами товарів. Ігровий режим, відображений на рис. 5, включає додатково перевірку вантажу транспортного засобу з використанням системи «Rapiscan» [10].

Висновки

1. Організація робочих процесів відповідно до принципів Agile методології дозволяє досягти прозорості щодо статусу виконання задач, термінів виконання задач, організувати ефективну комунікацію і командну роботу, ефективну систему зберігання коду, ресурсів та артефактів проекту. Застосування гнучкої методології дозволяє забезпечити реалізацію запланованого функціоналу додатку у короткі строки та отримувати результат кожної ітерації, покращувати робочі процеси у наступних ітераціях.

2. Успішність застосування принципів гнучкої методології істотно залежить від коректності виконання початкового етапу (проектування архітектури та логіки додатку). Архітектура та логіка додатку

повинні передбачати можливість доповнення функціональності та внесення змін, які дозволять підвищити конкурентоспроможність готового продукту та задовольнити замовника. Такий підхід гарантує дотримання одного з основних принципів гнучкої методології: готовність до змін важливіша за дотримання початкового плану.

Список літератури

- [1] Н. Рябоконт, Б. Рябоконт, А. Рябоконт, «Впровадження методології Agile: ціннісно орієнтований підхід», *Збірник наукових праць Черкаського державного технологічного університету*. Серія: Економічні науки, 49, 34-42. 2018. doi.org/10.24025/2306-4420.0.49.2018.136152.
- [2] І. О. Пилипенко, «Підвищення ефективності реалізації проектів в умовах високого рівня невизначеності на прикладі проекту будівництва житлового будинку», *Управління розвитком складних систем*, 14, 72-75. 2013. doi: 10.32347/2412-9933.2013.14.
- [3] R. Hoda, J. Noble, S. Marshall, «Self-organizing roles on agile software development teams», *IEEE Transactions on Software Engineering*, 39(3), 422-444. 2013.
- [4] E. S. Hidalgo, «Management of a multidisciplinary research project: A case study on adopting Agile methods», *Journal of Research Practice*, 14 (1). 2018.
- [5] J. Galvis-Ardila, L. V. Anduquia, H. M. Diez-Silva, «Adopting Communications Management Practices in Project Management: A Preliminary Study in Bogotá, Colombia. In N. Moreno-Monsalve», *Handbook of Research on Project Management Strategies and Tools for Organizational Success*, 339-352. 2020.
- [6] E. S. Hidalgo, «Adapting the scrum framework for agile project management in science: case study of a distributed research initiative», *Heliyon*, 5 (3), 14-47. 2019.
- [7] The 15th State of Agile Report is copyrighted by Digital.ai, 2021. [Online]. Available: <https://stateofagile.com/#ufh-i-661275008-15th-state-of-agile-report/7027494>.
- [8] R. Kortmann, C. Hartevel, «Agile game development: lessons learned from software engineering. Learn to game, game to learn». *National University of Singapore, Singapore, 40th ISAGA Conference*, 29/06/09, 1-11. [Online]. Available: https://web.northeastern.edu/casperhartevel/pubs/Agile_ISAGA2009.pdf
- [9] Jing-Wei Liu, Chia-Yu Ho, Jamie Y.T. Chang & Jacob Chia-An Tsai, «The role of Sprint planning and feedback in game development projects: Implications for game quality», *Journal of Systems and Software*, 2019, 154, 79-91.
- [10] Rapiscan Systems. Cargo & vehicle inspection. [Online]. Available: <https://www.rapiscansystems.com/en/products/category/cargo-and-vehicle-inspection>
- [11] Engstrom, H., Marklund, B. B., Backlund, P. & Toftedahl, M., *Game development from a software and creative product perspective a quantitative literature review approach*. Entertainment Computing, 2018, 10-22.
- [12] Schwaber, K., Sutherland, J. (2020). *The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game* [Online]. Available: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>.
- [13] Ojiako, G. U., Hamdan, A., & Bashir, H., «Modeling and analyzing interrelationships among project success factors and criteria». *Proceedings of the 9th International Conference on Industrial Engineering and Operations Management*. Bangkok, Thailand, 2019, March 5-7, 910-917. <http://www.ieomsociety.org/ieom2019/papers/257.pdf>.

Стаття надійшла: 14.11.2021

References

- [1] N. Ryabokon, B. Ryabokon, A. Ryabokon, «Agile methodology: value-oriented approach», *Proceedings of Scientific Works of Cherkasy State Technological University*, Series: Economic Sciences. 49, 34-42. 2018. <https://doi.org/10.24025/2306-4420.0.49.2018.136152>.
- [2] I. O. Pylypenko, «Improving the efficiency of project implementation in conditions of a high level of uncertainty on the example of a residential construction project», *Management of complex systems development*, 14, 72-75. 2013. doi: 10.32347/2412-9933.2013.14.
- [3] R. Hoda, J. Noble, S. Marshall, «Self-organizing roles on agile software development teams», *IEEE Transactions on Software Engineering*, 39(3), 422-444. 2013.
- [4] E. S. Hidalgo, «Management of a multidisciplinary research project: A case study on adopting Agile methods», *Journal of Research Practice*, 14 (1). 2018.
- [5] J. Galvis-Ardila, L. V. Anduquia, H. M. Diez-Silva, «Adopting Communications Management Practices in Project Management: A Preliminary Study in Bogotá, Colombia. In N. Moreno-Monsalve», *Handbook of Research on Project Management Strategies and Tools for Organizational Success*, 339-35. 2020.
- [6] E. S. Hidalgo, «Adapting the scrum framework for agile project management in science: case study of

- a distributed research initiative», *Heliyon*, 5 (3), 14-47. 2019.
- [7] The 15th State of Agile Report is copyrighted by Digital.ai, 2021. [Online]. Available: <https://stateofagile.com/#ufh-i-661275008-15th-state-of-agile-report/7027494>.
- [8] R. Kortmann, C. Harteveld, «Agile game development: lessons learned from software engineering. Learn to game, game to learn». *National University of Singapore, Singapore, 40th ISAGA Conference*, 29/06/09, 1-11. [Online]. Available: https://web.northeastern.edu/casperhartevelde/pubs/Agile_ISAGA2009.pdf
- [9] Jing-Wei Liu, Chia-Yu Ho, Jamie Y.T. Chang & Jacob Chia-An Tsai, «The role of Sprint planning and feedback in game development projects: Implications for game quality», *Journal of Systems and Software*, 2019, 154, 79-91.
- [10] Rapiscan Systems. Cargo & vehicle inspection. [Online]. Available: <https://www.rapiscansystems.com/en/products/category/cargo-and-vehicle-inspection>
- [11] Engstrom, H., Marklund, B. B., Backlund, P. & Toftedahl, M., *Game development from a software and creative product perspective a quantitative literature review approach*. Entertainment Computing, 2018, 10-22.
- [12] Schwaber, K., Sutherland, J. (2020). *The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game* [Online]. Available: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>.
- [13] Ojiako, G. U., Hamdan, A., & Bashir, H., «Modeling and analyzing interrelationships among project success factors and criteria». *Proceedings of the 9th International Conference on Industrial Engineering and Operations Management*. Bangkok, Thailand, 2019, March 5-7, 910-917. <http://www.ieomsociety.org/ieom2019/papers/257.pdf>.

Відомості про авторів

Яковенко Вадим Олександрович – доктор технічних наук, доцент, професор кафедри.

Ульяновська Юлія Вікторівна – кандидат технічних наук, доцент, завідувач кафедри.

Яковенко Тетяна Юрївна – кандидат економічних наук, доцент, доцент кафедри.

Чупілко Тетяна Анатоліївна – кандидат технічних наук, доцент, доцент кафедри.

В. А. Яковенко, Ю. В. Ульяновська, Т. Ю. Яковенко, Т. А. Чупілко

АДАПТАЦІЯ ПРИНЦИПОВ AGILE МЕТОДОЛОГІЇ ДЛЯ УПРАВЛЕННЯ ПРОЕКТОМ РАЗРАБОТКИ ПРОГРАМНОГО ПРИЛОЖЕННЯ

Університет таможенного дела и финансов, Днепр

V. Yakovenko, Yu. Ulianova, T. Yakovenko, T. Chupilko

ADAPTING THE PRINCIPLES OF AGILE METHODOLOGY TO MANAGE AN APPLICATION DEVELOPMENT PROGRAM PROJECT

University of Customs and Finance, Dnipro