



УДК 004.413

Ю. С. Кордунова

Львівський державний університет безпеки життєдіяльності, м. Львів, Україна

АНАЛІЗ ТА РОЗРОБЛЕННЯ КОНЦЕПТУАЛЬНОЇ МОДЕЛІ УПРАВЛІННЯ ЖИТТЄВИМ ЦИКЛОМ СПЕЦІАЛІЗОВАНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ БЕЗПЕКО-ОРІЄНТОВАНОГО СПРЯМУВАННЯ

Наведено детальний огляд розроблення спеціалізованого програмного забезпечення безпеко-орієнтованого спрямування. Виявлено невідповідність відомих сьогодні методів розроблення до умов, в яких відбувається створення інноваційних програмних систем безпекового спрямування. Саме тому мета роботи – дослідження наявних моделей управління життєвим циклом спеціалізованого програмного забезпечення та розроблення концептуальної моделі процесу управління життєвим циклом програмних систем безпеко-орієнтованого спрямування, адаптованої під специфіку роботи Державної служби України із надзвичайних ситуацій та корелює із принципами гнучкої методології.

Досліджено відомі на сьогодні гнучкі методи та підходи до розроблення критично важливих сервісів, зокрема у військовій, залізничній, аерокосмічній, медичній та інших сферах діяльності. Обґрунтовано, як гнучкість може допомогти в інноваціях спеціалізованого безпеко-орієнтованого програмного забезпечення, об'єднано основні концепції методів гнучкої методології управління життєвим циклом програмних систем, зважаючи на специфіку розроблення для служби порятунку; розширено емпіричні дані про можливість та переваги використання гнучких методів у безпековій галузі. Удосконалено гнучкий метод Scrum, зокрема запропоновано використовувати математичний апарат теорії графів (мережеве планування) для автоматизації етапу планування та визначення критично важливих функцій для розроблення мінімально життєздатного продукту безпеко-орієнтованої системи.

Розроблено концептуальну модель процесу управління життєвим циклом безпеко-орієнтованих сервісів, яка ґрунтується на гнучкому підході до розроблення програмного забезпечення, що своєю чергою дасть змогу автоматизувати роботу проектних команд та удосконалити розроблення спеціалізованого програмного забезпечення. Згодом на її основі запропоновано розробити інформаційну систему підтримки прийняття рішень щодо управління життєвим циклом розроблення програмних систем безпеко-орієнтованого спрямування, актуальну для Державної служби України з надзвичайних ситуацій.

Ключові слова: agile методологія; Scrum метод; теорія графів; життєвий цикл програмного забезпечення.

Вступ / Introduction

У сучасному цифровому світі розроблення програмного забезпечення займає центральне місце практично у кожній сфері діяльності. Не є винятком і рятувальна сфера. Особливо актуальне в наш час розроблення безпеко-орієнтованих сервісів для Державної служби України із надзвичайних ситуацій (ДСНС України), які спрямовані на забезпечення безпеки та допомогу рятувальникам під час виконання їхніх службових обов'язків. Ці сервіси мають велике значення для інформування та запобігання надзвичайним ситуаціям, ефективності та швидкості реагування на небезпеку, збільшення шансів на врятування життя людей та мінімізацію можливих збитків.

Розроблення безпеко-орієнтованих програмних систем потребує специфічних підходів та методів, які притаманні критично важливим сервісам. Саме тому постає потреба у розробленні нових та удосконаленні вже відомих підходів до управління життєвим циклом програмного забезпечення, враховуючи специфіку роботи служби порятунку.

На перший погляд, ідеальною методологією розроблення безпеко-орієнтованих сервісів є каскадна модель, оскільки саме її прийнято використовувати для державних проєктів, де необхідні чітке документування, розрахунок бюджету та аналіз всіх можливих процесів та ризиків. Для того, щоб застосувати каскадну модель, команда розробників повинна: працювати зі стабільними вимогами; мати глибокі знання про продукт, що розробляється; чітко дотримуватись планів та термінів [2]. Цей підхід добре працюватиме для вже наявних інформаційних систем, проте дуже важко буде накласти ці принципи на умови, в яких розробляється новий, інноваційний програмний продукт. Насправді інновації щодо безпеко-орієнтованих систем є дуже складним, оскільки нові технології та завдання можуть спричинити невизначені ризики, впливати на надійність та якість продукту, який розробляють. Використання гнучких методологій у процесі розроблення забезпечить мінімізацію ризиків за рахунок ітераційного випуску продукту, безперервної комунікації розробників та замовника, а також постійного поліпшення та можливості внесення змін під час розроблення системи [2].

Об'єкт дослідження – процеси управління життєвим циклом програмного забезпечення.

Предмет дослідження – методи та моделі управління життєвим циклом спеціалізованого програмного забезпечення безпеко-орієнтованого спрямування в динамічному оточенні.

Мета роботи – дослідження наявних моделей управління життєвим циклом спеціалізованого програмного забезпечення та розроблення концептуальної моделі процесу управління життєвим циклом програмних систем безпеко-орієнтованого спрямування, яка адаптована під специфіку роботи ДСНС України та корелює із принципами гнучкої методології.

Для досягнення зазначеної мети поставлено такі основні завдання дослідження:

- проаналізувати відомі сьогодні методи управління життєвим циклом спеціалізованого програмного забезпечення (критично важливих сервісів);
- обґрунтувати, як гнучкість може допомогти у інноваціях спеціалізованого безпеко-орієнтованого програмного забезпечення, щоб підвищити ефективність та надійність таких програмних продуктів;
- об'єднати основні концепції методів гнучкої методології управління життєвим циклом програмного забезпечення, зважаючи на специфіку розроблення такого ПЗ для служби по-рятунку;
- розробити концептуальну модель процесу управління життєвим циклом безпеко-орієнтованих сервісів, яка ґрунтується на гнучкому підході до розроблення програмного забезпечення.

Матеріали та методи дослідження. Для досягнення поставленої мети у роботі використано метод теоретичного аналізу відомих підходів до розроблення критично важливих сервісів, а для створення концептуальної моделі управління життєвим циклом безпеко-орієнтованих систем – теорію графів (мережевий метод планування).

Аналіз останніх досліджень та публікацій. Питання вибору методології управління життєвим циклом спеціалізованих програмних систем розглянуто у працях багатьох вчених. У деяких роботах навіть зроблено висновки про несумісність використання гнучких методологій та розроблення спеціалізованого програмного забезпечення [1], [3]. Але новіші результати поставили під сумнів ці висновки, визначивши основні переваги, що виникають під час впровадження гнучких методів у розроблення критично важливих систем, а саме: документація, оскільки вона не є необхідною для гнучкого розроблення програмного забезпечення; змінні вимоги, оскільки традиційні методи цьому не сприяють [4]; життєвий цикл розроблення програмного продукту, оскільки ці проекти не розробляються ні ітераційно, ні поступово (проблема із плануванням випуску програмного продукту) [5]; тестування, яке, у традиційній методології, відбувається лише на завершальних стадіях розроблення продукту [6].

Сьогодні все частіше успішно використовують гнучкі методології управління життєвим циклом програм-

них систем у військовій [7], [8], [9], [13], залізничній [10], [24], аерокосмічній [11], [12], медичній [6], [22] та інших сферах діяльності. Зокрема, у роботі [13] здійснено дослідження на основі розроблення системи командування та управління для Генерального штабу армії Італії із використанням гнучких методологій управління. Після тринадцяти п'ятиденних спринтів розробники змогли надати готовий продукт, який відповідав усім функціональним та нормативним вимогам армії. Попри витрати, спрямовані на навчання та пристосування до нового методу розроблення програмного забезпечення, витрати на саму розробку стали нижчими, ніж раніше із використанням традиційних методів управління життєвим циклом програмних систем. У роботі [12] викладено результати реорганізації команд розробників у Scrum-команди в Agile Transformation Бразильського аеронавтичного обчислювального центру, зокрема був побудований симулятор польоту та розроблена система, яка контролює політ іноземних літаків у повітряному просторі Бразилії. Автори у роботі [6] описали традиційний підхід та виклики, з якими стикаються команди розробників під час створення програмного забезпечення для медичних установ, а також успішно впроваджені гнучкі практики до розроблення у цій галузі. А у джерелі [14] наведено рекомендації стосовно відповідності міжнародним стандартам та документам регуляторних органів, із використанням гнучких практик agile у розробленні програмного забезпечення для медичних установ.

Автори у роботі [15] запропонували метод, оснований на розширенні підходу DevOps до безпечних систем. Цей автоматизований підхід легко зіставити із відомим методом гнучкої методології Scrum [16], [19]. У роботі [10] автори також розширили цей метод, який отримав назву Scrum for Safety (S4S). Він спрямований на допомогу науково-дослідницьким групам у розробленні безпечних рішень для залізничної галузі. У статті [18] висвітлено застосування гнучкого підходу до забезпечення безпеки критично важливого вбудованого програмного забезпечення для космічного корабля NASA Orion. Зокрема, вказано виклики, пов'язані із забезпеченням критичних функцій програмної системи та необхідність використання адаптивного підходу до забезпечення якості та валідації цих функцій у контексті еволюції методів розробки.

Автори у роботі [20] запропонували новий метод, що підтримує гнучку методологію розроблення програмного забезпечення, ґрунтується на концепції MVC і складається із трьох основних частин: модель, представлення та контролер. Ідея полягає у тому, що замовник самостійно проєктує користувацький інтерфейс, що зменшує ризик відхилення або обмеженого використання програмного забезпечення і сприяє простішому та безпомилковому розробленню програмного забезпечення.

З огляду на результати аналізу, існує потребу в створенні нового гнучкого методу розроблення спеціалізованого програмного забезпечення, котрий буде адаптований під специфіку роботи Державної служби України із надзвичайних ситуацій та корелюватиме із принципами гнучкої методології управління життєвим циклом програмного забезпечення.

Результати дослідження та їх обговорення / Research results and their discussion

Відомо, що кожна ітерація (спринт) процесу розроблення програмного забезпечення за найпопулярнішою сьогодні гнучкою моделлю Scrum складається із таких етапів: створення беклогу продукту, початок спринту та визначення *mvр*, створення беклогу спринту, пріоритизація користувацьких історій (завдань із беклогу) та їх оцінювання, подрібнення користувацьких історій та їх оцінювання, розроблення продукту, демо та ретроспектива [16]. Проте означені процеси адаптовані та апробовані під використання у традиційних (класичних) проектних командах, що працюють за гнучкими моделями розроблення програмного забезпечення. Це команди розробників, що налічують п'ять – дев'ять осіб із розподілом обов'язків за ролями (позиціями), які більшість часу впродовж спринту зосереджуються на вирішенні поставлених завдань. Динамічність такого процесу розроблення полягає лише у можливій зміні обсягу або змісту робіт. Чисельність команди та час на виконання фіксовані. Проте змінність оточення процесів розроблення програмного забезпечення, що реалізується за Agile-методологією, може полягати не лише в можливій зміні визначеного обсягу подій, а й у динаміці визначеного часового ресурсу.

Наприклад, як під час воєнного стану, запровадженого на території України, так і у період функціонування оперативних служб у повсякденному режимі, окремі ІТ-підрозділи Державної служби України із надзвичайних ситуацій займаються розробленням програмного забезпечення, вбудованих систем та інших безпеко-орієнтованих сервісів (далі – БОС) [5], орієнтованих на підвищення ефективності функціонування оперативно-рятувальних служб.

Членами команд розроблення таких БОС в оперативних формуваннях є кадрові працівники відповідних служб, які в міру своєї службової підготовки повинні поєднувати діяльність, пов'язану із розробленням означених сервісів, із іншими різновидами оперативної та/або службової діяльності. Тому, зважаючи на специфіку роботи й особливих учасників проектних команд із розроблення безпеко-орієнтованих сервісів, динаміка проектного середовища набуває дещо іншого значення.

Динамічність тепер зосереджується не лише на обсягу робіт, а й часу їх реалізації. На перший погляд, можна розробити БОС у таких умовах за каскадною моделлю. Проте це припущення хибне, адже більшість БОС не мають чіткого терміну необхідних робіт, а функціонал зазнає динамічних змін, поправок у ході розроблення.

За означених умов розроблення безпеко-орієнтованих сервісів проектними командами оперативних формувань відбуватиметься відповідно до моделі, наведеної на рис. 1.

Як бачимо, динамічні процеси з регламентованим переліком робіт, а також відсутність контролю (обмеження) часу на їх виконання у цій моделі можуть істотно ускладнювати своєчасне виконання роботи. Саме тому виникає необхідність у побудові нового методу для розроблення БОС, що відповідатиме парадигмі гнучкого управління процесом розроблення програмного забезпечення в оперативних формуваннях.

Для вирішення цього питання прийнято застосовувати математичний апарат теорії графів та адаптувати процеси мережевого планування, оптимізувавши розрахунок його основних часових параметрів під динамічні умови проекту. Цей підхід дасть змогу у реальному часі здійснювати перепланування завдань зі спринту, враховуючи критично важливі завдання для створення мінімально життєздатного продукту (*mvр*) програмної системи.

У цьому випадку додатковим завданням для команди розробників буде після пріоритизації та оцінювання користувацьких історій у беклозі спринту визначити попередні користувацькі історії чи функції, які потрібно виконати, оскільки деякі завдання можна реалізувати лише після завершення попередніх. У такому випадку розроблення безпеко-орієнтованих сервісів набуде вигляду, як на рис. 2.

Ураховуючи висвітлену вище концептуальну модель, розроблення БОС починається із формування беклогу продукту. У такому разі беклог містить всі завдання (користувацькі історії) та функції, які необхідно виконати для розроблення усієї програмної системи. Після формування беклогу весь процес розроблення, як і в звичній Scrum команді, розділяється на ітерації (спринти), кожна з яких повинна завершуватись випуском готового функціоналу (мінімально життєздатного продукту, *mvр*) та ретроспективою.

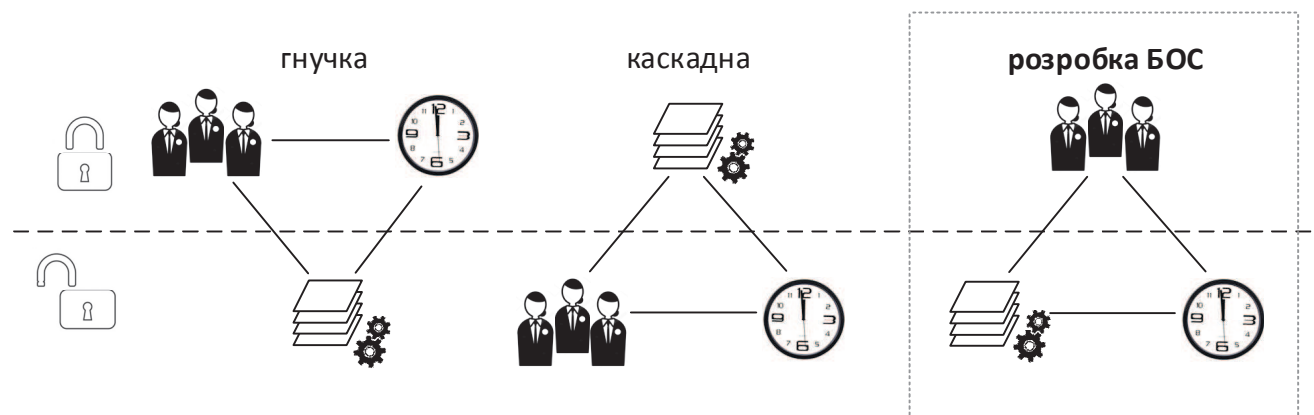


Рис. 1. Порівняння гнучкої, каскадної моделей розроблення програмного забезпечення з фактичною моделлю реалізації БОС[5] / Flexible and cascading software development models comparison with the actual SOS implementation model[5]

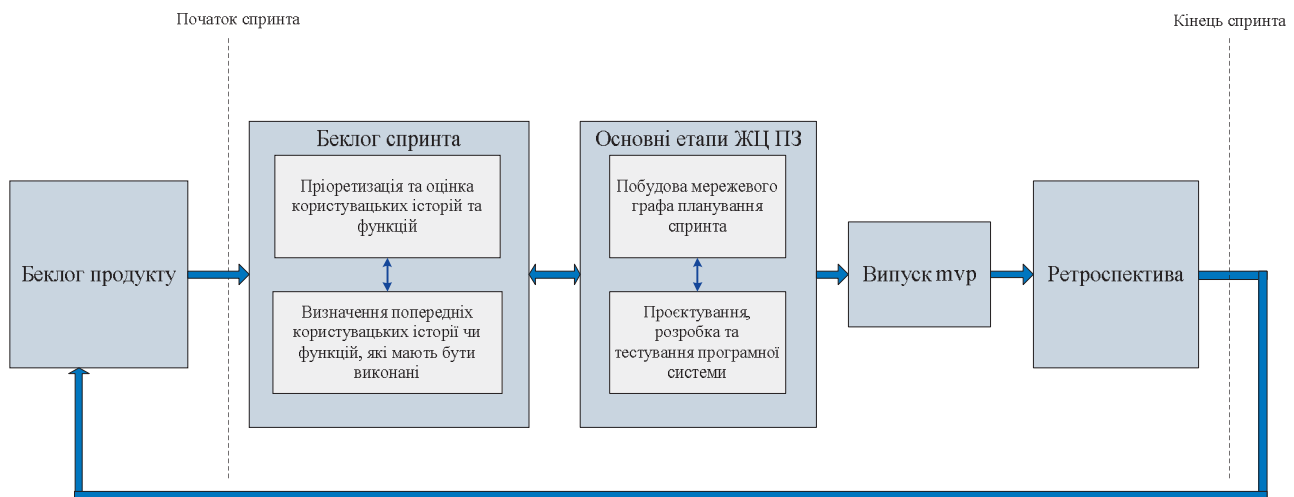


Рис. 2. Розроблена концептуальна модель процесу управління життєвим циклом БОС / The developed conceptual model of the SOS life cycle management

Зважаючи на специфіку розроблення безпеко-орієнтованих сервісів, беклог спринту формується так:

1. Пріоритезуються та оцінюються користувацькі історії та функції, які необхідно виконати за спринт.
2. Визначаються попередні користувацькі історії чи функції, які потрібно реалізувати, оскільки є такі завдання, які можна виконувати лише після завершення попередніх.

Між цими пунктами існує двосторонній зв'язок, оскільки у разі зміни пріоритету користувацької історії може змінюватись і попередня користувацька історія.

Після формування беклогу спринту відбувається розроблення самої системи. Для автоматизації етапу планування рекомендовано побудувати мережевий граф планування спринту та обчислити його основні показники (критичний шлях, ранні та пізні терміни виконання подій, резерви на виконання завдань). Такий підхід надасть проєктній команді можливість стежити в режимі реального часу за розробленням системи та, за потреби, змінювати хід виконання завдань. Мережеве планування дасть змогу ефективно розподіляти час на розроблення та автоматизувати процес визначення критично важливих функцій для випуску mvp.

Після планування відбуваються звичні етапи життєвого циклу програмного забезпечення (далі – ЖЦ ПЗ), такі як: проектування, розроблення, тестування програмної системи. На концептуальній моделі між плануванням та наступними етапами ЖЦ ПЗ зображено двосторонній зв'язок, оскільки залежно від процесу розроблення, складності завдань, досвіду проєктної команди та інших чинників тривалість розроблення може змінюватись, а отже, виникати потреба у переплануванні мережевого графа. Двосторонній зв'язок існує також і між формуванням беклогу спринту та основними етапами ЖЦ ПЗ, оскільки, залежно від визначеного під час мережевого планування критичного шляху, завдання та користувацькі історії беклогу спринта можуть змінюватись. І навпаки.

Завершується спринт випуском готового продукту, яким вже можуть скористатись у службовій діяльності

працівники ДСНС, та ретроспективою, на якій обговорюють, що можна поліпшити, та вносять корективи в наступну ітерацію.

Такий підхід до розроблення програмного забезпечення дасть змогу в режимі реального часу здійснювати перепланування процесу розроблення із мінімальними людськими та часовими ресурсами. Як зазначено вище (рис. 1), розроблення БОС характеризується суцільною невизначеністю в обсязі завдань та часових ресурсах. У зв'язку зі службовою діяльністю та специфічними умовами, в яких працюють співробітники ДСНС, завдання у беклозі можуть змінюватись у довольний час, відсутність досвіду та неправильна оцінка завдань можуть вплинути на дату випуску готового функціоналу, а це вплине на якість та швидкість роботи працівників служби порятунку. Розроблення безпеко-орієнтованих сервісів на основі цієї концептуальної моделі дасть змогу мінімізувати ресурси на етапі планування (без втрат для показника якості), змінювати беклог продукту у динамічних умовах (притаманних ДСНС) та найголовніше – уможливити оперативне перепланування та швидкий випуск готового функціоналу.

Обговорення результатів дослідження. Сьогодні є чимало гнучких підходів до розроблення програмного забезпечення, котрі розширюють роботу методу Scrum, зокрема це R-Scrum [26], Safe-Scrum [27], LeSS [28] та інші. Проте автори деяких наукових досліджень пропонують також нові інноваційні підходи до гнучкого розроблення спеціалізованого програмного забезпечення. Наприклад, у роботі [10] запропоновано ввести додаткові ролі верифікаторів, валідаторів та оцінювачів у команду розробників, щоб мати можливість перевіряти технічну якість програмного забезпечення та його адекватність початковій проблемі. Проте забезпечити таку кількість ролей та ефективне розроблення спеціалізованого програмного забезпечення в оперативних службах ДСНС неможливо. У дослідженні [27] автори наводять результат теоретичної роботи, в якій Scrum увідповіднено до різних стандартів. З огляду на труднощі незалежного тестування і управління докумен-

тацією, автори пропонують розроблення, кероване тестами, і наявність спеціальної команди для документації програмного забезпечення, що може бути не дуже ефективним і застосовним для критично важливих систем.

Проте, з огляду на отримані результати, відомі гнучкі методи розроблення програмних систем не корелюють із умовами, в яких відбувається розроблення безпеко-орієнтованих сервісів.

Розроблена концептуальна модель надасть змогу підвищити ефективність розроблення та планування окремих етапів управління життєвим циклом програмного забезпечення та коригувати їх у режимі реального часу. Надалі на її основі плануємо розробити інформаційну систему підтримки прийняття рішень щодо управління життєвим циклом розроблення програмних систем безпеко-орієнтованого спрямування, актуальну для Державної служби України з надзвичайних ситуацій.

Отже, за результатами виконаної роботи можна сформулювати наукову новизну та практичну значущість результатів дослідження.

Наукова новизна отриманих результатів дослідження – у результаті аналітичних досліджень наявних механізмів реалізації гнучкої методології управління IT проєктами розроблено концептуальну модель управління процесом розроблення спеціалізованих програмних систем, яка уможливує мінімізацію ресурсів IT проєктів у динамічному оточенні на етапі їх планування.

Практична значущість результатів дослідження – розроблена концептуальна модель ЖЦ ПЗ є підґрунтям для побудови інформаційної системи підтримки прийняття рішень, щодо управління життєвим циклом розроблення програмних систем безпеко-орієнтованого спрямування, що дасть змогу автоматизувати роботу проєктних команд та удосконалити процес розроблення спеціалізованого програмного забезпечення безпекового спрямування.

Висновок / Conclusions

У динамічних умовах вибір методології розроблення спеціалізованого програмного забезпечення є надзвичайно важливим. Як показав досвід таких розробок, відомі підходи та методи управління життєвим циклом таких програмних систем не корелюють з умовами, в яких здійснюється розроблення безпеко-орієнтованих сервісів, де, окрім змінних вимог, принципово важливий час виконання. Відповідно за результатами дослідження наявних методів управління життєвим циклом критично важливих сервісів у науковій праці отримано висвітлені нижче результати.

Обґрунтовано, як гнучкість може допомогти щодо інновацій спеціалізованого безпеко-орієнтованого програмного забезпечення, систематизовано основні концепції методів гнучкої методології управління життєвим циклом програмних систем, зважаючи на специфіку розроблення для служби порятунку; розширено наявні емпіричні дані про можливість та переваги використання гнучких методів у безпековій галузі (зокрема розроблено безпеко-орієнтовану систему на замовлення Державної служби України з надзвичайних ситуацій роботи “Розробка програмної системи кон-

сультаційної допомоги населенню при загрозі або виникненні надзвичайних ситуацій з інтегрованою функцією оповіщення на базі мобільних платформ”, а також систему для дослідження параметрів оцінки протипожежного стану об’єктів “Комп’ютерна аналітична система” [29]).

Розроблено концептуальну модель процесу управління життєвим циклом спеціалізованого програмного забезпечення (безпеко-орієнтованих сервісів), котра адаптована під специфіку роботи Державної служби України із надзвичайних ситуацій та корелює із принципами гнучкої методології управління життєвим циклом програмного забезпечення.

Надалі на основі одержаних результатів запропоновано розробити інформаційну систему підтримки прийняття рішень, щодо управління життєвим циклом розроблення програмних систем безпеко-орієнтованого спрямування, що, своєю чергою, дасть змогу автоматизувати роботу проєктних команд та удосконалити процес розроблення спеціалізованого програмного забезпечення безпекового спрямування.

References

- [1] Cawley, O., Wang, X., & Richardson, I. (2010). Lean/agile software development methodologies in regulated environments-state of the art. *Lean Enterprise Software and Systems. LESS 2010. Lecture Notes in Business Information Processing*, 65. https://doi.org/10.1007/978-3-642-16416-3_4
- [2] 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 (in Ukrainian). <https://doi.org/10.32347/2412-9933.2021.47.71-77>
- [3] Hajou, A., Batenburg, R., & Jansen, S. (2014). How the pharmaceutical industry and agile software development methods conflict: A systematic literature review. 14th International Conference on Computational Science and Its Applications. *IEEE*, 40–48. <https://doi.org/10.1109/ICCSA.2014.19>
- [4] Notander, J. P., Runeson, P., & Höst, M. (2013). A model-based framework for flexible safety-critical software development: a design study. *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 1137–1144. <https://doi.org/10.1145/2480362.2480575>
- [5] Kordunova, Y., Prydatko, O., Smotr, O. & Golovaty, R. (2023). Expert Decision Support System Modeling in Lifecycle Management of Specialized Software. *Lecture Notes in Data Engineering, Computational Intelligence, and Decision Making. ISDMCI 2022. Lecture Notes on Data Engineering and Communications Technologies*, 149. https://doi.org/10.1007/978-3-031-16203-9_22
- [6] McCaffery, F., Trektete, K. & Ozcan-Top, O. (2016). Agile – Is it Suitable for Medical Device Software Development? *Software Process Improvement and Capability Determination*, 609. https://doi.org/10.1007/978-3-319-38980-6_30
- [7] Messina, A., Fiore, F., Ruggiero, M., Ciancarini, P., & Russo, D. (2016). A new agile paradigm for mission-critical software development. *CrossTalk*, 29, 25-30.
- [8] Benedicenti, L., Cotugno, F., Ciancarini, P., Messina, A., Pedrycz, W., Sillitti, A., & Succi, G. (2016). Applying scrum to the army: a case study. 38th International Conference on Software Engineering Companion. *IEEE*, 725–727. <https://doi.org/10.1145/2889160.2892652>

- [9] Cotugno, F. R. & Messina, A. (2014). Adapting SCRUM to the Italian Army: Methods and (Open) Tools. Open Source Software: Mobile Open Source Technologies. OSS 2014. IFIP *Advances in Information and Communication Technology*, 427. https://doi.org/10.1007/978-3-642-55128-4_7
- [10] Barbareschi, M., Barone, S., Carbone, R. et al. (2022). Scrum for safety: an agile methodology for safety-critical software systems. *Software Qual. J.*, 30, 1067–1088. <https://doi.org/10.1007/s11219-022-09593-2>
- [11] Smith, J., Bradbury, J., Hayes, W., & Deadrick, W. (2019). Agile approach to assuring the safety-critical embedded software for NASA's orion spacecraft. 2019 IEEE Aerospace Conference, 1–10. <https://doi.org/10.1109/AERO.2019.8742095>
- [12] de Sá, F.R., Vieira, R. G. & da Cunha, A. M. (2019). Lessons Learned from the Agile Transformation of an Aeronautics Computing Center. Agile Methods. WBMA 2019. *Communications in Computer and Information Science*, 1106. https://doi.org/10.1007/978-3-030-36701-5_7
- [13] Benedicenti L., Messina A. & Sillitti A., (2017). iAgile: Mission Critical Military Software Development, 2017 International Conference on High Performance Computing & Simulation (HPCS), 545–552. <https://doi.org/10.1109/HPCS.2017.87>
- [14] ANSI Webstore. (2012). Guidance on the use of AGILE practices in the development of medical device software. URL: <https://webstore.ansi.org/standards/aami/aamitir452012r2018>
- [15] Casola, V., De Benedictis, A., Rak, M., & Villano, U. (2020). A novel security-by-design methodology: Modeling and assessing security by slas with a quantitative approach. *Journal of Systems and Software*, 163. <https://doi.org/10.1016/j.jss.2020.110537>
- [16] Schwaber, K., & Sutherland, J. (2020). The scrum guide. URL: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
- [17] de Sá, F. R. (2023). Scrum in Strongly Hierarchical Organizations: A Literature Review. Agile Methods. WBMA 2021. *Communications in Computer and Information Science*, 1642. https://doi.org/10.1007/978-3-031-25648-6_6
- [18] Smith J., Bradbury J., Hayes W. & Deadrick W. (2019). Agile Approach to Assuring the Safety-Critical Embedded Software for NASA's Orion Spacecraft. 2019 IEEE Aerospace Conference, *Big Sky*, 1–10. <https://doi.org/10.1109/AERO.2019.8742095>
- [19] Carbone, R., Barone, S., Barbareschi, M., & Casola, V. (2021). Scrum for Safety: Agile Development in Safety-Critical Software Systems. *Quality of Information and Communications Technology. Communications in Computer and Information Science*, 1439. https://doi.org/10.1007/978-3-030-85347-1_10
- [20] Avasthi A. & Mishra G. (2018). A New Framework for the Agile Software Development Method. Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), 436–438. <https://doi.org/10.1109/ICECA.2018.8474737>
- [21] Wang H. & Ma Z. (2023). Application and Improvement of Agile Development in Intelligent Health Hut Software Project. 7th International Conference on Management Engineering, Software Engineering and Service Sciences (ICMSS), 14–18. <https://doi.org/10.1109/ICMSS56787.2023.10118316>
- [22] Paez N., Fontdevila D. & Oliveros A. (2020). On the Influence of Agile in the Usage of Software Development Practices. IEEE Congreso Bional de Argentina (ARGENCON), Resistencia, Argentina, 1–7. <https://doi.org/10.1109/ARGENCON49523.2020.9505407>
- [23] Carbone, R., Barone, S., Barbareschi, M. & Casola, V. (2021). Scrum for Safety: Agile Development in Safety-Critical Software Systems. *Quality of Information and Communications Technology. QUATIC 2021. Communications in Computer and Information Science*, 1439. https://doi.org/10.1007/978-3-030-85347-1_10
- [24] Kordunova, Y., Feltynowski, M., Prydatko, O., & Smotr, O. (2023). Mathematical modeling of specialized safety-oriented software systems development process. *Bulletin of Lviv State University of Life Safety*, 27, 23–31. <https://doi.org/https://doi.org/10.32447/20784643.27.2023.03>
- [25] Steghöfer, J. P., Knauss, E., Horkoff, J. & Wohrab, R. (2019). Challenges of Scaled Agile for Safety-Critical Systems. In: Franch, X., Männistö, T., Martínez-Fernández, S. (eds) *Product-Focused Software Process Improvement. PROFES 2019. Lecture Notes in Computer Science*, 11915. Springer, Cham. https://doi.org/10.1007/978-3-030-35333-9_26
- [26] Anand, R. Vijay & Dinakaran M. (2018). Improved scrum method through staging priority and cyclomatic complexity to enhance software process and quality. *International Journal of Internet Technology and Secured Transactions*, 8 (2), 150–166. <https://doi.org/10.1504/IJTST.2018.093342>
- [27] Hanssen, G., Stålhane, T., & Myklebust, T. (2018). *SafeScrum – Agile Development of Safety-Critical Software*. New York: Springer. <https://doi.org/10.1007/978-3-319-99334-8>
- [28] Edison, H., Wang, X. & Conboy, K. (2022) Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review. *Transactions on Software Engineering*, 48(8), 2709–2731. <https://doi.org/10.1109/TSE.2021.3069039>
- [29] Chekryhin, O., & Prydatko, O. (2023) Study of the parameters for assessing the fire condition of objects and the construction of a computer analytical system “Computer analytical system” on their basis. (Report on research and development work No. 0122U200729). Lviv State University of Life Safety.

Yu. S. Kordunova

Lviv State University of Life Safety, Lviv, Ukraine

ANALYSIS AND DEVELOPMENT OF A CONCEPTUAL MODEL FOR LIFECYCLE MANAGEMENT OF SPECIALIZED SAFETY-ORIENTED SOFTWARE

A comprehensive examination of the process of developing specialized safety-oriented software is conducted. The incongruity of the presently known development methodologies with the circumstances in which innovative safety-oriented software systems are created is unveiled. For this reason, the purpose of this article is to investigate the existing methods and formulate novel approaches to managing the life cycle of specialized software (safety-oriented services), which are relevant to the specific requirements of the State Emergency Service of Ukraine and align with the principles of a flexible software life cycle management methodology.

A thorough exploration is conducted on the current agile methods and approaches employed in the development of critical services, specifically in sectors such as military, railway, aerospace, medical, and other domains of operation. The aim is to establish the merits of flexibility in fostering innovation within specialized safety-oriented software. The

fundamental concepts of agile methods for software system life cycle management are synergistically integrated, taking into consideration the unique requirements of rescue service development. Furthermore, the existing empirical data on the feasibility and advantages of implementing agile methods in the security industry are expanded upon. Especially, the agile Scrum method is enhanced through the proposal of employing the mathematical framework of graph theory (network planning method) to automate the planning stage and identify critical functions, including the critical path, the early execution time of events, the late execution times of events, and reserved time for tasks. This method aids in developing a minimum viable product for a safety-oriented system.

A conceptual model of the life cycle management process for safety-oriented services has been meticulously developed. This model is predicated on a flexible approach to software development, which aims to automate the tasks performed by project teams and enhance the overall efficacy of specialized software development. On its basis, it is proposed to develop a decision support system for managing the life cycle of safety-oriented software systems development, which is relevant for the State Emergency Service of Ukraine.

Keywords: agile methodology; Scrum method; graph theory; software life cycle.

Інформація про авторів:

Кордунова Юлія Сергіївна, ад'юнкт. Email: kordunovayulia@gmail.com; <https://orcid.org/0000-0003-0151-8285>

Цитування за ДСТУ: Кордунова Ю. С. Аналіз та розроблення концептуальної моделі управління життєвим циклом спеціалізованого програмного забезпечення безпеко-орієнтованого спрямування. *Український журнал інформаційних технологій*. 2023. Т. 5, № 2. С. 72–78.

Citation APA: Kordunova Yu. S. (2023). Analysis and development of a conceptual model for lifecycle management of specialized safety-oriented software. *Ukrainian Journal of Information Technology*, 5(2), 72–78. <https://doi.org/10.23939/ujit2023.02.072>