УДК 005.8:004.4 DOI: https://doi.org/10.30837/IYF.IIS.2024.275

ДОСЛІДЖЕННЯ МОЖЛИВОСТІ ФОРМУВАННЯ ПІДМНОЖИНИ УНІВЕРСАЛЬНИХ МОДЕЛЕЙ ТА МЕТОДІВ ТЕСТУВАННЯ ІТ-ПРОДУКТІВ

Таранченко С. І.
Науковий керівник – проф. Євланов М. В.
Харківський національний університет радіоелектроніки, каф. ІУС
м. Харків, Україна
e-mail: sofiia.taranchenko@nure.ua

This research addresses the challenge of increasing complexity in IT projects by exploring the adaptation of universal testing methods amidst diverse management methodologies. Through examining five key testing approaches—ISTQB, TMap, Agile Testing, DevOps Testing, and Risk-Based Testing—it highlights the necessity for a flexible, efficient testing framework that doesn't sacrifice software quality. The analysis seeks to identify a unified testing strategy by finding common phases across these methodologies, aiming to enhance testing practices and contribute to the development of a standardized testing framework suitable for various project environments.

Зростаюча складність ІТ-проєктів та спеціалізація методологій управління як проєкту в цілому, так і процесу тестування, породжує необхідність в розробці та адаптації універсальних методів тестування ІТ-продуктів. Такі методи мають на меті забезпечувати гнучкість та ефективність в рамках будь-якого проєкту без втрати якості, надійності та стабільності програмного продукту. Існує чимало підходів до тестування, які є універсальними для певних методологій управління, але є зовсім не відповідними для інших.

Взявши до уваги п’ять популярних підходів до тестування, а саме ISTQB, TMap, Agile Testing, DevOps Testing і Risk-Based Testing, було виявлено, що кожен підхід має свої власні переваги та недоліки відносно різних методологій управління проєктами. Наприклад, підходи ISTQB (International Software Testing Qualifications Board) та TMap (Test Management Approach) надають комплексний підхід до управління тестуванням. Вони охоплюють ширший спектр процесів тестування, починаючи від планування й закінчуючи аналізом результатів. З іншого боку Agile Testing та DevOps Testing підходи визначаються своєю гнучкістю та здатністю інтегруватися в швидко змінюючи середовища розробки програмного забезпечення. Agile Testing зосереджується на постійній комунікації та співпраці між розробниками та тестувальниками, а DevOps Testing підкреслює автоматизацію тестування та неперервну інтеграцію. В свою чергу підхід Risk-Based Testing визначається своїм акцентом на аналізі ризиків та призначенням пріоритетів у тестуванні залежно від потенційного впливу на проєкт.

За результатами аналізу етапів цих п'яти підходів до управління тестуванням виявлено перетин таких етапів: визначення тестових вимог; планування тестування; виконання тестування; аналіз результатів та підготовка звітності.

Питання побудови універсальних етапів процесу тестування було розглянуто автором Ian Londesbrough в рамках дослідження [2]. Він визначив шість основних етапів тестування:

– етап ідеї;
– визначення вимог;
– специфікація та дизайн;
– побудова;
– впровадження;
– реалізація цінності.

Порівнюючи результати даного дослідження з визначеними чотирма етапами, що перетинаються у п'яти підходах до управління тестуванням, виявлено, що три з чотирьох етапів мають альтернативи в рамках дослідження, а один етап не має.

Крім того, розглянемо підхід до управління тестуванням STLC [1] (Software Testing Life Cycle). Цей підхід включає шість основних етапів управління тестуванням:

– аналіз вимог;
– стратегія тестування;
– створення тестових сценаріїв;
– підготовка тестового середовища;
– виконання тестів;
– завершення тестування.

На відміну від [2], підхід STLC має альтернативи для всіх чотирьох етапів, визначених у п'яти підходах до управління тестуванням. Проте два етапи STLC не містять альтернатив.

Питання пошуку єдиного універсального підходу до управління процесом тестування є актуальним, але на даний момент остаточного рішення ще не знайдено. Аналіз існуючих підходів та наявних робіт показує певний збіг обов’язкових етапів, але даний збіг не є повним. Подальший аналіз різних методологій управління тестуванням може сприяти формуванню єдиного стандарту або фреймворку, що підвищить ефективність тестування у програмній розробці.

Список використаних джерел:

1. Software Testing Life Cycle. URL: https://www.guru99.com/software-testing-life-cycle.html (дата звернення: 01.01.2024).

2. Londesbrough I. A Test Process for all Lifecycles. 2008 IEEE International Conference on Software Testing Verification and Validation Workshop. Lillehammer, 2008. P.1–4.