УДК 004.415.2

ФОРМУВАННЯ ПЕРЕВІРЮВАНИХ КРИТЕРІЇВ ПРИЙМАННЯ НА ОСНОВІ КОРИСТУВАЦЬКИХ ІСТОРІЙ

Попова М.О., к.е.н., доц.
Національний університет «Одеська політехніка»
Україна, Одеса
popova.m.o@op.edu.ua

Анотація. *Запропоновано процедуру формування перевірюваних критеріїв приймання з користувацьких історій і бізнес-правил. На аналітичному прикладі показано визначення очікуваних результатів, альтернативних і граничних сценаріїв. Установлено зв'язки між історією, правилами та перевірками. Результати придатні для рецензування вимог.*

Ключові слова: *інженерія вимог; користувацькі історії; критерії приймання; перевірюваність вимог; простежуваність*.

Вступ. Користувацька історія стисло описує потребу певної ролі та очікувану цінність програмної функції. Проте формулювання «як користувач, я хочу виконати дію, щоб отримати результат» зазвичай не визначає всіх умов, за яких функцію можна вважати реалізованою належним чином. Якщо ці умови залишаються неузгодженими, розробник, тестувальник і замовник можуть по-різному оцінювати одну реалізацію. Перевірювані критерії приймання потрібні для встановлення спільного розуміння очікуваної поведінки до початку її реалізації.

У дослідженні G. Lucassen та співавторів якість користувацьких історій розглядається через синтаксичні, семантичні й прагматичні характеристики; критерії приймання є складником роботи з історією поряд із її текстом та обговоренням [1]. Сучасні підходи розширюють можливості автоматизації: SimAC використовує великі мовні моделі для імітації співпраці під час уточнення історій [2], а M. Ferreira та співавтори досліджують перехід від історій до сценаріїв приймання і виконуваних тестів [3]. Однак сформований текст ще потребує перевірки відповідності погодженим правилам і наявності однозначного способу оцінювання результату. Формальна впорядкованість опису сама по собі цих властивостей не забезпечує.

Метою дослідження є підвищення перевірюваності вимог до програмного забезпечення та зменшення неоднозначності їх приймання шляхом послідовного уточнення користувацьких історій, встановлення зв'язків із погодженими бізнес-правилами й формування критеріїв з однозначно визначеними умовами та спостережуваними результатами.

Основна частина. У роботі виконано аналітичне моделювання переходу від користувацької історії до критеріїв приймання. Узагальнено вимоги до їх перевірюваності, сформовано послідовність уточнення та застосовано її до сконструйованого прикладу.

Критерій приймання визначає умову, виконання якої необхідне для погодження певної історії. Тестовий сценарій конкретизує вихідні дані, дії та очікувані результати для перевірки цієї умови. Один критерій може перевірятися кількома сценаріями, зокрема для різних граничних значень. Загальні умови готовності роботи, наприклад проходження перевірки коду, застосовуються до багатьох історій і не замінюють специфічних для них критеріїв поведінки.

Запропонована процедура починається з виділення ролі, цілі та користувацької цінності. Далі встановлюються передумови виконання дії, обмеження предметної області та результати, які може спостерігати користувач або інша зацікавлена сторона. Джерелами уточнення є обговорення із замовником, бізнес-правила та пов'язані вимоги. Невідомі значення й непогоджені обмеження фіксуються як відкриті питання. Їх не слід довільно підставляти лише для завершення формулювання критерію.

На наступному етапі критерії описуються через передумову, подію та очікуваний результат. Для сценарного подання використовується структура Given-When-Then мови Gherkin: Given визначає вихідний контекст, When – дію або подію, Then – очікуваний результат [4]. Українською її можна передати як «За умови – Коли – Тоді». Така структура допомагає відокремити стан системи від дії користувача, але не робить опис автоматично виконуваним: для цього потрібні тестові дані та програмне зіставлення кроків із діями перевірки.

Для кожного критерію перевіряються однозначність передумови, можливість відтворення події та визначеність результату. Опис «система коректно обробляє запит» непридатний для приймання без пояснення того, що саме вважається коректним. Натомість можна встановити факт створення запису, доступність результату користувачу, незмінність даних у разі відмови або конкретну умову відхилення операції. Числові межі швидкодії чи допустимого обсягу даних включаються лише за наявності погодженого джерела.

Розглянемо історію І-01: «Як студент, я хочу записатися на вибіркову дисципліну, щоб сформувати індивідуальний навчальний план». Для аналітичного прикладу задано чотири умовні бізнес-правила: П-01 – запис доступний автентифікованому студенту протягом періоду реєстрації; П-02 – кількість записаних студентів не перевищує місткості групи; П-03 – повторний запис того самого студента на ту саму дисципліну не створюється; П-04 – запис допускається до моменту завершення реєстрації, а в сам цей момент і пізніше відхиляється. Ці правила є вихідними припущеннями прикладу, а для реального проєкту потребують погодження.

За правилом П-01 визначено критерій К-01 успішного запису: за умови автентифікованого студента, відкритої реєстрації, наявного місця та відсутності попереднього запису, коли студент підтверджує вибір дисципліни, система створює один запис і відображає дисципліну в його навчальному плані. Передумови охоплюють пов'язані правила П-02–П-04. Результат перевіряється через стан плану й відсутність дублювання, без прив'язування критерію до конкретної структури бази даних.

Критерій К-02 уточнює відмову за П-02: коли за інших допустимих умов студент підтверджує запис у заповнену групу, система не створює запису, не змінює склад групи та повідомляє про відсутність місць. Для перевірки межі додатково задається випадок одного вільного місця. Якщо на нього одночасно претендують два студенти, допускається лише один успішний запис; другий отримує відмову через вичерпання місткості. Така перевірка випливає з обмеження кількості місць, а не зі способу реалізації блокувань.

Критерій К-03 відповідає П-03: за наявного запису, коли той самий студент повторно підтверджує вибір цієї дисципліни, кількість його записів залишається рівною одному, склад групи не змінюється, а система повідомляє, що запис уже існує. Для К-04 за правилом П-04 встановлено відмову при зверненні в момент завершення реєстрації та після нього. Сценарії відтворюють стан безпосередньо перед межею, на межі та після неї з урахуванням погоджених часового поясу, джерела часу й точності його вимірювання. Без цих уточнень часовий критерій залишається неповним.

Простежуваність забезпечується збереженням зв’язку «історія – бізнес-правило – критерій – тестовий сценарій». Наприклад, К-02 пов’язується з П-02 і перевірками заповненої групи, одного вільного місця та одночасних запитів. Якщо місткість або порядок зарахування змінюються, цей зв’язок дає змогу визначити критерії й сценарії, які потребують перегляду. Кожне уточнення має зберігати редакцію та підставу погодження.

Завершальний перегляд охоплює основний перебіг подій, відмови й граничні стани. Для кожного погодженого правила встановлюється критерій і спосіб перевірки. Кількість критеріїв не є самостійним показником якості, а зв’язність із правилами не доводить повноти самих правил. Невирішені питання повертаються до уточнення. У навчанні процедуру можна застосовувати для взаємного рецензування: один студент формує критерії, інший намагається побудувати перевірку й аргументує виявлені недоліки.

Висновки. Запропоновано процедуру формування перевірюваних критеріїв приймання, що поєднує уточнення користувацької потреби, встановлення бізнес-правил, сценарний опис і перевірку простежуваності. На аналітичному прикладі сформовано критерії успішного запису, відмови через вичерпання місць, запобігання дублюванню та дотримання строку реєстрації. Показано, як із погоджених обмежень отримати граничні перевірки, зокрема для одночасних запитів. Практична цінність результату полягає у створенні спільної основи для уточнення вимог і підготовки приймальних перевірок. Ефект застосування процедури в реальних проєктах має бути оцінений емпірично.

Використання інструментів ШІ. ChatGPT (OpenAI, модель GPT-6) використано для мовного редагування українського й англійського текстів. Штучний інтелект (ШІ) не використовувався для отримання емпіричних даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Lucassen, G., Dalpiaz, F., van der Werf, J. M. E. M., & Brinkkemper, S. (2016). Improving agile requirements: The Quality User Story framework and tool. *Requirements Engineering, 21*(3), 383-403. https://doi.org/10.1007/s00766-016-0250-x

[2] Li, Y., Keung, J., Yang, Z., Ma, X., Zhang, J., & Liu, S. (2024). SimAC: Simulating agile collaboration to generate acceptance criteria in user story elaboration. *Automated Software Engineering, 31*(2), Article 55. https://doi.org/10.1007/s10515-024-00448-7

[3] Ferreira, M., Viegas, L., Faria, J. P., & Lima, B. (2025). *Acceptance test generation with large language models: An industrial case study* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2504.07244

[4] Cucumber. (n.d.). *Gherkin reference*. https://cucumber.io/docs/gherkin/reference/

Popova M.O.

Formulating verifiable acceptance criteria from user stories

Abstract*. A procedure for deriving acceptance criteria from user stories and agreed business rules is proposed. Conditions for criterion verifiability and traceability are defined. A constructed elective course enrolment example illustrates the formulation of main, alternative and boundary scenarios. The results support requirements analysis and acceptance testing preparation.*

Keywords *requirements engineering; user stories; acceptance criteria; requirements verifiability; traceability.*