SOFTWARE AND INFORMATION SUPPORT FOR COMPUTERIZED SYSTEMS OF BUSINESS PROCESS MANAGEMENT

DOI https://doi.org/10.30525/978-9934-26-506-8-102

ANALYSIS OF APPROACHES TO DEFINING SOFTWARE REQUIREMENTS QUALITY

АНАЛІЗ ПІДХОДІВ ДО ВИЗНАЧЕННЯ ЯКОСТІ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Gobov D.A.,
PhD (Engineering), Associate Professor, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

Гобов Д.А.,
к.т.н., доцент, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

Shevchenko N.Yu.,
PhD (Economics), Associate Professor, LLC "Technical university "Metinvest polytechnic", Zaporizhzhia, Ukraine

Шевченко Н.Ю.,
к.е.н., доцент, ТОВ «Технічний університет «Метінвест політехніка», м. Запоріжжя, Україна

Формалізація вимог до програмного забезпечення є критичним аспектом забезпечення його якості продукту та якості під час застосування. У зв'язку зі зростанням складності інформаційних систем, а також підвищенням вимог до якості програмних продуктів, роль бізнес-аналізу в процесі виявлення, аналізу, документування та моделювання вимог набуває особливої актуальності. Від якості виконання задач бізнес-аналізу залежить рівень структурованості вимог, точність постановки задач команді розробки, а також можливість ефективного контролю змін. Це дозволяє уникнути проблем, пов'язаних з «розповзанням вимог» та затримками в розробці, забезпечуючи надійну основу для реалізації успішних проєктів з розробки програмного забезпечення. Залежно від обраної методології розробки, характеру проєкту та вимог замовника, для документування вимог можуть

застосовуватись різні підходи. Ключові підходи до формалізації вимог і забезпечення їхньої якості були визначені авторами у роботі [1].

Традиційні методи, наприклад, Waterfall, ґрунтуються на детальному документуванні вимог у вигляді специфікацій, таких як Software Requirements Specification (SRS). Згідно з IEEE Std 830-1998, SRS повинна відповідати низці критеріїв якості, включаючи завершеність, послідовність, можливість перевірки, модифікованість та відстежуваність. Деталізована структура специфікацій сприяє створенню чіткого орієнтира для розробки ПЗ і мінімізації непорозумінь між представниками замовника та командою розробників. Недоліком цього підходу є його обмежена адаптивність до змін, що може знижувати ефективність за динамічних умов проєкту.

Гнучкі методології, такі як Scrum та SAFe, передбачають створення вимог у форматі коротких описів, зокрема, користувацьких історій (User Story), які відображають потреби користувачів і бізнес-цілі. User Story формують беклог продукту (Product Backlog) і використовуються для швидкої адаптації до змін у проєкті. Критерії якості вимог в форматі User Story базуються на принципі INVEST: незалежність (Independent), обговорюваність (Negotiable), цінність (Valuable), оцінюваність (Estimable), малий розмір (Small) і можливість перевірки (Testable). Цей підхід забезпечує гнучкість вимог, однак іноді потребує додаткового документування у випадках підвищених вимог до звітності чи регуляторного контролю.

На проєктах, що потребують підвищеної надійності та точності використовуються формальні специфікації. Підхід базується на використанні формальних мов, таких як Vienna Development Method, Spec# та RAISE Specification Language, що забезпечують точність і можливість логічного доведення коректності вимог. Формальні специфікації застосовуються у критичних системах, де точність і однозначність є пріоритетними. Перевагою цього підходу є можливість проведення формальної верифікації вимог, що знижує ймовірність неузгодженостей і забезпечує відповідність високим стандартам надійності.

В процесі забезпечення якості програмного забезпечення широко використовуються міжнародні стандарти, які окреслюють ключові характеристики, необхідні для оцінки та контролю якості програмних продуктів. Відомі стандарти, такі як ISO/IEC 25019:2023, ISO/IEC 5055:2021 та ISO/IEC/IEEE 12207:2017, містять концептуальні основи, що дозволяють розробникам формалізувати вимоги до якості,

відстежувати їхню відповідність та забезпечувати очікування зацікавлених сторін [2].

ISO/IEC 25019:2023 – стандарт SQuaRE визначає якість програмного забезпечення за кількома вимірами, зокрема внутрішньою, зовнішньою якістю та якістю під час використання. Внутрішня якість описує характеристики, які можна оцінити до етапу розгортання, зовнішня якість стосується продуктивності ПЗ у роботі, а якість під час використання охоплює можливість програмного забезпечення відповідати очікуванням кінцевих користувачів під час експлуатації. Стандарт також встановлює 8 характеристик якості, таких як функціональність, надійність та зручність використання, що допомагає визначити основні аспекти забезпечення якості в різних умовах застосування ПЗ.

Стандарт ISO/IEC 5055:2021 запроваджує поняття "структурна якість", що описує статичні характеристики програмного забезпечення та вимірює їхню відповідність заявленим і прихованим потребам. Зазначений в стандарті підхід спрямований на автоматизоване вимірювання якості коду, включаючи перевірку його складності, безпеки та надійності, що є важливим для великих проєктів, де кількість змін коду та ризик помилок є значними.

Стандарт життєвого циклу ПЗ ISO/IEC/IEEE 12207:2017 акцентує увагу на забезпеченні якості з самого початку процесу розробки шляхом чіткої ідентифікації та аналізу вимог. Це підкреслює важливість залучення бізнес-аналітиків та інженерів вимог для визначення чітких вимог на ранніх етапах, що створює передумови для ефективної верифікації та валідації на більш пізніх етапах.

Незалежно від обраного підходу до документування вимог якість вимог до програмного забезпечення значною мірою залежить від ретельного рецензування, яке здійснюється шляхом перегляду та обговорення вимог серед зацікавлених сторін, зокрема бізнес-аналітиків, розробників, тестувальників та кінцевих користувачів. Рецензування дозволяє виявити недоліки, двозначності або суперечності, які можуть вплинути на процес розробки чи подальшу експлуатацію програмного продукту. Такий спільний аналіз сприяє підвищенню чіткості вимог та їхньої відповідності початковим бізнес-цілям проєкту, знижуючи ризики внесення змін на пізніших етапах розробки, що є критично важливим для дотримання бюджету та строків.

Важливими етапами контролю якості є також валідація та верифікація. Валідація забезпечує перевірку вимог на предмет відповідності очікуванням кінцевих користувачів та загальним цілям

бізнесу. Її основна мета – гарантувати, що вимоги справді вирішують конкретні проблеми чи задовольняють потреби користувачів, що значно знижує ризик невдоволення кінцевих користувачів після впровадження продукту. Верифікація орієнтована на технічний аспект вимог: вона визначає, чи достатньо вимоги деталізовані, послідовні й логічні, щоб їх можна було використовувати як базу для подальшої розробки. Разом ці процеси дозволяють забезпечити повноту, узгодженість і технічну доцільність вимог, що є важливими умовами для створення якісного програмного продукту.

Однак зазначимо, що ефективність формалізації вимог до програмного забезпечення залежить не лише від обраної методології та рівня адаптивності проєкту до змін, але й від чіткого визначення архітектури вимог як бізнес-аналітичного продукту, що забезпечує цілісну структуру для систематизації вимог на різних рівнях деталізації [1]. Архітектура вимог дозволяє інтегрувати підходи до узгодження, повноти та якості вимог, створюючи єдину інформаційну базу для зацікавлених сторін та команди розробників. Зважаючи на те, що потреби стейкхолдерів можуть змінюватися протягом життєвого циклу проекту, архітектура вимог виконує важливу роль у підтримці послідовності та гнучкості вимог, зберігаючи їхню відповідність початковим бізнес-цілям проекту.

Перелік використаних джерел

1. Гобов Д. А., Шевченко Н. Ю. Визначення архітектури вимог до IT-рішення як бізнес-аналітичного продукту. Сучасний стан наукових досліджень та технологій в промисловості. 2024. № 1 (27). С. 26–38. DOI: https://doi.org/10.30837/ITSSI.2024.27.026.

2. Gobov D.A., Zuieva O. V. Examining software quality concept: business analysis perspective. Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. 2023. № 2 (10). С. 9–14. DOI: https://doi.org/10.20998/2079-0023.2023.02.02.