УДК 004.9: 005.334

ОСОБЛИВОСТІ РИЗИК-МЕНЕДЖМЕНТУ ПРОТЯГОМ ЖИТТЄВОГО ЦИКЛУ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

І. М. Лях, Ю. В. Кіш

ДВНЗ «Ужгородський національний університет», пл. Народна, 3, Ужгород, 88000, Україна

Розкрито сучасне розуміння понять «ризик» та «ризик-менеджмент». Досліджено подвійну природу ризику, в основі якої лежить невизначеність. Виявлено передумови застосування підходу до забезпечення якості програмного забезпечення на основі ризиків. Проаналізовано основні етапи управління ризиками на проєкті з розробки ПЗ, їхні особливості та артефакти, що потрібно отримати при роботі з невизначеністю. Запропоновано власну шкалу оцінки потенційних загроз, які можуть виникнути на проєкті, що дає змогу чітко пріоритезувати ризики як в плані першочерговості роботи з ними, так і технічної складності їх вирішення. Опрацьовано основні стратегії роботи з ризиками якості, що передбачають не лише заходи реагування на потенційні небезпеки, а й активності для розвитку позитивної складової невизначеності. Визначено основні переваги, які можна отримати від впровадження системи управління якістю, що заснована на ризиках.

Ключові слова: *ризик, ризик-менеджмент, тестування на основі ризиків, ідентифікація ризиків, аналіз та оцінка ризиків, пом'якшення ризиків, моніторинг та контроль ризиків, матриця ризиків продукту.*

Постановка проблеми. Невизначеність — фактор, що найбільше непокоїть бізнес. Її непросто запланувати, нелегко спрогнозувати та оцінити ступінь впливу на розвиток проєкту. Невизначеність і лежить в основі ризику. У наш час ризик-менеджмент став життєвого необхідною складовою частиною процесів управління розробки ІТ-продуктів. Розроблено чимало концепцій, теорій, навіть робочих моделей з управління ризиками. Вміння працювати з ризиком є головним викликом, що необхідно подолати задля успішної реалізації проєкту.

Аналіз останніх досліджень та публікацій. Розуміння ризику може суттєво відрізнятися в різних спеціалістів. ISTQB, провідна організація у сфері стандартизації та сертифікації QA менеджмент процесів, визначає ризик як «фактор, який може призвести до негативних наслідків у майбутньому» [1]. На думку класика теорії тестування Сема Кенера, ризик — це «можливість зазнати шкоди або втрати» [2]. Джеймс Бах, батько підходу Rapid Software Testing, у своїй праці «How to conduct heuristic risk analysis» зазначає, що «ризик це проблема, що може статися» [3]. Згідно з поданими визначеннями, ризик має явно негативний характер, хіба що Джеймс Бах надає відносно нейтральне визначення. З іншого боку, останнім часом

спостерігається тенденція до переосмислення сутності ризику. За правильного підходу робота з невизначеністю може дати й позитивний результат. Ризик є не тільки джерелом проблем, а й може стати джерелом можливостей. Розглянемо визначення Томаса Гемілтона, де «ризик — це виникнення невизначеної події з позитивним або негативним впливом на показники успіху проекту, що піддаються вимірюванню» [4]. Схожої думки дотримуються й Омдев Дагія та Камна Соланкі. Вони виділяють «позитивні» та «негативні» ризики. «Позитивні ризики» сприяють стабільності бізнесу і називаються можливостями. «Негативні ризики» називаються загрозами і їх необхідно усунути або мінімізувати, щоб проєкт був успішним [5]. Головним підтвердженням того, що розуміння природи ризику змінюється, є визначення, що міститься у Довіднику з управління проектами, всесвітньовідомому PMBOK Guide, — «ризик — невизначена подія чи умова, що у разі настання матиме позитивний чи негативний вплив на одну чи більше цілей проекту» [6].

Перейдемо до розгляду поняття «ризик-менеджмент». Міжнародний стандарт якості ISO 31000 дає таке визначення: «набір скоординованих заходів, які дозволяють керувати організацією та контролювати її щодо ризиків» [7]. ISTQB визначає ризик-менеджмент ще простіше — «процес управління ризиками» [8]. Більш повним є визначення, яке надали спеціалісти GeeksforGeeks, — «послідовність кроків, які допомагають команді розробки програмного забезпечення зрозуміти, проаналізувати та керувати невизначеністю» [9].

Мета статті — розкрити суть поняття ризику в сучасній науці, а також розглянути різні підходи до визначення цього поняття серед фахівців із забезпечення якості. Визначити роль ризик-менеджменту у побудові та функціонуванні системи менеджменту якості на проєкті.

Виклад основного матеріалу дослідження. SDLC (Software Development Life Cycle), або життєвий цикл розробки програмного забезпечення, добре відомий, а про життєвий цикл тестування програмного забезпечення STLC (Software Testing Life Cycle) нерідко забувають, хоча він є не менш важливим [10].

Рис. 1. Життєвий цикл тестування програмного забезпечення

Надзвичайно важливо починати роботу над ризиками якомога раніше. Управління ризиками починається вже на етапі аналізу вимог і не закінчується до

завершального етапу розробки ПЗ. Існує спеціальний підхід до забезпечення якості програмного забезпечення — RBT (Risk Based Testing). RBT — це підхід до тестування, що визначає пріоритетність функціоналу для перевірки, на основі ймовірності виникнення та ступеня впливу ризику на продукт. Фактори, які вказують на необхідність застосування RBT на проєкті: обмеженість ресурсів, складність проєкту, критично важливий софт, застарілі системи, дефіцит часу [11].

Перейдемо до розгляду основних етапів, які необхідно пройти, щоб ефективно впровадити підхід до забезпечення якості ПЗ, заснований на оцінці ризиків.

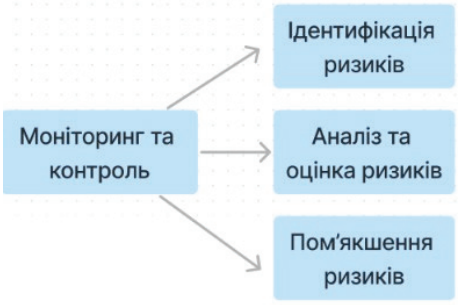

Рис. 2. Основні етапи ризик-менеджменту при RBT підході

Ідентифікація ризиків. Ось декілька способів ідентифікувати ризики: спеціальні воркшопи, експертні думки, мозкові штурми із залученням стейкхолдерів, аналіз аналогічних продуктів та проєктів.

Оцінка ризиків. Під час оцінки ризиків необхідно:

– провести класифікацію ризиків;
– визначити ймовірність виникнення ризику;
– оцінити потенційний вплив ризику;
– призначити особу, що відповідальна за ризик.

Раціональним є підхід, коли ризики класифікуються за різними об'єктами впливу: Severity — технічний вплив на систему, Priority — бізнес-вплив на продукт.

Існують різні підходи до градації Severity. Розглянемо один з найбільш поширених: S1 Blocker — блокуюча помилка, що приводить ПЗ в неробочий стан; S2 Critical — неправильно працююча ключова бізнес-логіка; S3 Major — не критичний, є інша можливість для роботи з функціоналом; S4 Minor — незначна помилка, що не порушує бізнес-логіку частини ПЗ; S5 Trivial — тривіальна помилка, яка не стосується бізнес-логіки ПЗ [12].

Priority підкреслює важливість послідовності виконання завдань чи усунення недоліків. Що вище пріоритет, то важливіше негайно виправити баг.

Розглянемо градацію Priority: P1 High — цей баг є критичним для проєкту; P2 Medium — баг потрібно виправити в обов'язковому порядку, P3 Low — баг не критичний, його виправлення не є терміновим. Зазвичай ризики з високим рівнем Severity будуть мати і високий пріоритет [12].

Так як будь-який комерційний продукт призначений на продаж, тому бізнес-підхід (Priority) буде мати вирішальне зазначення. Доцільним є виділяти Severity

в рамках Priority. Наприклад, P1 High S1 Blocker; P2 Medium S3 Major; P3 Low S2 Critical.

Пом'якшення впливу ризиків. Після ідентифікації та оцінки ризиків з ними потрібно працювати та пом'якшувати. Пом'якшення відбувається через тест-дизайн, виконання тестів та аналізу результатів їх виконання. Основні зусилля тестування зосереджені на ризиках високого та середнього пріоритету. Це допомагає створювати програмне забезпечення з найвищими шансами на успіх як з погляду клієнтів, так і з технічного погляду.

Перед тим як почати втілювати цей підхід на практиці, необхідно отримати відповіді на такі запитання:

– Які артефакти та документи проєкту потрібно переглянути?
– Наскільки досвідчені мають бути тестувальники?
– Наскільки незалежними мають бути тестувальники?
– Який обсяг повторного тестування?
– Який ступінь необхідності та скоуп регресійного тестування?

Побудуємо матрицю ризиків на основі ймовірності їх виникнення та оцінки їх потенційного впливу на продукт.

Рис. 3. Матриця ризиків продукту

Як бачимо, ризики можна поділити на 4 групи.

Група 1. Ризики з високим рівнем впливу та високою ймовірністю настання — обов'язкові для тестування насамперед.

Група 2. Ризики з високим рівнем впливу та низькою ймовірністю настання — неодмінно мають бути протестовані.

Група 3. Ризики з низьким рівнем впливу та високою ймовірністю настання — можуть бути протестовані найперше.

Група 4. Ризики з низьким рівнем впливу та низькою ймовірністю настання — можна проігнорувати.

Моніторинг і контроль ризиків містить ідентифікацію, аналіз та планування нових ризиків, стеження за виявленими та тими, які увійшли до списку для постійного нагляду, перевірки та виконання заходів реагування на ризики, а також оцінку їх ефективності протягом усього життєвого циклу проєкту.

Розглянемо основні інструменти та методи управління ризиками. План управління ризиками визначає, як буде виконуватися процес управління ризиками вашого проєкту, а також містить методологію, реєстр ризиків, структуру їх розподілу, матрицю оцінки ризиків, бюджет, розподіл обов'язків всередині команди, терміни.

План реагування на ризики — це документ управління проєктом, що пояснює стратегії зменшення ризиків, які будуть використовуватися для ризик-менеджменту на проєкті.

Стратегії реагування на негативні ризики (загрози): уникнення ризику, пом'якшення впливу ризиків, передача роботи з усунення ризику проєкту третій стороні, прийняття ризику.

Стратегії реагування на позитивні ризики (можливості): експлуатація можливостей; активностей, що збільшують ймовірність появи позитивного ризику на проєкті; розповсюдження інформації про ризики.

Стратегії реагування, які можна застосувати як для позитивних, так і для негативних ризиків: прийняття ризику, ескалація.

Реєстр ризиків, List of Risks (оновлений). Оновлений реєстр ризиків містить результати перегляду ризиків, аудиту та періодичної перевірки ризиків, що передбачають фактичні результати ризиків проєктів та результати реагування на ці ризики.

Change Requests. Запити на зміни виникають у результаті необхідності зміни плану управління проєктом у відповідь на ризик. Схвалені запити зміни оформляються документально.

Під час впровадження RBT на проєкті можливі такі перешкоди: відсутність належного планування, труднощі з визначенням ризику, брак ресурсів, недостатнє тестове покриття, відсутність узгодженості.

Насамкінець розглянемо, які переваги можна отримати від впровадження системи управління якістю, що заснована на ризиках.

1. RBT більше зосереджується на потенційних критичних ризиках, які безпосередньо впливають на клієнтів. Це покращує продуктивність бізнесу, а також покращуються відгуки користувачів.

2. Підвищення якості програмного забезпечення. Тестування на основі оцінки ризиків зосереджується на виявленні ризиків високого пріоритету і гарантує, що найважливіші функції тестуються першими. Отже, програмне забезпечення можна випускати з упевненістю в тому, що фундаментальні та орієнтовані на клієнта функції відповідають очікуванням якості.

3. Більш структуроване тестування. Коли ризики ідентифіковано та їх вплив кількісно визначено, стає легше вирішити, що тестувати, з чого почати та припинити тестування. Це забезпечує структуру, необхідну для організації тисяч тестів у кожному окремому проєкті розробки.

4. Оптимізація використання ресурсів. Насправді, планування контролю якості залежить від того, скільки годин має команда, а не від того, скільки функцій і доповнень коду їм потрібно перевірити. Тестування лише тих компонентів, які

знаходяться в зоні найвищого ризику, є чудовим способом максимізувати цінність роботи тестувальників.

5. Оптимізація автоматизації. Може здатися, що автоматизація тестування обходить обмеження на людино-години, але комусь однаково потрібно писати, підтримувати та оновлювати автоматизовані тести. Знання про те, на що орієнтуватися, стане ефективним доповненням до списку пріоритетів ручного тестування.

6. Раннє виявлення проблем. Перше тестування критичних компонентів підвищує ймовірність завчасного виявлення серйозних багів.

7. Додаткова захищеність щодо відповідності нормативним вимогам є головною перевагою для будь-якої компанії, яка працює в специфічному домені. Регулятори часто очікують, що ваш підхід до тестування програмного забезпечення буде обґрунтованим. RBT — це ефективний підхід, який надає додатковий захист від неприємностей, якщо щось не вийде.

Висновки. Розкрито подвійну природу ризику. Досліджено різні підходи до визначення поняття ризику у сучасній науці. Виявлено роль ризик-менеджменту у побудові та функціонуванні системи менеджменту якості на проєкті, а також особливості його застосування під час різних стадій процесу тестування продукту. Виділено та проаналізовано ключові етапи впровадження системи забезпечення якості програмного забезпечення. Побудовано матрицю ризиків IT-продукту. Видозмінено традиційні підходи до встановлення системи оцінювання впливу потенційних загроз на проєкт. Для подолання протиріч між технічними та бізнес-цілями, запропоновано новий підхід, що дає змогу чітко пріоритезувати ризики як в плані першочерговості роботи з ними, так і технічної складності вирішення проблеми з ризиками. Визначено основні переваги для проєкту від впровадження системи управління якістю, що заснована на роботі з ризиками.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ISTQB Glossary. ISTQB. URL: https://glossary.istqb.org/en_US/term/risk.
2. Kaner C. QA Risks Basics. URL: https://kaner.com/pdfs/QAIRiskBasics.pdf.
3. Software Testing for Serious People. *Satisfice*. URL: https://www.satisfice.com/download/heuristic-risk-based-software-testing.
4. Risk-based testing. URL: https://en.wikipedia.org/wiki/Risk-based_testing.
5. Dahiya O., Solanki K., Dhankhar A. Risk-Based Testing: Identifying, Assessing, Mitigating & Managing Risks Efficiently in Software Testing. *International Journal of Advanced Research in Engineering and Technology (IJARET)*. 2020. № 11 (3). Pp. 192−203. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3565202.
6. A Guide to the Project Management Body of Knowledge, 250 – Newton Square, Pennsylvania, Project Management Institute Inc., 2021.
7. ISO, ISO 31000: risk management – Guidelines, Geneva, Switzerland, 2018. URL: https://bit.ly/3cmnZUF.
8. ISTQB Glossary. ISTQB. URL: https://glossary.istqb.org/en_US/term/risk-management-3-1.
9. Risk-Based Testing Approach: Benefits and Use Cases. Inoxoft. URL: https://inoxoft.com/blog/risk-based-testing-approach-benefits-and-use-cases.

10. Massoa J., Pinoc F. J., Pardob C., Garcíaa F., Piattini M. Risk management in the software life cycle: A systematic literature review. *Computer Standards & Interfaces*. 2020. № 71. URL: https://www.sciencedirect.com/science/article/abs/pii/S0920548919300881.
11. Tavares B., Carlos da Silva. Risk management analysis in Scrum software projects. *International Transactions in International Research*. 2019. № 5. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/itor.12401.
12. Honest N. Role of Testing in Software Development Life Cycle. *International Journal of Computer Sciences and Engineering.* 2019. № 5. URL: https://www.ijcseonline.org/full_paper_view.php?paper_id=4332.

REFERENCES

1. ISTQB Glossary. ISTQB. Retrieved from https://glossary.istqb.org/en_US/term/risk (in English).
2. Kaner, C. QA Risks Basics. Retrieved from https://kaner.com/pdfs/QAIRiskBasics.pdf (in English).
3. Software Testing for Serious People: Satisfice. Retrieved from https://www.satisfice.com/download/heuristic-risk-based-software-testing (in English).
4. Risk-based testing. Retrieved from https://en.wikipedia.org/wiki/Risk-based_testing (in English).
5. Dahiya, O., Solanki, K., & Dhankhar, A. (2020). Risk-Based Testing: Identifying, Assessing, Mitigating & Managing Risks Efficiently in Software Testing: International Journal of Advanced Research in Engineering and Technology (IJARET), 11 (3), 192−203. Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3565202 (in English).
6. A Guide to the Project Management Body of Knowledge, 250 – Newton Square, Pennsylvania, Project Management Institute Inc., 2021 (in English).
7. ISO, ISO 31000: risk management – Guidelines, Geneva, Switzerland, 2018. Retrieved from https://bit.ly/3cmnZUF (in English).
8. ISTQB Glossary. ISTQB. Retrieved from https://glossary.istqb.org/en_US/term/risk-management-3-1 (in English).
9. Risk-Based Testing Approach: Benefits and Use Cases. Inoxoft. Retrieved from https://inoxoft.com/blog/risk-based-testing-approach-benefits-and-use-cases (in English).
10. Massoa, J., Pinoc, F. J., Pardob, C., Garcíaa, F., & Piattini, M. (2020). Risk management in the software life cycle: A systematic literature review: Computer Standards & Interfaces, 71. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0920548919300881 (in English).
11. Tavares, B., & Carlos da Silva. (2019). Risk management analysis in Scrum software projects: International Transactions in International Research, 5. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/itor.12401 (in English).
12. Honest, N. (2019). Role of Testing in Software Development Life Cycle: International Journal of Computer Sciences and Engineering, 5. Retrieved from https://www.ijcseonline.org/full_paper_view.php?paper_id=4332 (in English).

doi: 10.32403/0554-4866-2023-2-86-71-78

RISK MANAGEMENT FEATURES DURING THE LIFE CYCLE OF SOFTWARE TESTING

I. M. Liakh, Y. V. Kish

Uzhhorod National University,
3, Narodna Square, Uzhhorod, 88000, Ukraine
igor.lyah@uzhnu.edu.ua,
yurii.kish@uzhnu.edu.ua

The modern understanding of the concepts of "risk" and "risk management" is revealed. The dual nature of risk, as a source of not only threats, but also opportunities for the project, based on uncertainty, is explored. The prerequisites for applying a risk-based approach to software quality assurance are identified, such as limited resources, project complexity, critical software, outdated systems, and time constraints. The main stages of risk management in a software development project are analysed — risk identification, risk assessment and analysis, risk mitigation, monitoring and control. The main tools and methods of risk management are considered. Artifacts that need to be obtained when working with uncertainty are defined – risk management plan, risk response plan, list of risks, change request. Modern approaches to risk assessment are studied. An own scale for assessing potential threats that may arise on the project is proposed, which allows clearly prioritizing risks, both in terms of the priority of working with them, and in terms of the technical complexity of their solution. A product risk matrix is constructed. Basic strategies for working with quality risks are developed, including not only measures to respond to potential dangers, but also activities for the development of a positive component of uncertainty. Typical obstacles to the implementation of testing are identified – lack of proper planning, difficulty in identifying risk, lack of resources, insufficient test coverage, lack of consistency. The main benefits that can be obtained from the implementation of a risk-based quality management system are identified, such as increased business productivity and end-user satisfaction, software quality improvement, structured testing, resource use optimization, issues early detection, additional security for compliance with regulations

Keywords: *risk, risk management, risk based testing, risk identification, risk assessment and analysis, risk mitigation, risk monitoring and control, product risk matrix.*

Стаття надійшла до редакції 22.08.2023.
Received 22.08.2023.