

УДК 004.4

**А.М. Акименко**, канд. фіз.-мат. наук

Чернігівський державний технологічний університет, м.Чернігів, Україна

## ІДЕНТИФІКАЦІЯ РИЗИКІВ ПРОГРАМНОГО ПРОЕКТУ

*Розглянуто проблеми, пов'язані з ідентифікацією ризиків на початкових стадіях життєвого циклу програмного забезпечення, а саме аналізу та проектування. Запропоновано використання мови моделювання програмних проектів UML для представлення детального опису проекту.*

### Постановка проблеми

Аналіз проектних ризиків у програмній інженерії сьогодні є однією з найпоширеніших та найважливіших проблем, що виникають у процесі розробки програмних продуктів, з кількох причин. По-перше, практично всі без виключення виробники програм мають проектно-орієнтовану бізнес-модель. По-друге, програмне забезпечення практично з самого початку було досить складним інженерним продуктом, який потребує більшої кількості людино-годин праці висококваліфікованого персоналу, що не може не викликати помилки, затримки та пов'язані з цим ризики.

Все це змушує усвідомити, що проектні ризики можна і потрібно аналізувати. В сучасному світі аналіз ризиків проекту є невід'ємною частиною обґрунтування проекту.

### Управління ризиками проекту

Наведемо визначення поняття «ризик».

*Ризик* – це ступінь небезпеки для успішної реалізації проекту. Ризик є прогнозованим, передбачуваним, і в більшості випадків його проявів, при прийнятті завчасних заходів, є можливість його уникнути. Ризик характеризується трьома факторами: ймовірністю настання події, величиною втрат або збитків у разі настання події та самою подією.

Ризики також розділяють на внутрішні та зовнішні.

*Внутрішні ризики* – це ризикові події, прояв яких залежить від внутрішніх факторів виконання проекту. Команда, що виконує проект, має можливість такі ризики контролювати та попереджати (наприклад, визначення тривалості робіт, підбір кадрів, складання кошторису витрат і т.ін.).

*Зовнішні ризики* – це ті ризикові події, які можуть проявлятися за межами компетенцій команди розробників (наприклад, вплив ринку, нове законодавство та ін.).

Інститут PMI (Project Management Institute – Інститут Управління Проектами) описує системний підхід до процесу управління ризиками, який визначено в РМВОК (Project Management Body Of Knowledge). В процесі управління ризиками виділяють шість основних підпроцесів [1]:

- планування управління ризиками;
- ідентифікація ризиків;
- якісний аналіз ризиків;
- кількісний аналіз ризиків;
- планування реакції на ризики;
- моніторинг і управління ризиками.

Оскільки метою статті є вивчення питання про можливість аналізу ризиків на ранніх стадіях розробки програмного продукту (а саме на етапах аналізу і проектування ПЗ), тому особливу увагу слід звернути на такий підпроцес, як *ідентифікація ризиків*.

### Формулювання цілей статті

Метою цієї статті є вивчення питання про ідентифікацію ризиків програмного проекту та можливість застосування загальноприйнятих засобів проектування програмних продуктів для виявлення ризиків на початкових стадіях життєвого циклу.

### Ідентифікація ризиків

Якщо розглядати ідентифікацію ризиків як процес, тоді у нього повинні бути вхідні дані та кінцевий результат. У цьому випадку на вході процесу ідентифікації ризиків будуть:

- детальний опис проекту, який включає в себе: вимоги, попередню архітектуру ПЗ, опис конфігурації апаратного забезпечення, опис програмних інструментів, які будуть використовуватися при створенні програмного забезпечення;
- всілякі оцінки термінів, часу та вартості розробки модулів і компонентів проекту;
- звіти про попередні проекти, що виконувались.

В якості результату роботи, на виході процесу ідентифікації ризиків, менеджер проекту повинен отримати наступні дані:

- джерела ризиків;
- можливі події при прояві ризиків;
- втрати в разі настання ризикової події, виражені у часі і (або) грошовому еквіваленті;
- симптоми настання ризикової події.

Для спрощення процедури ідентифікації ризиків використовується класифікація, яка була розроблена Software Engineering Institute (SEI). Класифікація складена з урахуванням типових процесів життєвого циклу (ЖЦ) ПЗ і охоплює найбільш загальні області ризиків проекту, які стосуються характеристик ПЗ, середовища та процесів розробки і обмежень проекту. Оскільки основну увагу в статті ми приділяємо аналізу ризиків проекту на ранніх стадіях ЖЦ, тому, в першу чергу, нас цікавлять внутрішні ризики, пов'язані з технічними аспектами розробки ПЗ, які наведені в таблиці 1 [2].

Таблиця 1

Класифікація ризиків SEI

Клас джерела ризику	Характеристика класу	Елемент класу	Атрибут елемента
1	2	3	4
Технічні аспекти розробки ПЗ	Пов'язана з процесами на стадіях ЖЦ ПЗ (розробка вимог, проектування, кодування, тестування та ін.), а також характеристиками ПЗ (вимоги, проекту, коду та ін.) на цих стадіях	Вимоги	Стабільність
			Повнота
			Однозначність
			Достовірність
			Реалізованість
			Новизна
		Проект	Масштабність
			Функціональність
			Складність
			Інтерфейси
			Продуктивність
			Тестованість
		Нефункціональні характеристики ПЗ	Апаратні обмеження ПЗ, що купується
			Зручність супроводу
			Надійність
			Захищеність
			Безпека
			Людський фактор Специфікації

Для збору інформації про ризики можуть бути застосовані різноманітні методики. Найбільш поширеними серед них є:

- опитування експертів;
- мозковий штурм;
- метод Дельфі;
- картки Кроуфорда.

Всі вищеназвані методики ґрунтуються на інтерв'юванні кваліфікованих фахівців, яких називають експертами, і які можуть виконати попередню оцінку програмного проєкту, виходячи зі свого досвіду, знань та інформації, яка була надана.

На наш погляд, найважливішим компонентом кожної з методик, що використовується, є детальний опис проєкту. Від якості наданої інформації будуть залежати експертні висновки, тому особливу увагу необхідно приділити правилам формування опису програмної системи. Задача не є тривіальною, але сьогодні існує кілька методик, які дозволяють її виконати з необхідним ступенем достовірності.

Перш за все, це методики, які використовуються на стадіях аналізу та проєктування програмного забезпечення. Найпоширенішими можна вважати:

- діаграми структурного системного аналізу (ER-діаграми, SADT, IDEF);
- мова візуального проєктування UML.

UML на цей час фактично визнано стандартним засобом опису проєктів, які створюються за допомогою об'єктно-орієнтованих технологій. Враховуючи вищесказане, можна вважати найбільш доцільним використання мови UML для опису програмних систем як вхідної інформації для процесу ідентифікації ризиків.

Розглянемо питання застосування діаграм UML більш детально. Як відомо, модель складної системи, що створюється за допомогою UML, описує майбутній програмний продукт та має чотири складові частини:

- логічне представлення;
- представлення реалізації;
- представлення процесу функціонування;
- представлення розміщення.

Кожна складова включає в себе відповідний перелік діаграм [3], що відображають певний аспект моделі програмної системи. Відповідно до потреб експертів, діаграми використовуються для детального опису проєкту (табл. 2).

Таблиця 2

Використання діаграм UML

Назва діаграми UML	Опис проєкту
Діаграма Use Case	вимоги до програмного проєкту
Діаграма класів	попередня архітектура програмної системи
Діаграми поведінки, діаграми взаємодії	опис процесу реалізації функцій програмної системи
Діаграма розміщення	опис конфігурації апаратного забезпечення
Діаграма компонентів	перелік програмних інструментів, які будуть використовуватися при створенні ПЗ

Потрібно відзначити той факт, що застосування мови опису програмного проєкту дає можливість представити експертам інформацію в чіткому структурованому вигляді, але обмежує їхні інформаційні потреби тільки технічними аспектами розробки програмного забезпечення. Тобто, інформація не є повною і не дозволяє експертам виконати ідентифікацію ризиків у повному обсязі.

Згідно з класифікацією ризиків SEI (таблиця 1), UML дає можливість оцінити повністю або частково лише вимоги до програмного проекту та його характеристики (табл.3).

Таблиця 3

*Елементи класу, що ідентифікуються за допомогою UML*

Клас джерела ризику	Елемент класу	Атрибут елемента
Технічні аспекти розробки ПЗ	Вимоги	Стабільність
		Повнота
		Однозначність
		Достовірність
		Реалізованість
		Масштабність
	Проект	Функціональність
		Складність
		Інтерфейси
		Продуктивність
		Тестованість
		Апаратні обмеження ПЗ, що купується
	Нефункціональні характеристики ПЗ	Надійність
		Захищеність
		Безпека
		Специфікації

Перелік елементів, для ідентифікації яких потрібно використовувати інші підходи, наведено в таблиці 4.

Таблиця 4

*Елементи класу, що не ідентифікуються за допомогою UML*

Клас джерела ризику	Характеристика класу	Елемент класу	Атрибут елемента
Технічні аспекти розробки ПЗ	Пов'язана з процесами на стадіях ЖЦ ПЗ (розробка вимог, проектування, кодування, тестування та ін.), а також характеристиками ПЗ (вимоги до проекту, коду та ін.) на цих стадіях	Вимоги	Новизна
		Нефункціональні характеристики ПЗ	Зручність супроводу
			Людський фактор

Таким чином, очевидно, що за допомогою UML експерт має можливість оцінити 17 із 20 атрибутів, що суттєво спрощує організацію процесу аналізу ризиків програмного проекту та відповідно підвищує якісні характеристики програмного проекту, надаючи можливість уникнути або зменшити вірогідність багатьох ризиків ще на початкових етапах життєвого циклу (аналіз вимог та проектування).

Аналогічним чином мова моделювання може бути застосована для збереження інформації про програмні продукти, що виконувалися раніше. Переваги в цьому випадку

очевидні, оскільки єдиний підхід, що застосовується розробником протягом тривалого часу, дозволяє стабілізувати та стандартизувати технологічні процеси розробки програмного забезпечення.

Під час ідентифікації ризиків експерти повинні отримати в якості вхідної інформації, крім детального опису проекту, і звіти про попередні проекти, що виконувались. Якщо розробник використовує мову моделювання, ця проблема вирішується автоматично. Достатньо підняти архівні файли і експерти отримають вичерпну інформацію про виконані раніше проекти.

Однак, слід зазначити, що мова моделювання UML не є єдиним засобом моделювання, який дозволяє описувати майбутній програмний проект. Але потрібно підкреслити, що при застосуванні об'єктної методології розробки програмних систем застосування UML дає найбільші переваги.

Крім того, серед недоліків застосування UML під час ідентифікації ризиків програмного проекту слід відзначити, що деякі технічні аспекти розробки не можуть бути представлені за допомогою UML та вивчені експертами з аналізу ризиків. До таких аспектів відносяться новизна, зручність супроводу та, очевидно, людський фактор. Ще один недолік, на який потрібно звернути увагу, це терміни, час та вартість розробки. Для представлення інформації такого змісту використовуються стандартні програмні засоби.

Незважаючи на певні недоліки, переваги, які надає мова моделювання UML, особливо під час розробки складних програмних систем, є досить суттєвими для процесу ідентифікації ризиків програмного проекту.

#### **Висновки з даного дослідження**

- використання стандартизованих засобів опису моделі програмної системи (UML) дає можливість представити детальний опис проекту в структурованому вигляді;
- застосування мови UML дає можливість зберігати інформацію про попередні проекти, що виконані раніше.

#### **Список використаних джерел**

1. Project Management Body of Knowledge (PMBOK), PMI Standard Committee.
2. Software Engineering Body of Knowledge (SWEBOOK), IEEE, 2004.
3. Акименко А.М. Використання UML під час проектування складних програмних систем / Акименко А.М. // Вісник Чернігівського державного технологічного університету. – Чернігів: ЧДТУ, 2011. – №49. – С.164-170. – (Сер. Технічні науки).