

УДК 004.412

DOI: 10.31471/1993-9981-2023-1(50)-101-112

## ПРОГНОЗУВАННЯ ЯКОСТІ ПРОГРАМНИХ ЗАСОБІВ НА ОСНОВІ АНАЛІЗУ ЯКОСТІ ВИМОГ

*М. В. Кузь<sup>1</sup>, Б. С. Незамай<sup>2\*</sup>, В. А. Ровінський<sup>1</sup>, Н. Д. Подубинська<sup>2</sup>*

<sup>1</sup> Прикарпатський національний університет імені Василя Стефаника, вул. Шевченка,  
57, м. Івано-Франківськ, Україна

<sup>2</sup> Івано-Франківський національний технічний університет нафти і газу, вул.  
Карпатська, 15, м. Івано-Франківськ, Україна

Оцінка якості програмного забезпечення є важливою складовою частиною його розробки з метою як забезпечення останньої, так і обґрунтованого порівняння програм чи їх версій. В даний час оцінка якості програмного забезпечення базується на низці міжнародних стандартів. При цьому, як правило, використовуються методи експертного оцінювання на основі метричного підходу оцінки якості програмних засобів. Прогнозування характеристик якості програмних засобів – це наступний крок забезпечення якості, що має уможливити одержання обґрунтованих досяжних характеристик якості, так і бути застосованим для маркетингових досліджень.

У роботі розглянуто існуючі підходи до прогнозного моделювання характеристик якості програмних засобів. Проаналізовано фактори, які визначають прогноз якості програмних засобів. Встановлено, що одним із основних факторів, які визначають якість розроблюваного продукту, є якість вимог до нього. Запропоновано метод прогнозування характеристик якості програмних засобів на основі характеристик якості вимог. Характеристики якості вимог встановлюються на етапі тестування (верифікації) вимог. Вихідними даними, на яких базується прогнозна модель, є характеристики якості, що одержуються методом історичних аналогій. Прогнозна модель передбачає просту лінійну залежність впливу характеристик вимог на прогнозовані характеристики програмного засобу, але при цьому враховується кількість вимог (величина проекту) та їх класифікація. Фактично, метод прогнозування полягає у встановленні впливу оцінки характеристики якості вимог на характеристики якості майбутнього проекту, і здійснюється методом експертного оцінювання. Одержані значення можуть бути використані як для обґрунтування досяжних характеристик якості, так і для встановлення тісніших “зворотніх зв’язків” з замовниками для покращення вимог до проекту.

**Ключові слова:** програмні засоби, якість, прогнозування характеристик якості, тестування вимог.

Software quality assessment is an important component of its development in order to ensure both the latest and reasonable comparison of programs or their versions. Currently, software quality assessment is based on a number of international standards. At the same time, as a rule, methods of expert evaluation are used based on the metric approach of assessing the quality of software tools. Forecasting the quality characteristics of software tools is the next step in quality assurance, which should allow both obtaining reasonable achievable quality characteristics and serving, for example, for marketing research.

The paper examines existing approaches to predictive modeling of software quality characteristics. The factors that determine the forecast of software quality are analyzed. It was established that one of the main factors that determine the quality of the developed product is the quality of the requirements for it. A method of forecasting software quality characteristics based on requirements quality characteristics is proposed. Quality characteristics of requirements are established at the stage of testing (verification) of requirements. The initial data from which the predictive model is based are the quality characteristics obtained by the method of historical analogies. The predictive model assumes a simple linear dependence of the influence of the characteristics of the requirements on the predicted characteristics of the software tool, but at the same time, the number of requirements (the size of the project) and their classification are taken into account. In fact, the forecasting method consists in the established influence of the assessment of the quality characteristics of the requirements on the quality characteristics of the future project, and is carried out by the method of expert evaluation. The obtained values can be used both to substantiate the achievable quality characteristics and to improve "feedback" with customers to improve project requirements.

**Keywords:** software tools, quality, forecasting of quality characteristics, requirements testing.

**Вступ** На даний час оцінка якості програмного забезпечення базується на низці міжнародних стандартів. В основі методів оцінки лежить метричний підхід на основі методів експертного оцінювання. Це відкриває певні можливості для спроб оцінити досяжні характеристики якості програмних засобів на етапі проектування та розроблення – тобто до прогнозування якості програмних засобів. В процесі огляду та аналізу існуючих опублікованих досліджень, які безпосередньо чи опосередковано стосуються прогнозу якості програмного забезпечення, виділено такі їх групи: аналіз ризиків, прогнозування помилок, та, власне, прогнозування характеристик якості програмного забезпечення. Описані спроби прогнозування характеристик якості програмного забезпечення містять один з трьох основних принципів: класичні методи прогнозування [1, 2], генетичні алгоритми [3, 4] або реалізації методів машинного

навчання [5, 6, 7]. Останні дозволяють прогнозувати на основі встановлення різного роду класифікацій та співвідношення розроблюваного програмного засобу з одержаною класифікацією. Найпоширеніший інструмент при цьому – методи дерев рішень та навчання штучних нейронних мереж. **Мета роботи** – проаналізувати існуючий стан прогнозування якості програмних засобів, запропонувати метод прогнозування якості програмних засобів на основі аналізу вимог.

### Аналіз сучасних закордонних і вітчизняних досліджень і публікацій.

Описані нейромереві підходи до прогнозу якості базуються на SafetyCase методології (Safety computer aided softwareengineering) [8]. На рисунку 1 зображено узагальнену модель SafetyCase методології.

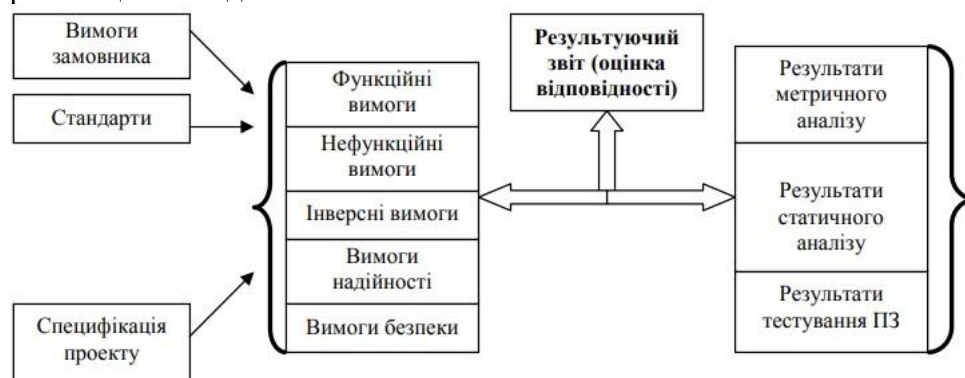


Рисунок 1 – Узагальнена модель SafetyCase методології [8]

Базовими частинами моделі при цьому є:

- профайл вимог, який формується базуючись на методиці розробки програмного забезпечення та відповідних стандартів якості для розробки програмних засобів;
- стандарти галузі, для якої розробляється програмне забезпечення;
- профайл результатів аналізу програмних засобів, який формується на основі метричного аналізу результату

розробки на поточний момент часу і результати тестування;

- оцінка відповідності результатів вимогам (профайл вимог).

Нейромеревий підхід використовується для спроб спрогнозувати складність та якість розроблюваного ПЗ на основі метрик етапу проектування і базується на навчанні багатосарових перцептронів [4, 6]. Вхідними даними при цьому є метрики складності етапу

проектування з поточними значеннями, прогнозовані значення складності етапу проектування, метрики якості етапу проектування прогнозовані значення.

Для оцінки результатів характеристик якості програмних засобів на основі роботи

з метриками проектування описано [9] інтелектуальну систему опрацювання і прогнозування складності та якості програмного забезпечення (рисунок 3).

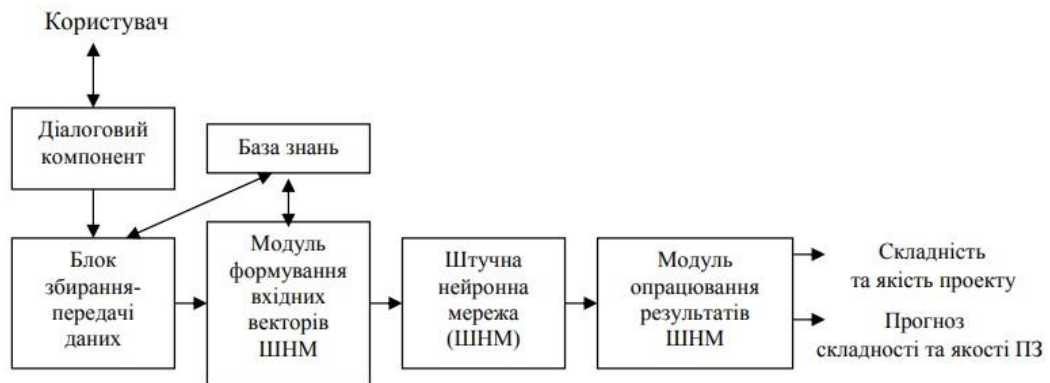


Рисунок 2 – Інтелектуальна система процесу прогнозування складності та якості програмних засобів [9]

Про описані методи використання нейромереж для прогнозування якості програмного забезпечення програмних засобів можна зробити наступні висновки:

- для навчання нейромереж буде потрібна велика база навчальних прикладів, з огляду на велике різноманіття задач, для розв'язку яких використовується програмне забезпечення;

- описані підходи часто дають прогноз оцінки якості, що базується на іншому прогнозі характеристик.

Безпосередньо стосується прогнозування якості програмного забезпечення прогнозування дефектів програмного забезпечення. Методи прогнозування, які оперують кількостями дефектів і не враховують при цьому структуру проекту, є так званими методами "чорного ящика" [10]. Методи прогнозування кількості дефектів, які враховують складність модулів і структуру програмного засобу в цілому називаються методами "білого ящика" (прозорі скриньки). Моделі росту надійності програмних засобів базуються на використанні функції надійності. Під

надійністю програмних засобів в даному випадку розуміють ймовірність того, що система за певних умов навантаження впродовж певного часу буде працювати певним чином. Для одержання цієї функції найчастіше використовують експоненційний розподіл. Параметри розподілу обираються на основі попереднього досвіду та поточної інформації про виявлені дефекти [10, 11].

Прогнозування кількості дефектів методом експертних оцінок може бути як простим ( експрес методом), так і складним в реалізації з задіяванням опитування значної кількості спеціалістів, які беруть участь у проекті ( на основі методів Делфі). Ідея методу прогнозування кількості дефектів в модулях програмних засобів полягає в тому, що в основу прогнозу покладається думка спеціалістів, що базується на їх професійному практичному досвіді роботи над поточним та аналогічними проектами. Даний метод корисний у випадку, коли прогнозування кількості дефектів має бути виконано на рівні всього проекту чи його великих частин [12]. Недоліком такого підходу прогнозування кількості дефектів є його

суб'єктивність та неможливість масштабувати на менші частини проекту.

Метод історичних аналогій базується на зборі та порівнянні метрик між минулими реалізованими та поточним проектом. Шляхом встановлення схожості між метриками роблять висновок про прогнозну схожість решти характеристик проектів. При цьому враховують тип програмного засобу, призначення модуля, складність та інші параметри і метрики. Аналіз при цьому може здійснюватись на всіх можливих рівнях – від проекту в цілому до конкретного класу [12]. Основним недоліком методу історичних аналогій є необхідність вибору проекту-аналогу. За невдалого вибору розбіжність між прогнозом та реальним станом будуть суттєвими за як завгодно доведеної аналогії між проектами.

Конструктивна модель якості використовує експертно-детерміновані підмоделі введення та усунення дефектів для побудови якісної моделі [12]. В рамках підходу спочатку оцінюється кількість нетривіальних вимог. Використовуються оцінки розміру програмного засобу, ряд атрибутів процесу автоматизації та проекту в цілому. Результатом моделі є оцінка (прогноз) кількості невиявлених дефектів на одиницю розміру [12, 13].

Класифікаційні методи дозволяють виявити програмні модулі з вищою ймовірністю помилок. Зазвичай вони використовуються на більш низьких рівнях деталізації, в тому числі і на рівні класів. Регресійні моделі прогнозування використовуються для вирішення задач, що вимагають дослідження відношення між двома і більше змінними, причому шуканим є кількість дефектів у програмному засобі. Параметри регресії оцінюються за метриками для уже наявних частин проекту (або попередніх ітерацій проекту). Одержана регресія використовується для прогнозування дефектів.

Для задач прогнозування дефектів найчастіше використовується логістична і лінійна регресія [13]. В основі такої моделі лежить допущення про те, що існує дискретний зовнішній фактор, який має визначальний вплив на досліджуваний процес, і цей вплив можна описати лінійною формулою

$$Z(t) = \alpha_0 + \alpha_1 X(t) + \varepsilon_t, \quad (1)$$

де:  $\alpha_0$  і  $\alpha_1$  – коефіцієнти регресії;  $\varepsilon_t$  – помилка моделі [14]. Причому в (1) для одержання прогнозного значення в певний момент часу треба знати значення цього фактору в цей же момент часу, що важко забезпечити на практиці.

Наступною групою методів, які використовуються для класифікації і прогнозування дефектів, є динамічні методи, які базуються на методах машинного навчання [2, 5, 13]. Першим кроком для побудови моделі прогнозування є створення екземплярів з архівів програмного забезпечення. Це використання системи контролю версій, системи відслідковування помилок, архівів файлів метрик. Кожний екземпляр може стосуватись частини проекту, модуля, класу чи функції в залежності від цілей та задач прогнозування, та від масштабу проекту. Кожний досліджуваний екземпляр має метрики і мітки, які вказують схильність до дефектів. Потім відбувається нормалізація даних за певною схемою. Після одержання нормалізованого набору даних відбувається навчання моделі прогнозування. Модель прогнозування може передбачати наявність помилок в новій ітерації програмного блоку, що розглядається. Прогнозування того, чи містить даний блок дефекти, є задачею класифікації. Прогнозування кількості помилок при цьому здійснюється за регресійною моделлю.

З допомогою моделі нечіткої логіки можна формалізувати величини та причинно-наслідкові зв'язки між прогнозованими параметрами та

впливаючими на них параметрами і сформулювати прогноз в умовах невизначеності параметрів прогнозування [13].

Дана методологія передбачає наступні кроки:

- визначення входів і виходів створюваної системи;
- визначення функцій приналежності для кожної метрики;
- розробка нечітких правил і одержання експертної думки;
- агрегування всіх окремих нечітких множин для різних правил;
- пошук чіткого вхідного значення шляхом дефазифікації.

Перевагою такого підходу є можливість використання як кількісних, так і якісних характеристик моделі. Недоліком є значна дуже швидке зростання складності при зростанні кількості входів.

#### **Висвітлення невіршених раніше частин загальної проблеми**

В результаті аналізу виявлено наступні недоліки: описані підходи орієнтуються на вузькі аспекти оцінки якості, наприклад, лише на прогноз кількості помилок; методи прогнозування стосуються конкретних класів задач і мов програмування; методи прогнозування складні та дорогі в впровадженні і використанні, що не гарантує якості прогнозу. Існуючі підходи до прогнозування якості програмних засобів не беруть до уваги якість вимог, що, на думку авторів, суттєво погіршує точність прогнозування.

#### **Формулювання цілей статті.**

На процес розробки програмного забезпечення впливає цілий ряд факторів, які роблять можливим прогнозування характеристик якості програмних засобів або навпаки, ускладнюють його і, відповідно, знижують якість проекту. Серед позитивних можна назвати наступні:

- наявність досвіду роботи команди розробників над схожими проектами;

поняття ‘схожих’ проєктів може бути розглянуте в рамках певної класифікації;

- високої якості технічного завдання на розробку програмних засобів; оцінка якості технічного завдання та їх вплив на прогнозовані характеристики якості програмних засобів можуть бути розглянуті;
- стабільність роботи команди; стійкість її “ядра”. Питання плинності кадрів на проєкті в рамках методики прогнозування характеристики якості може бути враховане;

- високий рівень покриття тестами проєкту в рамках інших метрик буде важливим фактором, який сприятиме якості моделі прогнозування;

- низькі ризики розробки програмного забезпечення незалежно від впровадженої моделі управління ризиками – це позитивний фактор для прогнозування характеристик якості;

- стабільність характеристик якості програмного засобу.

До факторів, вплив яких призведе до погіршення якості прогнозу слід віднести:

- велика складність проєкту, незалежно від моделі оцінки складності;
- зміни та велика кількість уточнень технічного завдання від замовників проєкту;
- значна плинність кадрів на проєкті;
- перше використання командою/ підрозділом/ фірмою мови/ бібліотеки/ фреймворку/тощо;

- перехід до нової методології (наприклад, від СКРАМУ на Керовану тестами розробку);

- низький відсоток покриття тестами як незалежний негативний фактор, що призводить до поганої якості прогнозування програмних засобів.

В цілому можна виділити наступні підходи до прогнозування якості:

- розробка класифікацій проєктів та програмних засобів, з метою прогнозування характеристик якості шляхом переносу “очікувань”, що може здійснюватись:

1) методами кластерного аналізу,

2) найрізноманітнішими методами машинного навчання,

3) розробкою штучних класифікацій на основі аналізу стандартів якості програмного забезпечення;

- прогнозування якості на основі пошуку взаємозв'язків між факторами:

1) аналіз впливу технічного завдання на прогноз якості;

2) кореляційні зв'язки між різними метриками та прогнозними характеристиками на ітераціях розвитку проекту;

- метод експертних оцінок: дану методологію доцільно використовувати у разі, коли наступні фактори ускладнюють можливість прийняти оптимальне рішення:

1) неможливо застосувати жодну модель прогнозування на основі відомої інформації;

2) наявність визначальних факторів, які не піддаються кількісній оцінці;

3) наявність декількох варіантів подальшого розвитку проекту та необхідність вибору одного з них;

4) неповторюваність та неможливість експериментальної перевірки прогнозованого перебігу подій і результатів процесів вирішення проблеми.

Проведений аналіз свідчить, що в ідеальних умовах, визначальним фактом, який впливає на адекватність прогнозу якості програмного засобу, є якість вимог на його розробку [9, 15]. Основна ціль статті – метод прогнозування якості програмних засобів на основі аналізу якості вимог.

#### Висвітлення основного матеріалу дослідження

Як свідчать численні джерела (наприклад, [1, 9, 13]), до 85% причин дефектів полягають у погано сформульованих вимогах, і до 50% дефектів при розробці програмних засобів знаходяться у вимогах. Це є підставою для врахування при прогнозуванні якості програмних засобів характеристик вимог. Першим етапом прогнозування в рамках

даної роботи пропонується історичний підхід (підхід історичних аналогій) – встановлення початкових прогнозних значень методом аналогій.

На основі [16] для розгляду задачі прогнозування якості програмних засобів розглянуто наступні типи вимог:

– функціональні вимоги:

1) бізнес-вимоги,

2) вимоги користувача,

3) функціональні вимоги;

- нефункціональні вимоги:

1) якісні характеристики (quality attributes):

a) зручність у використанні (useability),

b) ефективність (effisency),

c) переносимість (portability),

d) цілісність (integrity);

2) продуктивність (performance),

3) документованість (documentation),

4) вимоги до середовища (environment).

3 метою оцінки вимог розглядаються основні характеристики вимог, а саме [16]:

1) – необхідність (necessary) оцінка того, наскільки важлива дана вимога для виконання основної функції розроблюваного програмного застосунку чи системи в цілому;

2) – повнота (complate) детальність опису вимоги;

3) – узгодженість (consistant) кожна вимога не має суперечити і повинна дублювати інші;

4) – однозначність (unambiquous) безваріантне розуміння вимоги всіма учасниками процесів;

5) – тестованість (veryfible, testible) можливість перевірки, та підтвердження реалізації вимоги;

б) – відслідковуваність (traceble). В рамках даної роботи характеристика відслідковуваності звужується до розуміння узгодженості вимоги в ієрархії “бізнес-вимога – користувацька вимога – функціональна вимога” (хоча насправді це ширша характеристика [16]).

Відповідно до ДСТУ [17] користувацька якість програмного забезпечення базується на внутрішній та зовнішній якості (рис. 3).

Для оцінки впливу характеристики вимог на прогнозне значення характеристики якості введемо наступну шкали (таблиця 1). В таблиці під ‘хорошими вимогами’ розуміється вимоги, які відповідають здоровому глузду та

стандартам роботи згідно [17]. Будь-яка обґрунтована підозра на відсутню вимогу погіршить прогноз вказаних характеристик на умовну одиницю. Приклад характеристики впливів показано на рисунку 5.



Рисунок 3 – Характеристики якості програмних засобів [17]



Рисунок 4 – Модель якості програмного забезпечення [17]

Результуючі показники на основі наведених взаємозалежностей коректуються за формулою [18]:

$$q = q_i + k_B \sum kpb + k_U \sum kri + k_F \sum kpf + k_{NF} \sum kpnf \quad (2)$$

де  $q$  – прогнозована метрика характеристики якості,

$q_i$  – прогноз на основі історичного аналізу,

$k_B$  – ваговий коефіцієнт впливу на прогноз якості бізнес вимог,

$kpb$  – корекції прогнозу за бізнес-вимогами,

$k_U$  – ваговий коефіцієнт впливу на прогноз якості вимог користувача,

$kri$  – корекції прогнозу за вимогами користувача,

$k_F$  – ваговий коефіцієнт впливу на прогноз якості вимог до функцій,

$k_{pf}$  – корекції прогнозу за вимогами до функцій,

$k_{NF}$  – ваговий коефіцієнт впливу на прогноз якості нефункціональних вимог,

$k_{pnf}$  – корекції прогнозу за нефункціональними вимогами.

В загальному випадку вагові коефіцієнти впливу обираються в залежності від прийнятої шкали та

масштабу проекту (кількості вимог). При цьому, якщо одержані значення прогнозів “впираються” в верх шкали, то робиться висновок, що ніщо не перешкоджає досягненню високої якості програмного продукту; якщо в малі – треба працювати з вимогами.

Таблиця 1 – Кількісна оцінка (шкалювання) впливу характеристик вимог на прогнозне значення характеристик якості програмних засобів

Характеристика вимоги	Оцінка	Характеристика оцінки
Необхідність	-1	Незрозуміло, наскільки важливою є дана вимога для цілі розроблюваного програмного засобу
	0	Опис відповідає вимогам
	+1	Надано вичерпне пояснення
Повнота опису	-1	Недостаній опис вимоги
	0	Опис формально відповідає вимогам до “хороших вимог”
	+1	Опис гранично ‘прозорий’ для можливого застосування
Узгодженість	-1	Необхідним є аналіз вимог на несуперечливість
	0	Опис відповідає вимогам до вимог на розробку
	+1	Надано вичерпне пояснення
Однозначність	-1	Необхідно складати глосарій, щоб виявити/уточнити можливі різбіжності
	0	Опис відповідає вимогам до вимог на розробку
	+1	Надано вичерпне пояснення
Можливість перевірки	-1	Інформація про можливості перевірки недостатньо
	0	Опис формально відповідає вимогам до “хороших вимог”
	+1	Наявна вся можлива інформація для перевірок до тестових прикладів включно, де це можливо
Відслідковуваність	-1	Інформації для відслідковуваності недостатньо
	0	Опис формально відповідає вимогам до “хороших вимог”
	+1	Роль конкретної вимоги в ієрархії вимог та цілях проекту зрозуміла



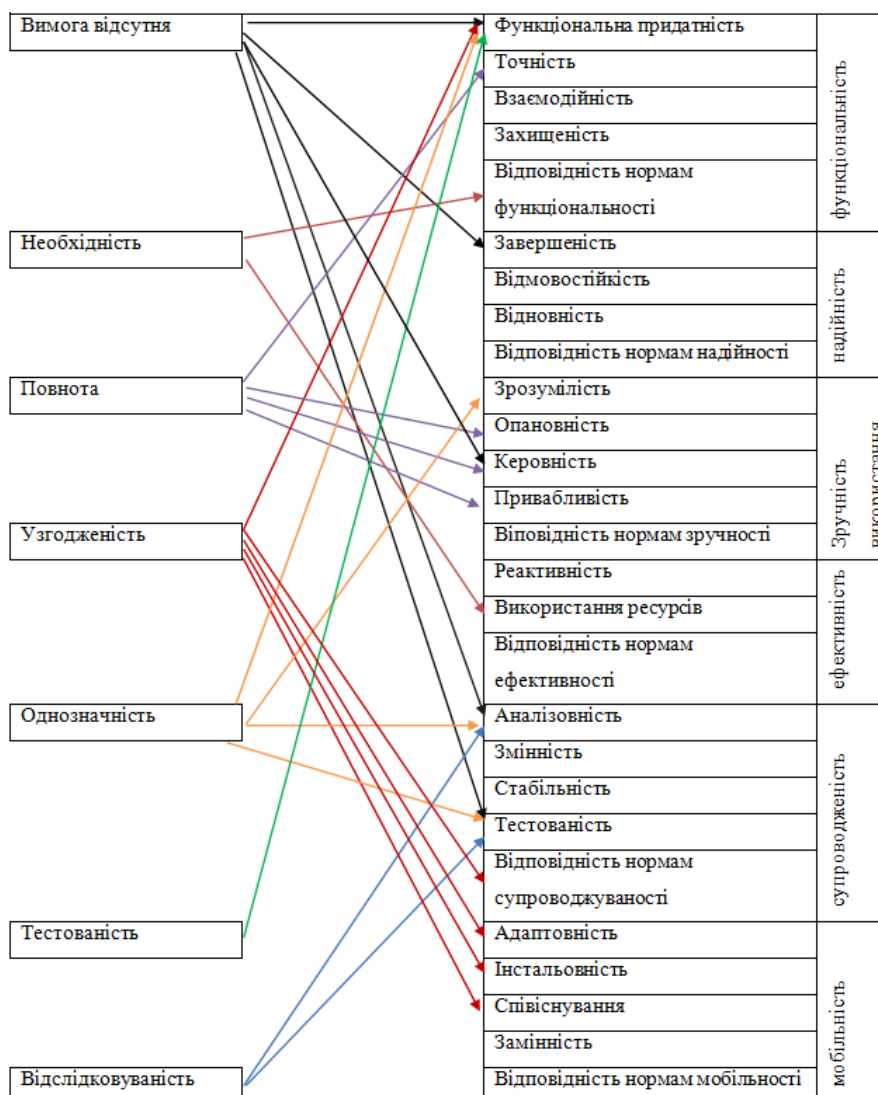


Рисунок 5 – Вплив на прогноз характеристик вимог користувача (приклад, розроблений для типового редактора коду)

### Висновки

Проаналізовано існуючі підходи та методи прогнозування характеристик якості програмних засобів в рамках метричного підходу. Здійснено аналіз впливів, які впливають на якість розроблених програмних засобів. Встановлено, що за інших рівних вимог визначальний вплив на досяжні характеристики якості має якість вимог на розробку програмного засобу. Пропонований метод прогнозування характеристики якості вимог ґрунтується і відштовхується від етапу тестування

(верифікації) вимог. Запропоновано адаптовну шкалу для оцінки характеристик якості вимог для прогнозування характеристик якості програмних засобів. Метод історичних аналогій дає точку відліку характеристикам якості. Якість вимог дозволяє уточнити (власне, спрогнозувати) в рамках запропонованої прогностичної моделі прогнозовані значення характеристики якості конкретного майбутнього проекту. Одержані значення можуть бути використані як для обґрунтування досяжних

характеристик якості, так і для покращення зворотніх зв'язків з замовниками для покращення вимог до проекту.

### Список використаних джерел

1. Lanubile F. and Visaggio G. Evaluating predictive quality models derived from software measures: lessons learned. *Journal of Systems and Software*. 1997. Vol. 38. Pp. 225–234.
2. Hong E., Wu C. Criticality models using SDL metrics set. *4 th Asia-Pacific Software Engineering and International Computer Science Conference*. 1997. P. 23–30.
3. Evett M., Khoshgoftar T., Chien P. and Allen E. GP-based software quality prediction, *Third Annual Genetic Programming Conference*. 1998. P. 60–65.
4. Zhang D., Tsai J.J.P. Machine learning and software engineering. *Software Quality Journal*. 2003. Vol. 11. Issue 2, P. 87–119.
5. Ganesan K., Khoshgoftar T., Allen E. Casedbased software quality prediction, *International Journal of Software Engineering and Knowledge Engineering*. 2000. Vol. 10, N 2. P. 139–152.
6. Khoshgoftar T., Seliya N. Analogy-Based Practical Classification Rules for Software Quality Estimation. *Empirical Software Engineering*. 2003. Vol. 8. N 4. P. 325–350.
7. Khoshgoftar T., Allen E.B., Deng J. Using regression trees to classify fault-prone software modules, *IEEE Transactions on Reliability*, 2002, Vol. 51, N 4, P. 455–462.
8. Netkachova K. I. Safety Case Methodology: Architecting principles. *Радіоелектроніка і комп'ютерні системи*. Харків: НАУ"ХАІ". 2010. №7. С.109-112.
9. Говорущенко Т.О. Теоретичні та прикладні засади інформаційної технології оцінювання достатності інформації щодо якості у специфікаціях вимог до програмного забезпечення – Кваліфікаційна наукова праця на правах рукопису. Дисертація на здобуття наукового ступеня доктора технічних наук за спеціальністю 05.13.06 «Інформаційні технології» (12 – Інформаційні технології) – Хмельницький національний університет, Українська академія друкарства, Львів, 2018, 310 с.
10. Білецький Т. П., Федасюк Д. В. Прогнозування дефектів у програмному забезпеченні алгоритмами глибинного навчання CNNта RNN. *Науковий вісник НЛТУ України*. 2021. Т. 31. №2. С. 114 – 120.
11. Subnis P., Kadam A. Software reliability growth model with bug ccle and duplicate detection techniques. *Bharati Vidyapeech Deemed univ. CollegeofEng. PuneIndia*. 2013. P 345-349.
12. Яковина В. С. Симець І. І. Прогнозування дефектів програмного забезпечення ансамблем нейронних мереж. *Науковий вісник НЛТУ України*. 2021. Т. 31, №6. С.104 – 111.
13. Науково-технічні дослідження у галузі інформаційних технологій: колективна монографія / заг. ред. М. В. Кузь. – Академія технічних наук України. – Івано-Франківськ: Видавець Кушнір Г.М. 2022. Т1. 152 с.
14. Єріна А. М. Статистичне моделювання та прогнозування. Київ: КНТЕУ, 2001. 114с.
15. Незамай Б. С. Методика прогнозування якості програмних засобів, *Перспективи розвитку науки, освіти та технологій в контексті євроінтеграції: збірник тез доповідей міжнародної науково-практичної конференції (Полтава, 18 серпня 2022 р.)*. Полтава: ЦФЕНД. 2022. С. 65-66.
16. George Koelsch Herndon *Requirements Writing for System Engineering*. Virginia, USA. 2016. 409с.
17. ДСТУ ISO/IEC 9126-1:2013. Програмна інженерія: якість продукту Частина 1. Модель якості (ISO/IEC TR 9126-1:2001, IDT). (Національний стандарт України)

18. Незамай Б. С. Прогнозування якості програмного забезпечення. 2022 *International Conference on Innovative Solutions in Software Engineering (ICISSE) присвячена 50-річчю від дня народження професора Павла Федорука*. Івано-Франківськ, 30 листопада 2022 р. Івано-Франківськ: ПНУ. 2022. С. 86-89.

### References

1. Lanubile F. and Visaggio G. Evaluating predictive quality models derived from software measures: lessons learned. *Journal of Systems and Software*. 1997. Vol. 38. P. 225–234.
2. Hong E., Wu C. Criticality models using SDL metrics set, *4 th Asia-Pacific Software Engineering and International Computer Science Conference*. 1997. P. 23–30.
3. Evett M., Khoshgoftar T., Chien P. and Allen E. GP-based software quality prediction, *Third Annual Genetic Programming Conference*, 1998, P. 60–65.
4. Zhang D., Tsai J.J.P. Machine learning and software engineering, *Software Quality Journal*. 2003. Vol. 11, Issue 2. P. 87–119.
5. Ganesan K., Khoshgoftar T., Allen E. Casedbased software quality prediction, *International Journal of Software Engineering and Knowledge Engineering*. 2000. Vol. 10. No 2. P. 139–152.
6. Khoshgoftar T., Seliya N. Analogy-Based Practical Classification Rules for Software Quality Estimation, *Empirical Software Engineering*. 2003. Vol. 8, No 4. P. 325–350.
7. Khoshgoftar T., Allen E.B., Deng J. Using regression trees to classify fault-prone software modules, *IEEE Transactions on Reliability*. 2002. Vol. 51, No 4. P. 455–462.
8. Netkachova K. I. Safety Case Methodology: Architecting principles. *Радіоелектроніка і комп'ютерні системи*. Харків: НАУ"ХАІ". 2010. №7. С.109-112.
9. Hovorushchenko T.O. Teoretychni ta prykladni zasady informatsiynoyi tekhnolohiyi otsynuyannya dostatnosti informatsiyi shchodo yakosti u spetsyfikatsiyakh vymoh do prohramnoho zabezpechennya – Kvalifikatsiyna naukova pratsya na pravakh rukopysu. Dysertatsiya na zdobuttya naukovooho stupenya doktora tekhnichnykh nauk za spetsial'nistyuu 05.13.06 «Informatsiyni tekhnolohiyi» (12 – Informatsiyni tekhnolohiyi) – Khmel'nyts'kyu natsional'nyy universytet, Ukrayins'ka akademiya drukarstva, L'viv. 2018. 310 p. [in Ukrainian].
10. Bilets'kyu T. P., Fedasyuk D. V. Prohnozuvannya defektiv u prohramnomu zabezpechenni alhorytmamy hlybnyynoho navchannya CNNta RNN. *Naukovyy visnyk NLTU Ukrayiny*. 2021. T. 31, №2. S. 114 – 120. [in Ukrainian].
11. Subnis P., Kadam A. Software reliability growth model with bug ccle and duplicate detection techniques. *Bharati Vidyapeech Deemed univ. College of Eng. Pune India*. 2013. P 345-349.
12. Yakovyna V. S. Symets' I. I. Prohnozuvannya defektiv prohramnoho zabezpechennya ansamblem neyronnykh merezh. *Naukovyy visnyk NLTU Ukrayiny*. 2021. T. 31, №6. P. 104 – 111. [in Ukrainian].
13. Naukovo-tekhnichni doslidzhennya u haluzi informatsiynykh tekhnolohiy: kolektyvna monohrafiya / zah. red. M. V. Kuz'. Akademiya tekhnichnykh nauk Ukrayiny. Ivano-Frankivs'k: Vydavets' Kushnir H.M. 2022. Vol 1. 152 p. [in Ukrainian].
14. Yerina A. M. Statystychno modelyuvannya ta prohnozuvannya. Kyiv: KNTEU. 2001. 114 p. [in Ukrainian].
15. Nezamay B. S. Metodyka prohnozuvannya yakosti prohramnykh zasobiv, Perspektivy rozvytku nauky, osvity ta tekhnolohiy v konteksti yevrointehratsiyi: zbirnyk tez dopovidey mizhnarodnoyi naukovo-praktychnoyi konferentsiyi (Poltava, 18 08 2022.), Poltava: TSFEND. 2022. P. 65-66. [in Ukrainian].
16. George Koelsch Herndon Requirements Writing for System Engineering. Virginia, USA. 2016. 409 p.

17. DSTU ISO/IEC 9126-1:2013. Prohramna inzheneriya: YAKIST' PRODUKTU Chastyna 1. Model' yakosti (ISO/IEC TR 9126-1:2001, IDT). (Natsional'nyy standart Ukrayiny) [in Ukrainian].

18. Nezamay B. S. Prohnozuvannya yakosti prohramnoho zabezpechennya. 2022 International Conference on Innovative Solutions in Software Engineering (ICISSE) prysvyachena 50-richchyu vid dnya narodzhennya profesora Pavla Fedoruka. Ivano-Frankiv'k, 30 11 2022. Ivano-Frankiv's'k: PNU. 2022. p. 86-89. [in Ukrainian].