

УДК 683.1+ 004.5

DOI 10.31471/1993-9981-2020-2(45)-40-57

МЕТОДОЛОГІЧНІ ПІДХОДИ ЩОДО ОЦІНКИ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОБ'ЄКТІВ НАФТОГАЗОВОГО КОМПЛЕКСУ

В.М. Юрчишин

*Івано-Франківський національний технічний університет нафти і газу,
вул. Карпатська 15, м. Івано-Франківськ, 76019, e-mail: pubic@iung.edu.ua*

Метою даного дослідження є визначення методологічних підходів щодо оцінки якості програмного забезпечення при впровадженні на нафтогазовидобувних підприємствах сучасних інформаційних технологій. Відповідно до проведеного автором комплексного аналізу методичних підходів з оцінки програмного забезпечення запропоновано методичні положення оцінки якості програмного продукту.

Розглянуті та проаналізовані найбільш поширені програмні засоби, методичні підходи для оцінки якості програмних продуктів для інформаційних систем.

Сформульовано та розглянуто методологічні підходи щодо оцінки якості програмного забезпечення для об'єктів нафтогазового комплексу. Розглянуто сучасний стан проблеми та попередні дослідження у цій сфері, а також основні, хоча і далеко не всі, методики оцінки якості програмного забезпечення.

Описано і проаналізовано деякі з проблем, що пов'язані зі створенням і застосуванням методик оцінки якості програмних кодів загалом і зокрема баз знань в нафтогазовій предметній області.

В статті представлені аргументи, що забезпечують необхідність оцінки затрат на якість програмного забезпечення.

Ключові слова: *інформаційні технології; програмний продукт; якість програмного коду; показники якості; стандарти якості; нафтогазова предметна область.*

Аннотація. *Целью данного исследования есть определение методологических подходов к оценке качества программного обеспечения при внедрении на нефтегазодобывающих предприятиях современных информационных технологий. Соответственно проведенному автором комплексному анализу методических подходов к оценке качества программного обеспечения предложено методические положения оценки качества программного продукта.*

Рассмотрены и проанализированы наиболее распространенные программные средства, методические подходы для оценки программных продуктов для информационных систем.

Сформулированы и рассмотрены методологические подходы к оценке качества программного обеспечения для объектов нефтегазового комплекса.

Рассмотрено современное состояние проблемы и предыдущие исследования в этой среде, а также основные, хотя и далеко не все, методики оценки качества программы.

Описано и проанализировано некоторые проблемы, которые связаны с созданием и использованием методик оценки качества программных кодов в целом и отдельно баз знаний в нефтегазовой предметной области.

В статье представлены аргументы, которые обеспечивают необходимость оценки затрат на качество программного обеспечения.

Ключевые слова: *информационные технологии; программный продукт; качество программного кода; показатели качества; стандарты качества; нефтегазовая предметная область.*

***Annotation.** This article establishes methodological approaches to software quality assessment in the process of information technology implementation in oil and gas companies. This study offers methodological basis of software quality assessment drawing on complex analyses of methodological approaches to software quality assessment conducted by the author/*

The article investigates and analyses the most commonly used approaches to software quality assessment for information systems.

The study examines methodological approaches to software quality assessment for oil and gas complex objects.

The article describes current state of affairs and the results of previous research in this field including the main methods of software quality assessment.

This research describes and analyses some of the problems connected to creation and implementation of software code quality assessment methods in general and oil and gas industry knowledge base in particular.

The article presents arguments for necessity for assessment of software quality expenses.

Key words: *information technology, software product, software code quality; quality indicators; quality standard; oil and gas industry.*

Вступ

Ефективне функціонування інформаційних технологій в нафтогазовій галузі визначається якістю компонент її складових: технічної, математичної, програмної, інформаційної, правової та інших. Одним з найбільш важливих складових є програмне забезпечення, управління якістю якого значною мірою визначається якість інформаційної системи. В даний час широко використовуються досить ефективні методи, моделі та способи оцінки якості програмного забезпечення, що включають такі показники якості програмних продуктів, як функціональність, надійність, практичність, ефективність, мобільність, інтерактивність тощо [1].

Створення високоякісного програмного забезпечення є однією з важливих завдань розвитку нафтогазової галузі. Від того, наскільки вдало зроблено програмне забезпечення системи, залежить в кінцевому результаті ефективне прийняття рішень при управлінні роботою нафтогазових об'єктів. Якість програмного забезпечення (ПЗ) є основною його характеристикою в різних сферах використання інформаційних технологій, яка вказує на ступінь його відповідності встановленим вимогам міжнародним стандартам якості [2]. Зазвичай, такі вимоги трактують по-різному, що породжує декілька незалежних визначень цього терміну. Здебільшого, під якістю ПЗ розуміють

набір властивостей програмного продукту, що характеризують його здатність задовольнити встановлені або передбачувані потреби замовника, які він висловив у вигляді користувацьких вимог до нього на початкових етапах його розробки.

Стандарт ISO/IEC 9126 [3] регламентує зовнішні й внутрішні характеристики якості ПЗ. Якщо зовнішні характеристики відображають вимоги до функціонування ПЗ, то внутрішні використовують для підготовки плану досягнення необхідних значень зовнішніх характеристик його якості.

Як зовнішні, так і внутрішні характеристики якості відображають властивості самого ПЗ, а також погляди замовника і розробника на нього. Однак, безпосереднього користувача ПЗ в основному цікавить експлуатаційна його якість, тобто сукупний ефект від досягнення необхідних характеристик програми, значення яких вимірюється швидкістю та достовірністю отриманого результату, а не його властивістю.

Під оцінюванням якості ПЗ розуміють дії, що визначають, як саме ПЗ відповідає своєму призначенню. Якість ПЗ оцінюють з використанням моделі якості (ISO/IEC 9126-1, 2001). Таке оцінювання набуває особливого значення із розвитком і вдосконаленням технологій визначення якості ПЗ. Усе це призвело до потреби систематизації методів

оцінки якості програмного забезпечення, що використовується в нафтогазовій галузі.

Аналіз сучасних закордонних і вітчизняних досліджень та публікацій щодо розгляду питання оцінки якості програмного забезпечення дозволив скласти загальне уявлення про наявність існуючої методологічної бази сучасної науки в галузі інформаційних технологій [4-10]. Проведене дослідження показало, що оцінка якості програмних продуктів, що використовуються в нафтогазовій галузі на період сьогодні проводиться переважно з використанням розширених методів оцінювання техніко-економічних показників на різних стадіях впровадження нововведень, кількісних методів оцінювання якості програмного забезпечення. Безперечним є той факт, що проведення оцінки якості програмних продуктів з метою визначення рівня ефективності від використання їх на різних етапах видобування вуглеводнів потребує комплексного аналізу кількісних показників, які в найбільшій мірі будуть відповідати специфіці діяльності нафтогазовидобувного підприємства. Також в ході проведеного аналізу різних методичних підходів з оцінки якості програмного забезпечення було визначено наявність єдиної бази інструментарію оцінювання інноваційних дій на підприємстві, яка представляє собою поєднання роботи різних програм.

Таким чином, було виявлено наявність значного теоретичного підґрунтя для формування методологічної бази по оцінці якості програмного забезпечення, яке, на мою думку, потребує подальшої систематизації. На основі систематизації існуючих методичних підходів з оцінки якості програмного забезпечення можлива розробка єдиних методичних рекомендацій з цього питання, які повинні ґрунтуватися на принципах урахування загальних проблем впровадження інформаційних технологій на нафтогазовидобувних підприємствах.

Отже, необхідно визначити ефективні шляхи удосконалення методики оцінки якості програмного забезпечення на підприємствах нафтогазової галузі, які дозволять вирішити проблему невизначеності та недооцінки результатів в практиці господарської діяльності сучасних вітчизняних підприємств.

Згідно стандарту ISO/IEC 9126 для опису оцінки якості програмних продуктів важливими і досить незалежними є такі характеристики: 1) зручність експлуатації або модифікація; 2) життєздатність, чіткість, ефективність; 3) вартість; 4) мобільність і ступінь обліку людських факторів.

Кожна характеристика досягає свого найвищого значення тоді, коли вона є критерієм успіху групи, що розробляла програму. Усе більше фахівців починають визнавати важливість якості програмних засобів, і намагаються впливати на якісні характеристики програмного забезпечення.

Зроблено певні кроки у напрямку визнання важливості критерію надійності при проектуванні систем програмного забезпечення. З'явилися матеріали, присвячені проблемам створення метрики програмного забезпечення й аналізу його якості. Так, наприклад, книга Майерса стосується питань надійності програмного забезпечення [11], у якій увага зосереджується на методах конструювання високонадійних програм і способах оцінки надійності програмного забезпечення за результатами спостереження за процесами налагодження програм і усунення помилок.

Проведений аналіз показав відсутність єдиного підходу з окресленої проблеми як в наукових колах, так і в практиці застосування єдиних заходів з оцінювання якості програмного забезпечення. Тому, при такій оцінці, незважаючи на різні напрямки проведення такого оцінювання, ключовими аспектами процедури оцінювання виступають – кінцевий результат обраного напрямку оцінювання, або, іншими словами, результативність, а також ефективність застосування ПЗ.

Якості програмного забезпечення присвячено багато публікацій різного характеру. Їх автори в основному посилаються на один і той же стандарт серії ISO 9000, а також на національну нормативну базу. Основним способом забезпечення і підтвердження якості програмних продуктів пропонується вважати використання сертифікації на відповідність цим стандартам, але це явно недостатньо для реального забезпечення якості. Практичний наш досвід свідчить про відносну дійсність сертифікації і постійно ставить програмістів перед необхідністю реально вирішувати задачі

забезпечення, підтримання і підвищення якості своїх програмних продуктів.

Огляд літературних джерел та їх аналіз стосовно якості програмного забезпечення свідчить, що основними складовими є [8,9]: 1) якість інфраструктури (infrastructure quality), тобто апаратного відповідного програмного забезпечення (якість операційних систем, комп'ютерних мереж тощо); 2) якість даних (data quality), які використовуються інформаційною системою на вході; 3) якість адміністративного управління (administrative quality), тобто якість менеджменту, включаючи планування і календарний контроль; 4) якість сервісу (service quality), тобто якість навчання системною підтримкою.

Порівняльний аналіз стандартів показує, що в різних нормативних документах номенклатура показників якості програмних засобів помітно відрізняється один від одного. Стандарти практично не містять рекомендацій по вибору, застосуванню і впорядкуванню необхідної сукупності показників якості. Крім того, для більшості показників відсутні методики їх оцінки і порівняння з встановленими вимогами або критеріями. Практичне застосування нормативних документів ускладнюється тим, що ні один із них не містить повного переліку факторів, що впливають на якість програмних засобів і методик їх оцінки.

На жаль, в даний час добиватися задоволення усіх цих якостей при розробці програмного забезпечення досить складно. З іншого боку швидке збільшення складності і розмірів сучасних програмних комплексів при одночасному рості відповідальності функцій, що виконуються, різко підвищилися вимоги зі сторони замовників і користувачів до їх якості і ефективності застосування. В зв'язку з цим стратегічною задачею в життєвому циклі сучасного програмного забезпечення стало забезпечення необхідної якості програмних продуктів.

Отже, відповідно до визначення обмеженості наукових поглядів в методичних підходах по оцінці якості програмних продуктів при впровадженні інформаційних технологій на нафтогазовидобувних підприємствах вимагає систематизація такої оцінки.

Метою даного дослідження є визначення методологічних підходів щодо оцінювання якості програмного забезпечення при впровадженні на нафтогазовидобувних підприємствах сучасних інформаційних технологій

Відповідно до проведеного автором комплексного аналізу методичних підходів з оцінювання програмного забезпечення необхідно розробити методичні положення по оцінці якості програмного продукту. Таке поєднання видів оцінки може визначитися на основі інтегрального показника оцінювання рівня якості програмного продукту.

Отже, можна стверджувати, що метою оцінки якості ПЗ є комплексний аналіз задоволення потреб та її впливу на найважливіші показники діяльності нафтогазовидобувного підприємства, визначення доцільності й оптимальних варіантів реалізації нововведень, оперативне коригування управління нафтогазовидобувним об'єктом.

Як свідчить наш досвід, найбільшу трудність в процесі управління якістю будь-якою продукцією, в тому числі програмною, викликає встановлення факторів, які викликають встановлення, тих чи інших характеристик якості. Ще більше складною і в той же необхідною процедурою є їх оцінка. Виміряти їх – означає отримати об'єктивну основу для вироблення оперативних і економічно ефективних керуючих дій при впровадженні сучасних інформаційних технологій на нафтогазовидобувних підприємствах. Тому актуальним є вибір і впровадження методів, що забезпечують виявлення конкретних залежностей між факторами і характеристиками якості, методів, які дозволяють приймати обґрунтовані рішення і здійснювати конкретні заходи.

Така увага до розробки методів оцінки якості програмного забезпечення при впровадженні сучасних інформаційних технологій на нафтогазовидобувних підприємствах, на мою думку, є цілком зваженою та обґрунтованою. Визначаючи сутність впровадження нових технологій на підприємстві без попередньої оцінки якості програмних продуктів може спричинити негативні наслідки. Підприємству може не вистачити коштів для завершення початого проекту, що негативно

відобразиться на забезпеченні ефективної поточної роботи нафтогазовидобувного підприємства або й взагалі призведе до зупинки певного об'єкту.

Застосування сучасних методів оцінки якості програмного забезпечення можуть служити основою для систем забезпечення якості програмних засобів, але використання їх вимагає певного корегування, а також адаптації деяких положень стандартів стосовно до принципових особливостей технологій створення і характеристикам програмної продукції. При цьому є значна кількість методик оцінки якості програмних продуктів, що дають інколи суб'єктивні експертні заключення, що не відповідають існуючим технологіям проектування, виробництва і оцінки якості інформаційних систем, а також сучасним міжнародним стандартам, які необхідно освоювати і застосовувати для забезпечення конкурентоздатності продукції ІТ-галузі на світовому ринку. Важливість поставленої проблеми визначається тим, що найбільш зацікавленими у якості програмного комплексу є кінцевий користувач, що використовує програмний продукт.

Отже, в сучасних умовах для нафтогазовидобувних підприємств гостро стоїть проблема вибору найбільш ефективного впровадження інформаційних технологій з врахуванням оцінки якості програмного забезпечення для прийняття ефективних рішень при управлінні нафтогазовими об'єктами, обумовлюють актуальність даної теми дослідження.

В даний час розробка програмного забезпечення для інформаційних систем перетворилася в одну з самих дорогих індустрій і будь-які вузькі місця в технологічному процесі його створення можуть привести до небажаних наслідків. Збільшення термінів розробки програмного забезпечення може привести до подорожчання кінцевого продукту, а не виявлення в ході оцінки якості програмного продукту помилки приводить як мінімум до зниження продуктивності. Але й примітивні помилки, незрозумілі повідомлення і недбалий інтерфейс дратують користувача, які в результаті вибирають більш якісний програмний продукт конкурента, ІТ-компанія ризикує втратити не тільки клієнта, але й свою

долю на ринку. Нерідко це виникає в зв'язку з тим, що контракти, попередні плани, технічні умови і завдання на створення складного програмного забезпечення підготовлюються і оцінюються некваліфіковано, на основі неформалізованих уявлень замовників, розробників і користувачів про необхідні функції та характеристики якості програмного забезпечення, що часто не відповідають вимогам міжнародних стандартів серії ISO. Тому багато програмних продуктів не здатне повністю виконувати необхідні функціональні задачі з гарантованою якістю, і їх приходиться довго і інколи безуспішно допрацьовувати для досягнення необхідної споживчої якості та надійності функціонування, витрачаючи додатково значні засоби і час.

Розглянемо методологію оцінки якості програмного продукту. Програми набувають високої якості не стільки в результаті комплексного тестування кінцевого продукту, скільки в процесі його розробки. Якщо методологія створення програмного продукту така, що помилки виявляють на регулярній основі і на всіх стадіях виконання проекту, то на виході ПЗ не буде мати помилок.

Корпорація ІВМ запропонувала методологію створення складних програмних комплексів, що одержала назву Cleanroom Software Engineering [12]. Вона орієнтована на професіоналів, що бажають удосконалити свої методи розробки програмних продуктів. Cleanroom – це сукупність адміністративно-технологічних процесів, що дозволяють колективам розробників ПЗ планувати, вимірювати, проектувати, кодувати, тестувати і сертифікувати програмні продукти. Методологія Cleanroom побудована на трьох концепціях: 1) модульному принципі специфікації й проектування; 2) математичному доказі правильності застосовуваних алгоритмів; 3) використанні статистики за результатами тестування як основи для оцінки надійності програм.

Метод покрокової деталізації з адміністративної точки зору методології Cleanroom полягає у покроковій деталізації проекту, коли кінцева функціональність системи досягається ітераційно. На кожному етапі реалізується визначений рівень функціональних можливостей, що тестується і сертифікується

автономно. Такий спосіб розробки має кілька плюсів. З одного боку, видно, як система розвивається, а з іншого – виникають добрі передумови для поліпшення не тільки самого продукту, але і процесів його виробництва – адже на кожному етапі аналізуються джерела виникнення помилок і відбувається їх усунення. На етапі формування архітектури майбутньої системи процедури тестування проводяться більш ретельно, що дозволяє локалізувати помилки на ранніх стадіях. Специфікації Cleanroom дають повний і точний опис функцій системи. Вироблення специфікацій сприяє більш глибокому розумінню вимог, пропонувананих до кінцевого продукту, і його функцій, а самі специфікації є основою для тестування, сертифікації і подальшого розвитку системи. Відповідно до методології Cleanroom системи будуються за модульним принципом.

Відповідність роботи програми закладеним специфікаціям перевіряється в ході верифікації. Інструментом тестування надійності й оцінки якості ПЗ в методології Cleanroom є середовище Cleanroom Certification Assistant, в основі якої покладено ідею використання статистичних результатів тестування для визначення надійності ПЗ математичними методами.

Існуючі на сьогодні методи тестування програмного забезпечення не дозволяють однозначно і повністю виявити усі дефекти та установити коректність функціонування програми, яка аналізується, тому, що всі існуючі методи тестування діють в рамках формального процесу перевірки програмного забезпечення. На сьогодні це практично єдина характеристика якості програмного забезпечення, яку можна встановити автоматичним шляхом.

Для систематизації методів оцінки якості програмного забезпечення розглянемо існуючі різноманітні визначення якості ПЗ.

На сьогоднішній день згідно міжнародних стандартів існують різні визначення якості програмного забезпечення, які подібні одне на одне, наприклад [13]: 1. Якість програмного забезпечення - ступінь, в якій воно має необхідні комбінації властивостей. 2. Якість програмного забезпечення - ступінь, в якій система компонентів задовольняє вимогам, потребам або очікуванням замовника чи користувача. 3.

Якість програмного забезпечення - здатність програмного продукту при заданих умовах задовольняє встановленим або очікуваним потребам. 4. Якість програмного забезпечення - це сукупність характеристик цього забезпечення, яка здатна задовільнити встановлені і очікувані потреби.

Отже, підсумовуючи можна зробити висновок, що якість ПЗ - це характеристика програмного забезпечення, ступінь відповідності його до вимог. При цьому вимоги можуть трактуватись по-різному, що породжує декілька незалежних визначень терміну. Якість ПЗ - набір властивостей продукту, що характеризують його здатність задовольнити встановлені або передбачувані потреби замовника. Поняття якості має різні інтерпретації залежно від конкретної програмної системи і вимог до неї.

Під час оцінки і покращення якості ПЗ необхідно на нафтогазовому підприємстві виконувати такі задачі: 1. Порівнювати процес розробки програмного забезпечення, існуючого в даний час на підприємстві з описаною в стандарті моделлю. Аналіз результатів дає можливість визначити сильні і слабкі сторони процесу. 2. Оцінити можливості покращення даного процесу на основі визначення поточних можливостей. 3. Виконати технічну реалізацію поставлених задач на основі сформульованих цілей удосконалення процесу. Після цього увесь цикл починається з початку.

З нашого досвіду якість програмного забезпечення повинна відповідати наступним вимогам: легко використовувати; хороша продуктивність; немає помилок; не зіпсує дані користувача при збоях; може працювати тривалий час; можна використовувати на різних платформах; легко добавляти нові можливості; задовольняє потреби користувача; добре документоване.

Детальніше зупинимося на зовнішніх і внутрішніх характеристиках які визначають якість ПЗ. До зовнішніх характеристик відносяться властивості, які усвідомлює користувач програми [13]: 1. Коректність - відсутність або наявність дефектів у специфікації, проекті та реалізації системи. 2. Практичність - легкість вивчення і використання системи. 3. Ефективність - ступінь використання системних ресурсів. Ця характеристика

враховує такі фактори, як швидкодія програми і необхідний їй обсяг пам'яті. 4. Надійність - здатність системи виконувати необхідні функції за певних умов; середній інтервал між відмовами. 5. Цілісність - здатність системи запобігати неавторизованому або некоректному доступу до своїх програм та даних. Ідея цілісності передбачає обмеження доступу до системи для неавторизованих користувачів, а також забезпечення правильності доступу до даних, тобто одночасну зміну взаємопов'язаних даних, зберігання лише допустимих значень тощо. 6. Адаптованість - можливість використання системи без її зміни в тих галузях або середовищах, на які вона не була орієнтована безпосередньо. 7. Правильність - ступінь безпомилковості системи, особливо щодо виведення кількісних даних. Правильність характеризує виконання системою її функцій, а не те, чи створена вона коректно. Цим правильність відрізняється від коректності. 8. Живучість - здатність системи продовжувати роботу при введенні неприпустимих даних або в напружених умовах.

Деякі з цих характеристик перекриваються, проте кожна має свої відмінні риси, які в одних випадках виражені сильніше, а в інших слабше.

Зовнішні характеристики - єдина категорія властивостей ПЗ, яка турбує користувачів. Користувачів хвилює легкість роботи з ПЗ, а не легкість його зміни. Їх хвилює коректність ПЗ, а не зручність читання або структурованість коду. А програмістів хвилюють і зовнішні характеристики ПЗ, і внутрішні.

До внутрішніх характеристик програмного забезпечення відносяться: 1. Зручність супроводу - легкість зміни програмної системи з метою реалізації додаткових можливостей, підвищення швидкодії, виправлення дефектів тощо. 2. Гнучкість - можливий масштаб зміни системи з метою використання її в тих галузях або середовищах, на які вона не була безпосередньо орієнтована. 3. Портваність - легкість зміни системи з метою використання у середовищах, на які вона не була орієнтована безпосередньо. 4. Можливість повторного використання - масштабність і легкість використання частин системи в інших системах. 5. Зручність читання

- легкість читання та розуміння вихідного коду системи, особливо на детальному рівні окремих операторів. 6. Тестованість - можливий ступінь виконання блокового і системного тестування програми та перевірки її відповідності вимогам. 7. Зрозумілість - легкість розуміння системи і на рівні загальної організації, і на детальному рівні окремих операторів. Зрозумілість характеризує узгодженість системи на більш загальному рівні, ніж легкість для читання.

Різниця між внутрішніми і зовнішніми характеристиками якості розмита, тому що на деякому рівні перші впливають на другі. Якщо програма є не досить зрозумілою або незручною в супроводі, у ній важко виправляти дефекти, що в свою чергу впливає на такі зовнішні характеристики, як коректність і надійність. ПЗ, що потерпає від нестачі гнучкості, не можна поліпшити у відповідь на запити користувачів, що позначається на його практичності.

Розглянемо класифікацію різних підходів до оцінки якості програмного забезпечення, використовуючи такі два напрямки.

Перший напрямок орієнтований або на проект, або процес. Для підвищення якості програмного забезпечення можна зосередитись на якості програмного забезпечення, наприклад, зробити його більш комфортним для користувача. Альтернативний підхід полягає в тому, що чим кращий процес, тим краща якість програмного забезпечення.

Другий напрямок пов'язаний або з відповідністю, або з удосконаленням. Під відповідністю будемо вважати відповідність якому-небудь стандарту. Вдосконалення має своєю метою перехід на більш вдосконалені методи і кращу практику для підвищення якості.

Першою організованою спробою розробки методів оцінки якості програмного забезпечення була почата Рубайном і Гартвиком [14, 12]. Метод, запропонований у цій роботі, полягав у тому, щоб визначити набір властивостей, що характеризує програму. Бажаними властивостям давався словесний опис, а як критерії виступали математичні вирази, у яких аргументами були параметри, що прямо або побічно описували конкретні властивості програми. Зокрема, називалися такі властивості, як "А₁ – правильність виконання математичних обчислень", "А₅ – зрозумілість програми", "А₆ – простота внесення змін", потім проводився

докладніший аналіз для визначення більш конкретних характеристик цих властивостей, які можна було б вимірювати і тим самим виявляти властивість характеристики якості для даної програми. І хоча така деталізація була виконана для кожної із запропонованих авторами характеристик якості, лише деяким з них вдалося поставити у відповідність виміряні показники, крім того, не наводилося жодних прикладів практичного застосування розробленого методу.

Розроблено сучасний програмний засіб для визначення якості ПЗ методами метричного аналізу, що дає змогу за допомогою показників якості розрахувати відповідні метрики і визначити значення комплексного показника якості програмного продукту [15]. З'ясовано особливості процесу оцінювання якості ПЗ, тобто проаналізовано поняття якості програмного продукту як предмету стандартизації, а також рівні подання моделі якості ПЗ, що дало змогу встановити можливість її підвищення шляхом формування відповідних вимог до критеріїв оцінювання якості, вдосконалення моделей метричного аналізу його якості та методів кількісного її вимірювання на всіх етапах реалізації програмного проекту. Встановлено особливості використання метричного аналізу для визначення якості ПЗ, згідно з якими існує відсутність єдиних стандартів на метрики, тому кожен постачальник вимірювальної системи пропонує власні методи оцінювання якості ПЗ і відповідні метрики. Також є складним завдання інтерпретації значень метрик, оскільки для більшості користувачів як метрики, так і їх значення не зовсім є зрозумілими та інформативними. Виявлено, що основними параметрами при виборі варіанту реалізації ПЗ є його вартість та тривалість процесу розроблення й репутація фірми-проектанта, але рішення прийняті на підставі цих параметрів, не завжди гарантують належну якість ПЗ.

На всіх етапах розробки ПЗ повинен проводитися такий контроль його якості: перевірка відповідності користувацьких вимог до ПЗ з критеріями і показниками його якості; перевірка та атестація проміжних результатів розроблення ПЗ на кожному з етапів реалізації програмного проекту і визначення ступеня задоволення досягнутих критеріїв і показників

його якості; тестування готового ПЗ, збирання даних про відмови, дефекти та інші помилки, які було виявлено в ПЗ під час його безпосередньої експлуатації; підбір моделей надійності ПЗ за отриманими результатами його тестування (дефекти, відмови тощо.); перевірка досяжності критеріїв і показників якості ПЗ, заданих у вимогах до нього.

Доведення правильності роботи ПЗ – це математична або логічна методика, яку використовують для того, щоб переконати розробника ПЗ й інших зацікавлених сторін у тому, що розроблене ПЗ відповідає тим вимогам, які було визначено та узгоджено з замовником. Таке доведення є формальним методом.

Незважаючи на представлений досить змістовний перелік додаткових методів оцінювання якості програмного забезпечення на мою думку, його варто використовувати як розширений методичний підхід.

Для створення методики оцінки якості програмного необхідно з'ясувати такі три моменти: 1) чи можна дати означення характеристикам якості ПЗ, які б можна було виміряти і забезпечити при цьому достатньо незалежну оцінку якості; 2) наскільки точно можна виміряти якість ПЗ в цілому і його окремі характеристики; 3) як інформацію про характеристики якості ПЗ використати для вдосконалення процесу його експлуатації надалі.

Складність полягає в тому, що існуючі показники якості ПЗ, як правило, неадекватно відображають ті чи інші їх властивості, що визначені потребами і вимогами користувача. Ефективну методику оцінки якісних показників доцільно робити на основі добре продуманих формувань відповідних характеристик з врахуванням їх важливості. Оскільки еталон програмного забезпечення не слід вважати абсолютним, загальний результат такої оцінки повинен розглядатись як інформація для проведення аналізу, а не як остаточний висновок. Три складові частини процесу створення якісного програмного забезпечення – це, по-перше, програмісти – розробники програм, по-друге, процеси, у вигляді яких організований випуск програмного забезпечення і, нарешті, технології, відповідно до яких реалізуються ці процеси. Очевидно, що

якість програмного продукту на виході технологічного ланцюжка визначається якістю його складових. Якість програмного забезпечення залежить від кваліфікації програмістів, склад яких неоднорідний у силу яскраво вираженої спеціалізації. Професійні виробники програмних продуктів давно вже переконалися, що найкращий спосіб поліпшити програму – це удосконалити процеси її створення на усіх етапах життєвого циклу.

В основі всіх цих концепцій лежить загальне розуміння життєвого циклу програмного забезпечення як сукупності фаз, що проходить програмний продукт у процесі свого розвитку. Це такі фази: 1) вироблення вихідних вимог до програмного забезпечення з боку користувача; 2) формулювання системних вимог до програмного забезпечення з боку розробника; 3) проектування архітектури; 4) детальна реалізація програмного забезпечення; 5) інсталяція ПЗ; 6) експлуатація.

Методи оцінки якості дозволяють переконатися, що певні характеристики якості програмного забезпечення досягнуті. Самі по собі вони не можуть допомогти їх досягнення, вони лише допомагають визначити чи вдалося отримати в результаті те, що хотілося, чи ні, а також знайти помилки, дефекти і відхилення від вимог. Методи оцінки якості програмного забезпечення можна класифікувати наступним чином: 1) методи, що зв'язані з виявленням властивостей програмного забезпечення під час його роботи; 2) методи визначення показників якості програмного забезпечення за допомогою моделей різного роду; 3) методи, що націлені на виявлення порушень формалізованих правил побудови початкового коду програмного забезпечення, проектних моделей і документації; 4) методи звичайного або формалізованого аналізу проектної документації і початкового коду для виявлення їх властивостей.

При дослідженні оцінки якості програмного продукту з'ясовано, що важливою є інформація про те, де і як з'являються дефекти програмного забезпечення, ніж про частоту їх появи. Найбільшою цінністю є такі автоматичні засоби аналізу якості програмного забезпечення, які не тільки діагностують програму, а і реєструють конкретні недоліки програми. Незважаючи на численні дослідження

програмних еталонів, у цій галузі, як і раніше, залишається багато невирішених питань. По-перше, технологія вимірювання якості ще не досягла зрілості. По-друге, відсутні єдині стандарти на еталони.

Фундаментом усіх методів оцінки якості є загальне розуміння життєвого циклу програмного забезпечення, як сукупності фаз, що проходить програмний продукт в процесі свого розвитку. Таким, зокрема є: вироблення початкових вимог до програмного забезпечення зі сторони користувача; формування загальних вимог до програмного забезпечення зі сторони розробника; проектування архітектури; детальна реалізація програмного забезпечення; інсталяція програмного забезпечення в організації замовника і його подальша експлуатація.

За допомогою експертних методів можуть бути отримані як вихідні дані для розрахунку оцінок значень показників якості, так і самі оцінки. Визначення характеристик показників якості експертним методом здійснюється групою експертів-фахівців, компетентних у вирішенні даного завдання. При цьому рішення базується на досвіді та інтуїції експертів. При використанні експертних методів необхідно оцінювати компетентність і сумлінність групи експертів. Визначення характеристик показників якості експертним методом передбачає наступний порядок дій: 1) підбір і підготовка групи експертів; 2) постановка завдання експертам; 3) контроль роботи експертів; 4) збір оцінок експертів; 5) оцінка компетентності та сумлінності групи експертів; 6) розрахунок експертної оцінки.

Методики отримання експертних оцінок значень показників якості з метою методологічної єдності повинні ґрунтуватися на загальному методі оцінки якості, правил вибору номенклатури показників якості та діючих методичних матеріалів по оцінці заданого показника якості ПЗ.

Розрахункові методи засновані на використанні теоретичних і емпіричних залежностей, статистичних даних, що накопичуються при випробуванні, експлуатації та супроводі ПЗ. За допомогою розрахункових методів визначаються тривалість і точність обчислень, час реакції, необхідні ресурси тощо.

Методи прийняття рішень про якість ПЗ можуть бути отримані при узагальненні оцінки

по групі показників якості, логічно пов'язаних та або не пов'язаних в номенклатурі показників якості, а також оцінки окремих показників якості і оцінка якості ПЗ в цілому.

Прийняття рішень про якість ПЗ здійснюється після того, як отримані необхідні оцінки показників якості. Як правило, цих оцінок може не вистачити для отримання узагальненої оцінки якості ПЗ в цілому, тому прийняття рішень про якість ПЗ здійснюється в умовах невизначеності і ризику. Залежно від особливостей ПЗ, що досліджується та характеру отриманих оцінок, завдань і цілей дослідження якості ПЗ ступінь невизначеності та ризику прийняття рішень про якість значно коливається. Основною причиною застосування методів прийняття рішень про якість ПЗ є відсутність інформації для завдання якості ПЗ на відповідній множині. Таким чином, завдання прийняття рішень зводиться до задачі визначення значущості отриманих оцінок показників якості.

До інших методів належить комплексний метод оцінки якості, заснований на логіці розумових заключень, що визначають якість ПЗ шляхом логічних операцій над характеристиками ПЗ і результатами його розробки та використання.

Найбільш поширеним серед інших методів оцінки якості ПЗ є метод порівняння ПЗ, що порівнюється з базовим зразком (еталоном). Цей метод передбачає: вибір базового зразка ПЗ; порівняння характеристик ПЗ, що оцінюється, з відповідними характеристиками базового зразка ПЗ; перерахунок оцінок показників якості базового зразка з порівнянням показників ПЗ, що оцінюється.

Розвиток ПЗ і розширення області застосування ПЗ обумовлюють можливість появи інших методів оцінки якості ПЗ. Склад методів оцінки якості ПЗ і їх класифікація є відкритими і допускають розширення.

Для розробки методу оцінки якості ПЗ необхідно вибрати модель оцінки програмного продукту. Першою широко відомою моделлю якості програмного забезпечення стала запропонована в 1977 МакКолом модель [4]. В ній характеристики якості розділені на три групи. 1. Фактори, які описують ПЗ з позицій користувача. Задаються вимогами. 2. Критерії,

які описують ПЗ з позицій розробника. Задаються як цілі. 3. Метрики, які використовуються для кількісного описання і вимірювання якості. Фактори якості, яких було виділено, групуються в три групи за різними способами роботи людей з ПЗ. Отримана структура відображається у вигляді трикутника МакКола Критерії якості – це числові рівні факторів, що визначені як цілі при розробці. Об'єктивно оцінити або виміряти фактори якості безпосередньо важко. Тому МакКол ввів метрики якості, які з його точки зору легше виміряти і оцінити. Кожна метрика впливає на оцінку деяких факторів якості. Числовий вираз фактору уявляє собою лінійну комбінацію значень впливаючих на нього метрик.

Під метриками вважають кількісну оцінку програмного продукту, процесу або проєкта, яка використовується безпосередньо або на основі якої здійснюються інші вимірювання чи виконується прогноз.

У 1978 Боем запропонував свою модель, яка по суті є розширенням моделі МакКола [6.]. Атрибути якості поділяють за способом використання ПЗ. Визначено 19 проміжних атрибутів, які включають усі 11 факторів якості по МакКолу. Проміжні атрибути розділяються на примітивні, які в свою чергу, можуть бути оцінені за допомогою метрик.

Така модель повинна бути побудована з врахуванням усіх складових якості програмного забезпечення і вимог міжнародних стандартів ISO на основі показників якості. Для практичної реалізації моделі пропонується використання системи підтримки і прийняття рішень на основі підходів багатокритеріального аналізу альтернатив і нечіткого моделювання.

Модель якості програмного забезпечення має наступні чотири рівня представлення згідно стандарту якості ISO 9126.

Перший рівень відповідає визначенню характеристик якості програмного забезпечення, кожна з яких відображає окрему точку зору користувача на якість, Згідно стандарту в модель якості входить шість характеристик або показників якості: функціональність (functionality); надійність (reliability); зручність використання або практичність (usability); продуктивність або ефективність (efficiency); зручність супроводу (maintainability); переносимість або мобільність (portability).

Другому рівню відповідають атрибути для кожної характеристики якості, які деталізують різні аспекти конкретної характеристики. Набір атрибутів характеристик якості викорчовують при оцінці якості.

Третій рівень призначений для вимірювання якості за допомогою метрик, кожна з яких згідно стандарту визначається як комбінація методу вимірювання атрибуту і шкали вимірювання значень атрибутів. Атрибут якості визначається за допомогою одною або декількох методик оцінки на етапах життєвого циклу і на завершальному етапі розробки програмного забезпечення.

Четвертий рівень – це оцінюючий елемент метрики (вага), який використовується для оцінки кількісного або якісного значення окремого атрибуту показника програмного забезпечення. В залежності від призначення, особливостей і умов супроводу програмного забезпечення вибираються найбільш важливі характеристики якості і їх атрибути.

Розглянемо критерії оцінки якості програмного забезпечення. Кожний програмний продукт повинен виконувати визначені функції, тобто робити те, що задумано розробниками. Якісний програмний продукт повинен мати набір властивостей, які дозволять успішно його використовувати упродовж тривалого періоду, тобто мати визначену якість. З визначення якості випливає, що, по-перше, якість поняття динамічне: оскільки з часом потреби можуть змінюватися, необхідно періодично переглядати і вимоги до якості. По-друге, оскільки якість – сукупність властивостей, необхідні показники для їх опису та кількісні характеристики для їх оцінки. Сукупність властивостей ПЗ, що забезпечує задовільну для користувача якість ПЗ, залежить від умов і характеру експлуатації цього ПЗ. Тому при опису якості ПЗ, насамперед, повинні бути визначені критерії оцінки якості ПЗ. В даний час критеріями якості ПЗ прийнято вважати ті характеристики якості програмного забезпечення, які входять в перший рівень: *функціональність* – здатність ПЗ виконувати певний набір функцій, які задовольняють потреби користувачів; *надійність* – здатність ПЗ підтримувати визначену працездатність у заданих умовах. *зручність використання або практичність* – це характеристика ПЗ, яка дозволяє мінімізувати

зусилля користувача по підготовці вхідних даних, і оцінці отриманих результатів; *продуктивність або ефективність* – це відношення рівня послуг, які надає ПЗ користувачу при заданих умовах, до обсягу використаних усіх типів ресурсів; *зручність супроводу* – це зручність проведення всіх видів діяльності, пов'язаних із супроводом програм і дозволяє мінімізувати зусилля по внесенню змін для усунення в ньому помилок та його модифікації відповідно до потреби користувача; *переносимість або мобільність* – здатність ПЗ зберігати працездатність при перенесенні з одного середовища в інше, включаючи організаційні, апаратні й програмні аспекти оточення.

Функціональність і надійність є обов'язковими критеріями якості ПЗ, причому забезпечення надійності червоною ниткою проходить по всім етапам і процесам розробки ПЗ. Зручність використання або практичність свідчить про результативність оцінки якості і вказує, в якій мірі було досягнуто мети, а ефективність дасть можливість охарактеризувати якісний ступінь задоволення потреб користувача. Інші критерії використовуються в залежності від потреб користувача у відповідності до вимог, що пред'являються до ПЗ.

Для того, щоб своєчасно враховувати фактори, що впливають на якість програмного забезпечення необхідно володіти знаннями, що складають ядро професійних навиків в області інформаційних технологій. Знання розробників програмних засобів відрізняються значною різноманітністю, які часто не повні, не узгоджені, спеціалізовані в певних предметних областях. В основі експертних систем, які використовуються при видобутку вуглеводнів є бази знань (БЗ).

Розглянемо застосування методики оцінки якості програмних продуктів на прикладі експертних систем, що використовуються в нафтогазовій галузі: POKLAD – для визначення наявності покладів флюїдів; COLECTOR – для визначення наявності нафтогазового колектора; PLAST – для вибору технології обмеження припливу пластових вод у свердловину; NAFTA – для вибору методу інтенсифікації видобутку нафти [16]. Створення наукових основ проектування експертних систем вимагає

дослідження проблем кількісної оцінки якості систем на різних стадіях їхнього життєвого циклу. Оскільки невід'ємними компонентами експертних систем є база знань і механізм логічного висновку, то виникає необхідність їх включення в загальну оцінку експертної системи якісних характеристик цих двох інтелектуальних компонентів.

Однією з основних задач проблеми оцінювання якості баз знань є вибір метрик і методів їхнього вимірювання. Проблема подання знань ускладнюється ще й деякими додатковими проблемами, включаючи подання знань в практичних додатках, таких як видоучення і внесення знань, ефективне подання знань, вибір інструментарію схем подання знань, модифікація знань при зміні середовища, оперування знаннями та використання відповідних знань в створенні рішень, що пропонуються. Під методикою у цьому випадку розуміють елементарну характеристику, що є характеристикою нижнього рівня, яка не розчленовується і підлягає безпосередньому вимірюванню.

Можна припустити, що по аналогії з базами даних відповідно до можливостей БЗ вони також утворюють чотири групи метрик адекватності БЗ як інформаційної моделі предметної області [17]: 1) метрики адекватності концептуальної схеми; 2) метрики адекватності логічної схеми; 3) метрики адекватності схеми бази знань; 4) метрики БЗ.

При виборі метрик і методів їхнього виміру варто керуватися такими положеннями: 1) метрики повинні бути коректно визначені, тобто метрики з різними іменами повинні мати різні визначення; 2) метрики повинні включати властивості об'єкта, і мати фундаментальні відмінності один від одного; 3) метрики повинні, з одного боку, мати практичну цінність, а з іншого – допускати максимальну можливість їхнього кількісного оцінювання; 4) метрики повинні брати участь у формуванні якості БЗ як у розрізі окремої інтегральної характеристики, так і загалом, забезпечуючи при цьому можливість використання єдиного, незалежного від номенклатури характеристик, алгоритму оцінювання; 5) методи виміру метрик повинні забезпечувати різну точність оцінок якості БЗ відповідно до висунутих до них вимог; 6) методи виміру метрик повинні бути

ефективними, тобто забезпечувати найбільш економічний вимір метрики.

Головним поняттям в системі оцінки якості загалом і програмного продукту зокрема, є поняття вимірювання, яке є відображенням фізичних величин їхніми значеннями за допомогою експерименту та обчислень із застосуванням спеціальних технічних засобів. У контексті програмного забезпечення вимірювання – множина однорідних значень, які відіграють роль індексів. Для певного вимірювання можуть бути визначені певні ієрархії значень. Це дозволяє здійснювати агрегацію показників. Найбільш раціональний спосіб дій з оцінки якості програмного забезпечення полягає в тому, щоб розробити певну систему індикації його недоліків і використовувати цю систему для подальшого вдосконалення програмних засобів.

Таким чином, стандарти, з одного боку, вимагають вимірювання показників якості, а з другого – в них відсутні конкретні переліки якості і методи їх оцінки. Крім цього, відсутні необхідні методичні рекомендації по аналізу відповідної інформації. Більш вагому користь при розробці індивідуального переліку показників розробник програмного забезпечення очікує отримати при використанні спеціальних міжнародних і вітчизняних нормативних документів.

Отже, для всіх етапів життєвого циклу програмних продуктів програміст повинен самостійно розробляти комплекси показників якості, які в сукупності утворюють систему показників. Він самостійно виявляти фактори, що впливають на якість. Тільки структурований індивідуальний підхід до вибору і утворення показників і факторів забезпечує ефективний контроль і управління якістю.

Вибір і обґрунтування факторів повинен здійснюватися комплексно, в залежності від прийнятої системи показників якості, висунутих гіпотез про причини зміни якості програмного продукту, встановлених цілей статистичного аналізу і інших причин.

Завжди є імовірність того, що не всі суттєві фактори включені до розгляду, а значення деякі з них не точні. Тому при виборі факторів більшу роль відіграє досвід експерта. Необхідно враховувати, що фактори здійснюють різний вплив на значення показників якості

При виборі методу оцінки якості програмного забезпечення має значення природа факторів - кількісна, якісна чи змішана. З нею зв'язані вид статистичної моделі якості програмного продукту і спосіб аналізу даних. Кількісні фактори оцінюють в основному регресійним аналізом, якісні дисперсійним аналізом, змішані коваріаційним аналізом [18].

Сукупність вибраних факторів не повинна містити протиречливих і взаємовиключних компонентів. З точки зору теорії планування результати експлуатації програмного продукту є пасивним експериментом, тому апріорне управління значеннями або факторами не можливе. Вони уявляють собою випадкові числа або події і повинні розглядатися як статистичні дані, отримані в результаті спостережень. Основою для вироблення ефективних дій на якість програмних продуктів служить оцінка факторів. Рішення про проведення такої оцінки приймається на відповідності показників якості встановленим вимогам. Фактори можна розглядати як причину зміни якості програмного продукту, а показники – як наслідок дії факторів. При цьому показники відображають структуру і значення факторів, що діють на програмний продукт і впливають на його якість.

Слід зауважити, що до негативних факторів, які впливають на якість програмного забезпечення відносяться: 1) внутрішні дефекти: проектування, алгоритмізації, програмування, захисту; 2) зовнішні дії: помилки персоналу, спотворення в каналах, відмовлення апаратури..

Підводячи підсумки аналізу проблеми оцінки факторів, що впливають на якість програмних продуктів, слід відзначити, що метод регресійного аналізу має ряд особливостей, які необхідно враховувати при його практичному застосуванні. Наприклад, для охоплення максимально можливого числа факторів і врахування динаміки необхідно суттєво збільшити розмірність задачі. Якщо ж період спостережень за результатами експлуатації програмного продукту незначний, то виникають труднощі з оцінкою адекватності моделі якості. Іншою важливою особливістю є необхідність використання математичних методів обчислення коефіцієнтів регресії. Але в цілому регресійний аналіз є працездатним

методом, забезпечує високу ефективність факторів, які впливають на якість програмних продуктів і можуть бути рекомендовані до практичного застосування.

Така методика не є єдино можливою. Основне, щоб запропонований спосіб був з ідейного, методологічного обчислювального боку достатньо пророблений і забезпечував практичну оцінку факторів і реальне управління якістю програмних продуктів.

Розглянемо фактори, які впливають на якість програмного коду: читабельність коду; легкість підтримки; низька складність коду; коректність обробки винятків.

Якість коду може визначатись різними критеріями. Деякі з них мають значення тільки з точки зору людини. Наприклад, форматування тексту програми є неважливим для комп'ютера, але може мати велике значення для супроводу. Багато з існуючих стандартів кодування, що визначають специфічні для мови програмування угоди та задають низку правил, мають на меті полегшити супровід ПЗ в майбутньому. Також існують інші критерії, що визначають чи вдало написаний код, наприклад, такі, як структурованість - ступінь логічного розділення коду на блоки.

Методом покращення якості коду є рефакторинг. Рефакторинг - це процес зміни коду без зміни функціональності. Тобто з точки зору користувача програма не змінюється. Але з точки зору програміста вона змінюється, причому у бік покращення, якщо рефакторинг проводиться грамотно. Рефакторинг полягає в чищенні програмного коду від дефектів і зміні архітектури з метою зробити його більш зрозумілим і гнучким [18, 5].

За наявності достатніх знань можна при бажанні записати до десяти рядків нормального коду в один рядок. Ціна за це буде невисокою - усього лише повна неможливість швидко виправити помилку в цьому коді. Якщо код, що створюється вимагає детального знання спеціальних правил пріоритету або змушує додатково робити аналіз літературних джерел, то цього не варта робити.

Зазвичай доводиться переносити або виклики функцій з великою кількістю аргументів. Але переносити треба обов'язково, так, щоб весь код було видно без прокручування його на екрані по горизонталі.

Багато середовищ програмування зараз автоматично вирівнюють код, але не всі, тому програмісту необхідно на це звернути увагу.

Щоб зробити програму високоякісну необхідно дотримуватися таких рекомендацій: 1. Код програми повинен бути модульним і добре написаним. 2. В розробці повинні використовуватися автоматичні тести, краще якщо тест написано до початку утворення коду, що тестується. 3. Необхідно мати хороший контакт з користувачами, які приймають участь в тестуванні та пропонували покращення. 4. Управління проектом повинно бути об'єктивним і далекоглядним. 5. Досить нав'язлива реклама шкідлива і цілком недопустима. 6. Важливим є вдала назва програми 7. Коментарі повинні бути частими [18].

Щоб зробити програмний код зрозумілішим і не створювати складнощів при подальшому супроводі необхідно обов'язково писати коментарі. Якщо програмний продукт не прокоментовано, то коли виникне потреба через певний час повернутися до нього для доопрацювання (а це часто доведеться робити), відсутність коментарів призведе до додаткових втрат робочого часу. Втрачений час неможливо компенсувати.

Слід, однак, зазначити, що написання коментарів - це теж мистецтво. Для досягнення майстерності у цьому виді діяльності необхідна практика. Коментарі бувають хороші й погані. Не треба писати занадто довгих коментарів. З іншого боку, не слід писати дуже коротких коментарів. Коментарі не тільки економлять час, але вони й самі потребують часу на прочитання, крім того, вони збільшують фактичні розміри програми на екрані монітора, в результаті чого необхідно одночасно зменшувати обсяг діючого програмного коду.

Кінцева мета така: написати програмний код таким чином, щоб стороння людина, не маюча уявлення про те, що цей код робить, могла зрозуміти його якомога швидше. Один з основних способів досягнення цієї мети полягає в тому, щоб давати змінним, функціям чи процедурам зрозумілі імена. При цьому необхідно давати об'єктам імена досить довгі й наочні для їх розуміння, однак не настільки довгі й громіздкі, щоб це ускладнювало читання програмного коду,

Якщо програма досить велика, в ній є безліч функцій і процедур, то потрібно кожен з них перевіряти на наявність помилок. Необхідно писати код так, щоб він перевіряв процедуру чи функцію на наявність підозрілих даних і захищався від них.

Переваги такого підходу не вичерпуються захистом програми від збоїв. Хороші механізми перевірки на наявність помилок також прискорюють налагодження програмного коду. Якщо в якій-небудь процедурі наявні всі механізми перевірки, то не доведеться проходити її крок за кроком у пошуках помилок.

Слід зазначити, що оптимізація часто йде в суперечності, у деяких випадках оптимізація необхідна. Проте майже ніколи не відомо заздалегідь, що саме необхідно оптимізувати, до тих пір, поки не протестовано реально функціонуючий код за допомогою інструменту під назвою профайлер (профайлер - це програма, яка спостерігає за програмним кодом і оцінює час, що витрачається на виконання окремих викликів) [19].

Використовуючи [профайлер](#) можна відзначити декілька основних проблем. По-перше, вважається, що це інструмент призначений тільки для програміста. Результати коду є настільки технологічними, що мало зрозумілі для тих, хто зацікавлений в підвищенні якості програмного продукту, але не готові вникати в деталі. По – друге, відносно поняття “якості” програмного продукту все ще поширене досить вузьке розуміння питання. По – третє, є цілком певний конфлікт інтересів. Програміст може бути зовсім не зацікавлений в тому, щоб узнати про свій код усі деталі. Для розробника програмного коду найціннішим ресурсом є час. Керівник розробки і так знаходиться в постійному цейтноті, планує терміни виходу версій і формує склад змін майбутньої версії. Про технічний борг він і так знає. Ну, виявить, аналізатор ще деякі промахи і недоліки, то поставить у чергу в кращому випадку. Тестувальники зайняті і мотивовані виявленням помилок у функціональності, а їх керівники не бачать загальної картини того, як добре чи погано програма в дійсності зроблена.

Код високоякісної програми повинен бути на стільки простим, на скільки це можливо. Простий код легший у читанні та у підтримці.

Оскільки він коротший і зрозуміліший, то значить, містить менше помилок,

Користувач в основному проявляє зацікавленість в застосуванні програмного забезпечення, його продуктивності і результатах використання. Користувачі оцінюють програмне забезпечення без вивчення його внутрішніх аспектів або того, як вона утворювалася. Користувача можуть цікавити наступні питання: 1. Чи є необхідні функції в програмному забезпеченні? 2. Наскільки надійне програмне забезпечення? 3. Наскільки ефективне програмне забезпечення? 4. Чи являється програмне забезпечення зручним для користувача? 5. Наскільки просто переноситься програмне забезпечення в інше середовище.

З точки зору користувача та розробника процес створення ПЗ вимагає використання одних і тих же характеристик якості програмного забезпечення, оскільки вони застосовуються для встановлення вимог і прийомів. Коли розробляється програмне забезпечення для продажу, у вимогах якості повинні бути відображені передбачені потреби. Так як розробники відповідають за створення програмного забезпечення, яке повинно задовольняти вимогам якості, вони зацікавлені в якості проміжних програмних кодів так само, як і в якості кінцевого програмного продукту.

Керівник нафтогазовидобувного підприємства може бути більш зацікавлений в загальній якості, ніж в конкретній характеристиці якості, і з цієї причини буде потребувати у визначені важливості значень, що відображають комерційні вимоги для індивідуальних характеристик. Керівникові можливо також знадобиться співставлення підвищення якості з критеріями управління, такими як планова затримка або перерозхід вартості, тому що він бажає оптимізувати якість програмного забезпечення в межах обмеженої вартості, трудових ресурсів і встановленого терміну його розробки.

Кожна компанія прагне зайняти саме високе положення на IT – ринку, тому повинна прикладати зусилля для створення якісного програмного забезпечення, яке подобається користувачам. На жаль, деколи тиск замовників заставляє розробників нехтувати якістю програмного забезпечення, і вони поставляють готовий продукт, що невідповідає очікуванням.

Низька якість проявляється тільки тоді коли програмне забезпечення перестає працювати.

Виявляється, що достатньо якісне програмне забезпечення створити досить складно. Це пояснюється сукупністю таких причин [20]: 1. Часто якість прагнуть визначити тільки в термінах дефектів (помилки). З точки зору користувача не менш важливим є готовність до певної дати. 2. Передбачається, що незначна кількість помилок рівносильна кращій якості, і користувач надає перевагу якраз такій якості. Але інколи користувач готовий іти на компроміси і примиритися з декількома помилками в обмін на більш швидке виконання роботи. 3. Ігнорування таких факторів, як моральний стан команди програмістів і адекватне умови для роботи.

Нажаль, “високоякісне” програмне забезпечення рідко буває достатньо високоякісним. Цікаве спостереження, яке полягає в тому, що деякі команди прагнуть до зниження інтелектуального рівня показників свого програмного продукту, що підкреслюють багато програмістів. IT-компаніям надзвичайно вигідно мати справу з людьми, технічна класифікація яких не дозволяє визначити реальні аспекти (наприклад, якість, складність, цінність) програмного продукту. Прикриваючись “спрощенням” роботи і підвищенням доступності комп’ютера для користувачів, компанії багатократно перевантажують і ускладнюють програмне забезпечення таким чином, щоб користувачеві було досить важко зрозуміти яким чином воно працює насправді і стати професіоналом.

А замовник буде задоволений і приведе з собою ще замовників, тільки в тому випадку, якщо куплений програмний продукт буде повністю задовольняти його реальним і життєвим потребам, як специфікованим, так і не специфікованим. Але все це не настільки абстрактно, на скільки може показатися з першого разу.

Якість ПЗ має цілком певний вартісний вираз. Якраз це заставляє програмістів забезпечувати необхідний рівень якості програмних продуктів. Економічні наслідки помилок в них можуть бути значні. Часті зусилля по ліквідації причин, що викликають претензії користувачів програмних продуктів

здатні довести до банкрутства будь-яку ІТ-компанію.

Концепція загальних втрат передбачає, що головним є загальні втрати, які утворюються в результаті недоліків даного програмного продукту. Вартість якості ПЗ може бути диференційована на вартість попередження дефектів, вартість оцінювання ефективності роботи, вартість внутрішніх і зовнішніх збоїв під час роботи ПЗ. Рушійною силою успішної реалізації програмних проектів є бажання їх керівників розробити таке ПЗ, яке б володіло певною цінністю, тобто було значущим для вирішення певних завдань або досягнення цілей – тактичних і стратегічних. Цінність ПЗ можна виражати у формі його вартості або в деякому іншому вигляді. Замовник, зазвичай, має своє уявлення про максимальну вартість вкладення коштів, повернення яких він очікує в разі досягнення основних цілей під час використання ПЗ. Він також може мати своє бачення щодо функціональних можливостей ПЗ та певні очікування щодо його якості. Замовник спочатку концентрує свою увагу на функціональних можливостях ПЗ й не замислюється над питаннями його якості, а тим більше над пов'язаною з ними вартістю його розроблення. Тому на початковому етапі реалізації програмного проекту предметом обговорення може стати питання про повне розуміння замовником вигоди від використання ПЗ і вартості його розроблення, пов'язаної з досягненням того чи іншого рівня якості ПЗ, а також про його залучення до процесу прийняття відповідних рішень. В ідеальному випадку більшість таких рішень потрібно приймати на етапі встановлення користувацьких вимог до ПЗ, але ці питання можуть підніматися протягом всіх етапів його розроблення. Не існує якихось "стандартних" правил щодо того, як саме необхідно приймати такі рішення. Однак, системні інженери повинні чітко уявляти різні альтернативні способи досягнення певного рівня якості ПЗ та відповідну вартість його розробки для того, щоб можна було спрогнозувати загальну вартість реалізації програмного проекту [21].

Таким чином, в статі сформульовано та розглянуто методологічні підходи щодо оцінки якості програмного забезпечення для об'єктів нафтогазового комплексу.

Розглянуто сучасний стан проблеми та попередні дослідження у цій сфері, а також основні, хоча і далеко не всі, методики оцінки якості програмного забезпечення.

Аналіз основних характеристик програмного забезпечення, таких як: функціональні можливості, надійність, практичність, ефективність, супровід, мобільність показав, що вони можуть бути покладені в основу споживчих оцінок якості програмних продуктів, а в формалізованому вигляді - використані в узагальненій моделі оцінки якості програмного забезпечення інформаційних систем, що можуть бути впроваджені в нафтогазовій галузі.

Описано і проаналізовано деякі з проблем, що пов'язані зі створенням і застосуванням методик оцінки якості ПЗ загалом і зокрема баз знань в нафтогазовій предметній області.

Література

1. Соммервил І. Інженерия програмного обеспечения. 6-издание.– Москва–Санкт–Петербург–Киев, 2002 –623 с.
2. ISO/IEC 12207:1995. Information technologies. Software lifecycle processes. // ISO/IEC. -1995. – 61 р.
3. Бутаков Е. А. Методы создания качественного программного обеспечения ЭВМ. — М.: Энергоатомиздат, 1984. — 232с. — С. 114.
4. В. В. Кулямин. Технологии программирования. Компонентный подход. М.: ИНТУИТ-Бином, 2007. 463 с.
5. Андон Ф.И., Коваль Г.И., Коротун Т.М. Основы инженерии качества программных систем. 2-е издание. — К.: Академперіодика, 2007. — 672 с.
6. Бозм Б., Браун Дж., Каспар Х. и др. Характеристики качества программного обеспечения. – М.: Мир, 1981. – 206 с.
7. Жарко Е.Ф. Сравнение моделей качества программного обеспечения: аналитический подход / XII Всероссийское совещание по проблемам управления. ВСПУ-2014. Москва, 16-19 июня 2014 г.: Труды. М.: ИПУ РАН, 2014. С. 4585-4594.

8. Воробьев В. И., Копыльцов А. В., Пальчун Б. П., Юсупов Р. Методы и модели оценивания качества программного обеспечения. М. С-Пб.: СПИИРАН.1992.-33с.

9. Липаев В.В. Обеспечение качества программных средств. Методы и стандарты. – М.: Синтег, 2001. 380 с.

10. Кулаков А.Ю. Оценка качества программ ЭВМ. Киев: Техніка.1984. 167с.

11. Майерс Г. Надежность программного обеспечения / Майерс Г. М.: Мир, 1980. – 360 с.

12. Вовк О. Б. Аналіз та оцінювання якості програмного продукту (поняття, терміни, означення)/ О. Б. Вовк // Вісник Національного університету "Львівська політехніка". 2003. № 489 :Інформаційні системи та мережі. – С. 64-73

13. Стив Макконелла Профессиональная разработка программного обеспечения. СПб.: Символ-Плюс, 2006. - 240 с.,

14. Rubey R.J., Hartwick R.D. Quantitative Measurement of Program Quality, Proceedings, ACM National Conference. – 1986. – P. 671–677.

15. Грицюк Ю. І., Андрущакевич О. Т. Засіб для визначення якості програмного забезпечення методами метричного аналізу. Науковий вісник НЛТУ України. 2018, т. 28, № 6. С. 159–171.

16. Юрчишин В.М., Шекета В.І., Юрчишин О.В. Інформаційне моделювання нафтогазових об'єктів - Івано-Франківськ; Видавництво Івано-Франківського національного технічного університету нафти і газу 2010 -192с.

17. . Оценка качества баз данных. Метрики адекватности баз данных. Методические материалы // ИПС АН Украины. – 1994. – 57

18. Стив Макконелл. Совершенный код. Мастер-класс / Пер. с англ. — М. : Издательско-торговый дом «Русская Редакция» ; СПб.: Питер, 2005. — 896 с.

19. Кнут Д. Э. Искусство программирования. Том 1. Основные алгоритмы — Москва: Вильямс, 2002. — Т. 1. — 720 с.

20. Коваль Г.И. Подход к прогнозированию надежности ПО при управлении проектом // «Проблемы программирования». –2002. - № 1 – 2. – С. 282 – 290.

21. Стив Макконелл. Сколько стоит программный проект ; СПб.: Питер, 2007. — 304 с.

References

1. 1.Sommervil I. Inzheneriyaprogrammnogoobespecheniya. 6-izdanie.– Moskva–Sankt– Peterburg–Kiev, 2002 – 623 s. [in Russian]

2. 2.ISO/IEC 12207:1995. Information technologies. Software life cycle processes. // ISO/IEC. -1995. – 61 p.

3. 3.Butakov E. A. Metodysozdaniyakachestvennogoprogrammnogoobespecheniya EVM. — М.: Energoatomizdat, 1984. — 232s. — S. 114.

4. 4.V. V. Kulyamin. Tekhnologiiiprogrammirovaniya. Komponentnyjpodhod. М.: INTUIT-Binom, 2007. 463 s.

5. 5.Andon F.I., Koval' G.I., Korotun T.M. Osnovyinzheniikachestvaprogrammnyhsistem. 2-e izdanie. — К.: Akadempriodika, 2007. — 672 s.

6. 6.Boem B., Braun Dzh., Kaspar H. i dr. Harakteristikikachestvaprogrammnogoobespecheniya. – М.: Mir, 1981. – 206 s.

7. 7. Zharko E.F. Sravneniemodelejkachestvaprogrammnogoobespecheniya: analiticheskijpodhod / XII Vserossijskoesoveshchaniepoproblemamupravleniya. VSPU-2014. Moskva, 16-19 iyunya 2014 g.: Trudy. М.: IPU RAN, 2014. S. 4585-4594.

8. 8.Vorob'ev V. I., Kopyl'cov A. V., Pal'chun B. P., YUsupov R. Metodyimodeliocenivaniyakachestvaprogrammnogoobespecheniya. М. S-Pb.: SPIIRAN.1992.-33s.

9. 9.Lipaev V.V. Obespecheniekachestvaprogrammnyhsredstv. Metodyistandarty. – М.: Sinteg, 2001. – 380 s.

10. 10.Kulakov A.YU. OcenkakachestvaprogrammEVM.–Kiev: Tekhnika.–1984.–167s.

11. 11.Majers G. Nadezhnost' programmogoobespecheniya / Majers G. – М.: Mir, 1980. – 360 s.

12. 12.Vovk O. B. Analiz ta ocinyuvannyayakostiprogramnogoproduktu (ponyattya, termini, oznachennya)/ O. B. Vovk // VisnikNacional'nogouniversitetu "L'vivs'kapolitekhnika". – 2003. – № 489 :Informacijnisistemi ta merezhi. – S. 64-73

13. 13.Stiv MakkonnellaProfessional'nayarazrobotkaprogrammn ogoobespecheniya. SPb.: Simvol-Plyus, 2006. - 240 s.
14. 14.Rubey R.J., Hartwick R.D. Quantitative Measurement of Program Quality, Proceedings, ACM National Conference. – 1986. – P. 671–677.
15. 15.Gricyuk YU. I., Andrushchakevich O. T. Zasibdlyaviznachennyayakostiprogramnogo zabezpe chennyametodamimetrichnogoanalizu. Naukovijvisnik NLTU Ukraïni. 2018, t. 28, № 6. S. 159–171.
16. 16.Yurchishin V.M., Sheketa V. I, Yurchishin O.V. Informacijnemodelyuvannyanaftogazovihob'ektiv - Ivano-Frankivs'k; VidavnictvoIvano-Frankivs'kogonacional'nogotekhnichnogouniversitet unafti i gazu 2010 -192s.
17. 17.Ocenka kachestvabazdannyh. Metrikiadekvatnostibazdannyh. Metodicheskimaterialy // IPS AN Ukrainy. – 1994. – 57
18. 18.Stiv Makkonnell. Sovershennyjkod. Master-klass / Per. s angl. — M. :Izdatel'sko-torgovyjdom «RusskayaRedakciya» ; SPb.: Piter, 2005. — 896 s.
19. 19.Knut D. E. Iskusstvoprogrammirovaniya. Tom 1. Osnovnyealgoritmy — Moskva: Vil'yams, 2002. — T. 1. — 720 s.
20. 20.Koval' G.I.Podhod k prognozirovaniyunadezhnosti POpripravleniiiproektom // «Problemyprogrammirovaniya». –2002. - № 1 – 2. – C. 282 – 290.
21. 21.Stiv Makkonnell. Skol'kostoitprogrammnyjproekt ;SPb.: Piter, 2007. — 304 s.