

що похибка перетворення типу "контур – контур" пропорційно залежить від похибок апроксимації функції контурів зображень.

Література

1. Физика визуализации изображений в медицине : пер. с англ. – В 2 т. – Т. 1 / под ред. С. Уэбба. – М. : Изд-во "Мир", 1991. – 408 с.
2. Афанасьев Ю.И. Гистология, цитология и эмбриология / Ю.И. Афанасьев, Н.А. Юрина, Е.Ф. Котовский и др. – М. : Изд-во "Медицина", 2003. – 737 с.
3. Pratt W. Digital image processing: PIKS Scientific inside, 4th ed / William K. Pratt. – USA: John Wiley & Sons, 2007. – 782 p.
4. Абрамэйко С.В. Медицинские информационные технологии и системы / В.В. Анищенко, В.А. Лапицкий, А.В. Тузиков. – Минск : ОИПИ НАН Беларуси, 2007. – 176 с.
5. Абрамэйко С.В. Обработка оптических изображений клеточных структур в медицине / С.В. Абрамэйко, А.М. Недзьведь; Объед. ин-т проблем информатики. – Минск : ОИПИ НАН Беларуси, 2005. – 156 с.
6. Ахметшин А.М. Повышение чувствительности ультразвуковой диагностики на основе псевдоэллипсометрических параметров аналитических импульсных характеристик / А.М. Ахметшин, А.А. Степаненко // Автоматика. Автоматизация. Электротехнические комплексы и системы. – 2005. – № 1 (15). – С. 90-95.
7. Кожем'яко В.П. Оптико-електронні методи і засоби для оброблення та аналізу біомедичних зображень / В.П. Кожем'яко, С.В. Павлов, К.І. Станчук. – 2006. – 203 с.
8. Недзьведь А.М. Современные возможности обработки изображений при морфологическом исследовании карцином щитовидной железы / А.М. Недзьведь, М.В. Фридман, В.Е. Папок // Медицинские новости : науч.-практ. информац.-аналит. журнал для врачей и руководителей здравоохранения – 2006. – № 12. – С. 115-119.
9. Скобцов Ю.А. Моделирование и визуализация поведения потоков крови при патологических процессах / Ю.А. Скобцов, Ю.В. Родин, В.С. Оверко. – Донецк : ИПММ НАНУ, Изд-во Заславский, 2008. – 212 с.
10. Тузиков А.В. Построение поверхностных и объемных моделей объектов медицинских изображений / Снежко Эдуард Витальевич, Тузиков Александр Васильевич // Таврический Вестник информатики и математики – 2006. – № 1. – С. 91-96.
11. Автандилов Г.Г. Медицинская морфометрия. Руководство. / Г.Г. Автандилов. – М. : Изд-во "Медицина", 1990. – 384 с.
12. Автандилов Г.Г. Основы количественной паталогической анатомии / Г.Г. Автандилов. – М. : Изд-во "Медицина", 2002. – 238 с.
13. Егорова О.В. Компьютерная микроскопия / О.В. Егорова, Е.И. Клыккова, В.Г. Пантелеев. – М. : Вид-во "Техносфера", 2005. – 300 с.
14. Березский О.Н. Топологические методы и алгоритмы преобразования контуров и областей плоских изображений / О.Н. Березский // Проблемы управления и информатики. – 2010. – № 5. – С. 123-131.
15. Березький О.М. Методи та алгоритми перетворення контурів зображень в афінному просторі // Вісник Національного університету "Львівська політехніка". – Сер.: Комп'ютерні науки та інформаційні технології. – Львів : Вид-во НУ "Львівська політехніка". – 2009. – № 638. – С. 185-189.
16. Косневски Ч. Начальный курс алгебраической топологии / Ч. Косневски. – М. : Изд-во "Мир", 1983. – 304 с.
17. Березький О.М. Алгоритм проходження контуром об'єкта з використанням зворотного ходу / О.М. Березький, Ю.М. Батько // Штучний інтелект : наук.-теорет. журнал. – 2009. – № 3. – С. 116-122.

Березский О.Н. Исследование погрешности преобразования контуров биомедицинских изображений

Проведен анализ характерных признаков гистологических и цитологических изображений и автоматизированных систем обработки биомедицинских изображений. Исследованы составляющие погрешности преобразования типа "контур – контур" и осуществлена их оценка. Проведены компьютерные эксперименты определения погрешности преобразования контуров на примере цитологических изображений.

Ключевые слова: преобразование, цитологические изображения, погрешность, контур.

Berezsky O.N. Research error of biomedical image contours transformation

The analysis of histological and cytological image characteristic features and automated systems of biomedical image processing is carried out. The errors components of "contour – contour" transformation are analysed and their estimation is carried out. The computer experiments of contours transformation errors are conducted on the example of cytological image.

Keywords: transformation, cytological image, error, contour.

УДК 004.4

Доц. В.В. Яцишин¹, канд. техн. наук; доц. О.Г. Харченко², канд. техн. наук; проф. О.А. Пастух¹, д-р техн. наук; асист. І.О. Боднарчук¹

МЕТОД ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ПРОГРАМНИХ СИСТЕМ НА РАННІХ СТАДІЯХ ЖИТТЄВОГО ЦИКЛУ

Розглянуто метод забезпечення якості вимог, що є одним з аспектів управління та забезпечення якості програмних систем. На основі теоретико-множинних нотацій проведено формалізацію вимог якості до програмного забезпечення, проаналізовано критерії якості вимог та запропоновано способи і методи їх інтеграції у процес розробки ПЗ. Розроблено алгоритм збирання і класифікації вимог якості, процедуру побудови вимог до програмних систем на основі моделей якості стандарту ISO 25010, що дало змогу застосувати метод комунікації вимог якості на стадіях життєвого циклу.

Ключові слова: якість вимог, якість програмного забезпечення, критерії якості вимог, управління якістю програмного забезпечення.

Вступ. Актуальність. Сучасне програмне забезпечення характеризується високою функціональною інтегрованістю, мобільністю, кросплатформністю та орієнтацією на широке коло користувачів. Тому важливим і одним з основних аспектів розробки будь-якого програмного забезпечення є гарантування якості кінцевого продукту. Саме якість продукту безпосередньо впливає на його конкурентоспроможність та можливість масового використання.

Як показують дослідження [1, 2], якість програмних продуктів залишається не надто високою. Основними негативними факторами впливу на якість кінцевого продукту є слабка формалізація вимог якості, відсутність методів та процедур їх комунікації на стадіях життєвого циклу, а також термінологічні розбіжності між замовниками та розробниками ПЗ. Для нівелювання негативного впливу, наведених вище факторів на якість кінцевого програмного продукту, необхідно вирішити низку проблем, пов'язаних з інтерпретацією вимог до ПЗ та інтеграцією процесу управління вимогами на стадіях життєвого циклу.

Продукто-орієнтовані та процесо-орієнтовані підходи до розробки програмного забезпечення не гарантують якості продукту, оскільки існують потенційні ризики перевищення планової вартості та часу виконання проекту. Це пов'язано з недосконалістю методів контролю та управління якістю ПЗ на стадіях проектування або ж орієнтацією лише на якість виконання процесів життєво-

¹ Тернопільський НТУ ім. Івана Пулюя;

² Національний авіаційний університет, м. Київ

го циклу. Тому актуальним завданням у сфері забезпечення якості ПЗ є створення нового підходу, який би поєднував переваги продукто- та процесорієнтованих підходів до розроблення програмного забезпечення.

1. Метод забезпечення якості вимог до програмних систем. Оскільки вимоги є фундаментом у плануванні проекту, то для забезпечення якості програмного продукту, передусім, потрібно забезпечити якість самих вимог. Для цього необхідно визначити критерії якості, виходячи з рекомендацій міжнародних стандартів та практики розроблення програмного забезпечення.

Стандарт [5] визначає такі основні критерії якості вимог:

- коректність – відповідність реальним потребам замовника.
- однозначність – однозначність розуміння (трактування) вимог.
- повнота – вимоги мають відображати всі основні потреби у ПС.
- узгодженість – узгодженість між різними типами вимог.
- впорядкованість – впорядкованість вимог за важливістю і стабільністю.
- тестованість – можливість перевірки виконання вимог.
- модифікованість – здатність до редагування (додавання, видалення) вимог.
- простежуваність – можливість пов'язати вимогу з підсистемами, модулями та операціями, які відповідають за їх виконання, і з тестами, що перевіряють їх виконання.

Для забезпечення якості готового програмного продукту вимоги до якості ПС повинні бути простежуваними на всіх стадіях ЖЦ, а тому їх представлення мають задовольняти такі вимоги [4, 7]:

- бути зрозумілими та формалізованими;
- бути об'єктивними і вимірюваними;
- містити критерії оцінювання;
- бути відстежуваними і контрольованими на стадіях ЖЦ.

Виходячи з практики розроблення програмного забезпечення, процес управління якістю вимогами має базуватись на структурованих та документованих вимогах. З метою забезпечення однозначності, коректності та повноти вимог до програмного забезпечення запропоновано їх представляти в термінах моделей якості стандарту ISO 25010 з попередньо проведеною формалізацією у вигляді теоретико-множинних нотацій.

На стадії аналізу вимог до ПЗ визначено низку процесів, першим з яких є процес збирання вимог (потреб). Зазвичай, такі потреби отримують шляхом анкетування, інтерв'ю та аналізу предметної області у довільній текстовій формі. Для формалізованого представлення вимог запропоновано такий вираз:

$$\{P_i, C_{ik}\}, i = \overline{1, I}, k = \overline{1, K_i}, \quad (1)$$

де: P_i – вимоги користувача в текстовому вигляді, C_{ik} – обмеження на потреби.

Приведення вимог (1) до уніфікованого вигляду виконано шляхом їх відображення на структуру моделі якості у використанні стандарту ISO 25010:

$$R_{use} = \{H_i^u, A_{ij}^u, C_{ij}^u, M_{ij}^u\}, i \in N_u, j = \overline{1, F_i^u}, \quad (2)$$

де: H_i^u – характеристики моделі якості у використанні, A_{ij}^u – атрибути якості; M_{ij}^u – метрики моделі якості у використанні.

Для задоволення і подальшого оцінювання критеріїв простежуваності та здатності до модифікації вимог до ПЗ запропоновано таку процедуру відображення (1) на структуру (2):

- представлення (1) у вигляді шаблону $\{s_1, s_2, s_3, s_4\}$ де: s_1 – поле "ідентифікатор вимоги", s_2 – поле "назва компоненту, до якого сформульована вимога", s_3 – поле "атрибут або характеристика якості", виділені з тексту (1) та s_4 – поле "метрика вимірювання";
- класифікація атрибутів s_3 за стандартизованими наборами характеристик і метрик з використанням бази знань, сформованої експертним шляхом. У базі знань містяться асоціації між атрибутом шаблона та стандартною характеристикою і відповідним їй атрибутом якості, визначеним з аналізу предметної області та специфіки класу, до якого належить ПЗ. Класифікацію проводять шляхом пошуку в базі знань такої пари $\{s_{1n}, s_{2n}, s_{3n}, s_{4n}\}$ та $\{H_i^u, A_{ij}^u, M_{ij}^u\}$ для якої виконується нерівність $\{Supp_l\} \supseteq \{\overline{Supp_l}\}$, де $\{Supp_l\}, l = \overline{1, L}$, – підтримка асоціації, $\{\overline{Supp_l}\}$ – визначений граничний рівень асоціації.

Оскільки на якість програмного продукту впливають ризики, пов'язані з невиконанням або частковим виконанням вимог, тому для кожного атрибута A_{ij}^u моделі якості стандарту доцільно встановлювати ступінь ризику $Risk\{F_{ij}^u\}$, F_{ij}^u – фактор ризику. Ризики можна категоризувати відносно характеристик якості стандарту ISO 25010. Тоді основним завданням гарантування, а відтак і управління якістю ПЗ, є мінімізація ризиків, тобто

$$Risk\{F_{ij}^u\} \rightarrow \min. \quad (2)$$

Однак при мінімізації ризиків необхідно враховувати ще й фактори терміну і вартості їх усунення.

Комунікацію (трасування) вимог якості на стадіях життєвого циклу запропоновано реалізувати відповідно до методології QFD, основою якої є встановлення залежності між користувацькими властивостями та технічними характеристиками продукту. Суть методу, стосовно задачі комунікації вимог, полягає у встановленні залежності між вимогами якості на суміжних стадіях життєвого циклу, шляхом побудови "будинку якості". Схему застосування цього методу наведено у [9].

2. Алгоритм розробки вимог до програмних систем на основі моделей якості стандарту ISO 25010. Загальний процес розробки вимог до ПС, згідно з [8], починають з аналізу предметної області, на основі якого проводять збір і класифікацію вимог, а також визначають їх пріоритетність. При цьому розробники для формування та аналізу вимог використовують схему, наведену на рис. 1.

Ця схема (див. рис. 1) є узагальненою і не уніфікованою, оскільки кожен розробник використовує освоєні ним технології збирання, класифікації і специфікації вимог. При цьому якість процесу розробки вимог прямо залежить від цих технологій.

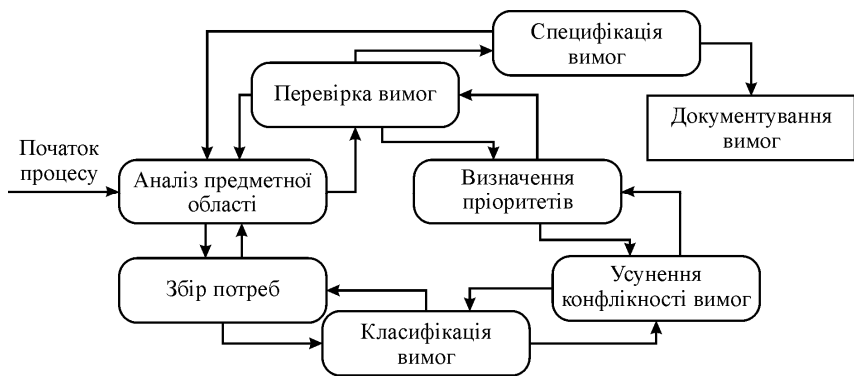


Рис. 1. Узагальнена схема алгоритму розробки та аналізу вимог до ПС

Для того, щоб структурувати і забезпечити комунікацію вимог, а також одночасно стандартизувати вимоги в загальному випадку, ми розробили алгоритм, який наведено на рис. 2.

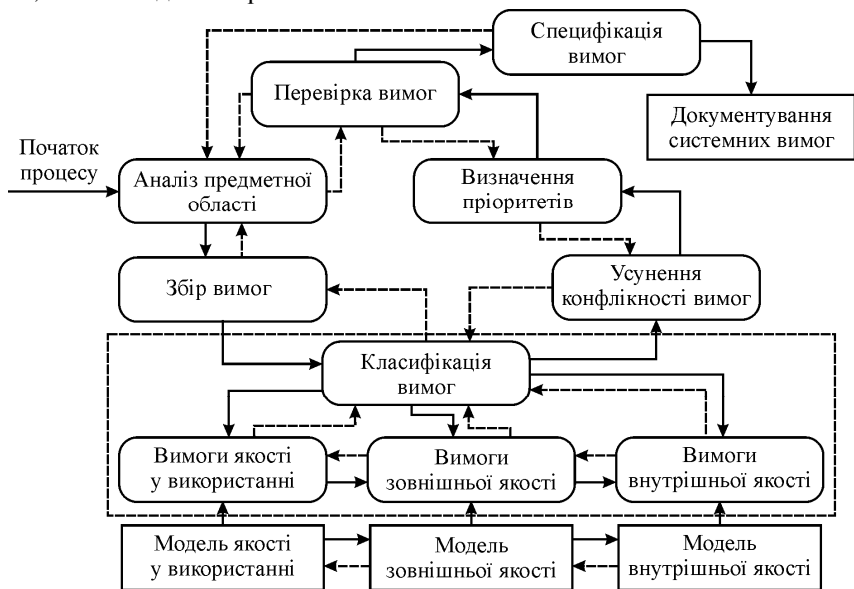


Рис. 2. Схема розробки та аналізу вимог до ПС на основі моделей якості стандарту ISO 25010

З аналізу схеми алгоритму процесу розробки та аналізу вимог, наведеного на рис. 2, видно, що формування вимог здійснюється за трьома моделями якості, характеристики яких є стандартизованими.

Алгоритм, наведений на рис. 2, є загальним відносно до реалізації самої процедури побудови моделей якості на етапі розробки вимог. Дамо детальніший його виклад, починаючи з аналізу предметної області і закінчуючи специфікацією системних вимог.

Замовник програмного продукту формулює свої потреби, які визначають функціональність ПС і загальний вигляд інтерфейсу, а також вимоги до якості, які виділяються з потреб. При цьому представлення потреб може бути подане у довільній формі, шляхом опису майбутніх функцій ПС та особливостей інтерфейсів, які здебільшого відповідають критеріям простоти та зрозумілості.

На наступному кроці замовник спільно з розробником проводять аналіз предметної області і сформульованих потреб $R_c = \{P_i, C_{ik}\}, i = \overline{1, N}, K = \overline{1, M_i}$. Внаслідок аналізу визначаються вимоги якості користувача (замовника), які записуються у вигляді множини атрибутів $\{A_{ik}\}, K = \overline{1, S_i}$.

На основі сформованого набору атрибутів $\{A_{ik}\}, K = \overline{1, S_i}$ будується модель якості у використанні. Основні завдання, які при цьому необхідно виконати, полягають у класифікації атрибутів за відповідними характеристиками моделі якості у використанні та розв'язанні конфліктності атрибутів.

Для класифікації атрибутів за характеристиками моделі якості у використанні розроблено шаблон, який наведено у вигляді табл. При цьому передбачається розподіл атрибутів множини $\{A_{ik}\}, K = \overline{1, S_i}$ за стандартизованими характеристиками моделі якості у використанні $\{H_i^u, M_{ij}^u\}, i = \overline{1, 4}, j = \overline{1, B_i}$. Наступна ітерація полягає в уточненні та узгодженні з розробником ПС моделі якості у використанні на основі аналізу вибраних метрик. Внаслідок цього одержуємо вимоги якості у використанні R_{use} , представлені у структурованому, стандартизованому вигляді $\{H_i^u, A_{ik}^u, C_{ik}^u, M_{ij}^u\}, i \in N_{i_u}, K = \overline{1, S_i}$.

Після побудови моделі якості у використанні будується модель зовнішньої якості. Для цього, з множини атрибутів, які не увійшли до складу першої моделі, проводиться класифікація за стандартизованими характеристиками та підхарактеристиками моделі зовнішньої якості $\{H_i^s, P_{ij}^s, M_{ij}^s\}, i = \overline{1, 6}, j = \overline{1, K_i}$. Процедура здійснюється знову ж таки на основі шаблону (див. табл.)

Табл. Шаблон класифікації атрибутів за структурою моделей якості

Характеристика	Інтерпретація характеристики моделі якості
Підхарактеристика	Інтерпретація підхарактеристики моделі якості
Назва атрибуту	Інтерпретація атрибута моделі якості
Визначення атрибуту	Короткий опис атрибута моделі якості, зв'язок з потребами
Мета/Мотивація	Призначення атрибута з позиції потреби, яку необхідно реалізувати
Шкала вимірювань	Тип шкали для вимірювання значення атрибута
Процедура визначення, протокол, X	Вказується процедура отримання та тип приналежності атрибута відповідному учаснику проекту.
	Примітки
Тип збору даних та підрахунку	Тип збору атрибута (ручний, автоматизований)
Метрика	Назва метрики
Пріоритетність	Вказується вага атрибута у загальній якості ПС

Однак з метою уточнення та доповнення моделі зовнішньої якості проводимо аналіз атрибутів моделі $\{H_i^s, A_{ik}^s, C_{ik}^s, M_{ij}^s\}, i \in N_{i_s}, K = \overline{1, S_i}$, які можуть ві-

дображатись у відповідні атрибути моделі, що будується. Внаслідок цього отримуємо вимоги зовнішньої якості ПС R_{ext} у вигляді $\{H_i^x, P_{ik}^x, A_{ik}^x, C_{ik}^x, M_{ik}^x\}$, $i \in N_x, K = \overline{1, F_i^x}$. Вимоги в такій формі використовуються на етапі проектування архітектури ПС для розробки тестів, тестуванні, та оцінювання якості робочого продукту цього етапу, що відповідає рекомендаціям стандарту ISO/ IEC 12207.

Побудова моделі внутрішньої якості здійснюється по аналогії до побудови моделі зовнішньої якості, однак при цьому враховуються відображення атрибутів моделі $\{H_i^x, P_{ik}^x, A_{ik}^x, C_{ik}^x, M_{ik}^x\}$, $i \in N_x, K = \overline{1, F_i^x}$ на її структуру. Внаслідок цього отримуємо вимоги внутрішньої якості R_{in} у вигляді $\{H_i^x, P_{ik}^x, A_{ik}^y, C_{ik}^y, M_{ik}^y\}$, $i \in N_x, K = \overline{1, F_i^x}$.

Виходячи з ітерацій, які необхідно виконати для розробки вимог у вигляді структури моделей якості, каскадну процедуру представимо у вигляді, як показано на рис. 3.

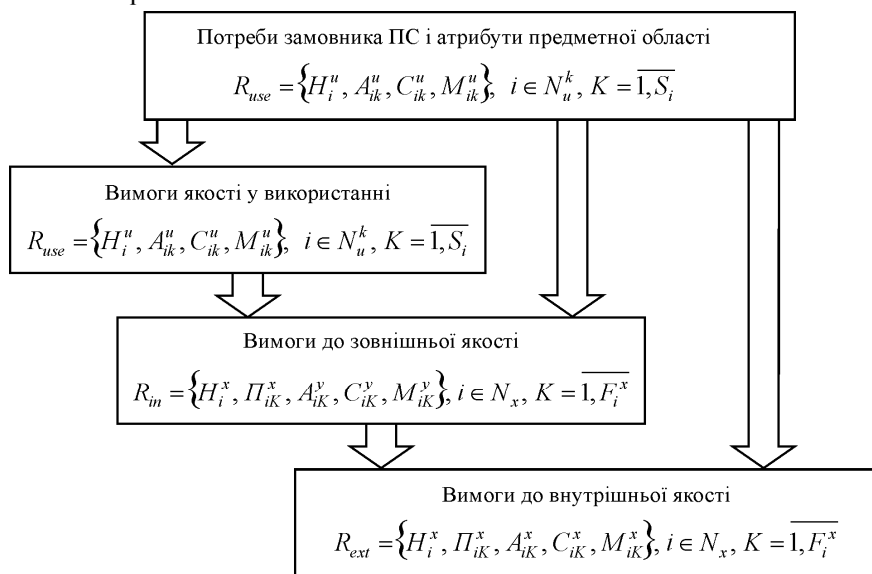


Рис. 3. Каскадна процедура розробки вимог якості

Аналіз процедури проектування вимог, що наведена на рис. 3, показує, що потреби замовника ПС і атрибути предметної області є початковими даними для розробки і структурування вимог за моделями якості.

Висновки:

1. На основі аналізу факторів впливу на якість вимог до програмних систем і критеріїв, яким вони повинні задовольнити, сформульовано задачу забезпечення якості вимог, як фундамент забезпечення якості програмних систем.
2. Розроблено метод забезпечення якості програмних систем, який використовує формалізовані представлення моделей якості стандарту ISO 25010 та метод комунікації вимог якості на стадіях життєвого циклу.

3. Розроблено алгоритми збирання та класифікації вимог якості, а також процедуру каскадного їх проектування, що дає змогу підвищити ефективність і повноту відображення вимог на наступних стадіях життєвого циклу.

Література

1. The Standish group report. CHAOS Manifesto 2013, Think Big, Act Small. [Electronic resource]. – Mode of access <http://www.versionone.com/assets/img/files/ChaosManifesto2013.pdf>.
2. The Standish group report. The CHAOS Manifesto 2012: The Year of the Executive Sponsor. [Electronic resource]. – Mode of access <http://www.versionone.com/assets/img/files / CHAOSManifesto2012.pdf>.
3. Kharchenko A. The method of quality management software / A Kharchenko, I. Galay, V. Yatsyshyn // Perspective Technologies and Methods in MEMS Design (MEMSTECH), 2011 Proceedings of VIIth International Conference on 11-14 May 2011.
4. Hull E. Requirements Engineerig / E. Hull, Ken Jackson, Jeremy Dick. // Springer Science+Business Media. – 2005. – 240 p.
5. IEEE 982.2 Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software.
6. ISO/IEC 25010. Software Engineering – product quality. Quality model4.
7. Sommerville I. Deriving Information Requirements from Responsibility Models / I. Sommerville, R. Lock. T. Storer, J. Dobson // Proc. CAiSE 2009. 21st International Conference on Advanced Information Systems Engineering, Amsterdam, June 2009. – Pp. 515-529.
8. Брауде Е. Технология разработки программного обеспечения / Е. Брауде. – СПб.: Изд-во "Питер", 2004. – 655 с.
9. Харченко О.Г. Методи забезпечення та контролю якості web-застосунків на стадіях життєвого циклу / О.Г. Харченко, В.В. Яцишин, І.О. Боднарчук // Вимірвальна та обчислювальна техніка в технологічних процесах. – Хмельницький. – 2011. – № 1. – С. 21– 27.

Яцишин В.В., Пастух О.А., Боднарчук І.О., Харченко О.Г. Метод забезпечення якості програмних систем на ранніх стадіях життєвого циклу

Рассмотрен метод обеспечения качества требований, что является одним из аспектов управления и обеспечения качества программных систем. На основе теоретико-множественных нотаций проведена формализация требований качества к программному обеспечению, проанализированы критерии качества требований и предложены способы и методы их интеграции в процесс разработки ПО. Разработан алгоритм сбора и классификации требований качества, процедуру построения требований к программным системам на основе моделей качества стандарта ISO 25010, что позволило применить метод коммуникации требований качества на стадиях жизненного цикла.

Ключевые слова: качество требований, качество программного обеспечения, критерии качества требований, управление качеством программного обеспечения.

Yatsyshyn V.V., Pastuh O.A., Bodnarchuk I.O., Kharchenko O.G. The method of quality assurance software systems on the early stages of life cycle

The article is devoted to the method of quality assurance requirements, which is one of the aspects of management and quality assurance software systems. On the set-theoretic notation was conducted formalization of quality requirements to software, analyzed quality criteria requirements and suggested ways and methods for their integration in the process of software development. An algorithm for the collection and classification of quality requirements, the procedure of building requirements for software systems based on models of quality standard ISO 25010 was developed. This algorithm and procedure are allowing us to apply the method of communication quality requirements on the stages of the life cycle.

Keywords: quality requirements, software quality, quality criteria requirements, quality management software.