21. Thaler R.H. Behavioral Economics: Past, Present and Future. American Economic Review. American Economic Association, 2016. Vol. 106(7). P. 1577 – 1600.
22. Tversky A., Kahneman D. Advances in Prospect Theory: Cumulative Representation of Uncertainty. Journal of Risk and Uncertainty. 1992. Vol. 5. P. 297–323.

Статтю подано до редакції 11.10.2020

УДК 004.415.53:004.414.38 DOI 10.33111/mise.100.9

Корзаченко О. В., к.е.н.,
доцент кафедри комп'ютерної математики та інформаційної безпеки,
Міщенко Д. С.,
студентка 4 курсу спеціальності «Системний аналіз»,
ДВНЗ «КНЕУ імені Вадима Гетьмана»

Korzachenko O. V., PhD,
Associate Professor of the Department of
Computer Mathematics and Information Security,
Mishchenko D. S.,
4th year student majoring in «System Analysis»,
SHEI KNEU named after V. Hetman

АНАЛІЗ СТРАТЕГІЙ ПРИЙМАЛЬНОГО КОРИСТУВАЦЬКОГО ТЕСТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

ANALYSIS OF STRATEGIES OF USER ACCEPTANCE TESTING OF INFORMATION SYSTEMS

Анотація. *Стаття присвячена дослідженню стратегій приймального користувацького тестування — остаточне тестування інформаційної системи, яке виконується спільно з кінцевим користувачем або клієнтом для того, щоб перевірити, чи система є повна і коректна для використання у реальному бізнес-середовищі. У роботі визначено, що підґрунтям успішного проведення UAT у проєкті є визначення вимог, розроблення критеріїв прийняття та вибір стратегії тестування. Проаналізовано концепцію та стратегії проведення UAT на основі чорної скриньки, поведінкових сценаріїв і тестування операцій. Досліджено формалізовану модель поведінкового UAT з деревами сценаріїв, розглянуто стратегії приймального тестування операцій і наведено приклад визначення обсягу тест кейсів на основі зваженої критичності. Виявлено основні переваги та недоліки стратегій UAT і зроблено їх порівняльний аналіз на основі таких критеріїв: підхід до тесту, залученість користувачів, усунення багів і критерії прийняття.*

Зроблено висновок, що приймальне користувацьке тестування є невід'ємним і важливим процесом, який потребує великого обсягу часу, адже на даний момент не існує інструменту, за допомогою якого можна зробити всі необхідні та продумані дії під час тестування. Слід зазначити, що вибір стратегії UAT має здійснюватися індивідуально кожною проєктною командою, враховуючи специфіку програмного продукту, який розроблюється. Вважаємо, що незалежно від вибору, до стратегії слід включати оцінювання ризиків, визначення елементів системи, операцій і бізнес-процесів, які найбільш схильні до впливу та наслідків через допущені помилки в них. Це дозволить ефективно розподілити ресурси та приділити особливу увагу елементам інформаційної системи, дефекти в яких матимуть найбільший вплив.
Ключові слова: *приймальне користувацьке тестування, тестування на основі чорної скриньки, поведінкове приймальне тестування на основі сценаріїв, стратегія приймального тестування операцій, формалізована модель, дерево сценаріїв, критична вага, критерій першого прийняття, критерій другого прийняття, інформаційна система, IT-проєкт.*

Abstract*. The article focuses on the study of User Acceptance Testing strategies — the final testing of the information system, which is performed in conjunction with the end user or client to verify that the system is complete and correct for use in a real business environment. It was determined that the basis for the successful implementation of UAT in the project is the definition of requirements, development of eligibility criteria and the choice of testing strategy. The concept and strategies of conducting UAT based on the black box, behavioral scenarios and testing of operations, and their purpose are analyzed. The formalized model of behavioral UAT with scenario trees is investigated, the model of strategy of User Acceptance Testing of operations is considered and the example of definition of volume of test cases on the basis of the weighted criticality is given. The main advantages and disadvantages of UAT strategies are identified and their comparative analysis is made on the basis of the following criteria: test approach, user involvement, bug fixes and acceptance criteria.*
It is concluded that the acceptance of User Acceptance Testing is an integral and important process that requires a lot of time, because at the moment there is no tool with the help of which you can do all the necessary and thoughtful actions during testing. It should be noted that the choice of UAT strategy should be made individually by each project team, taking into account the specifics of the software product to be developed. We believe that regardless of the choice, the strategy should include risk assessment, identification of elements of the system, operations and business processes that are most affected and consequences due to mistakes in them. This will allow for efficient allocation of resources and special attention to the elements of the system in which defects will have the greatest impact.
Keywords: *User Acceptance Testing, Black Box Testing, scenario-based behavioral acceptance testing, operations acceptance testing strategy, formalized model, scenario tree, critical weight, first acceptance criterion, second acceptance criterion, information system, IT project.*

Постановка проблеми. Особливо важливим і критичним етапом проєкту розробки будь-якої інформаційної системи, програми, додатку є фаза перевірки їх прийнятності користувачами, яка сигналізує про те, що реалізовано всі необхідні функціональні можливості та вимоги до програмного продукту. Приймальне ко-

ристувацьке тестування (User Acceptance Testing, UAT) — це перевірка відповідності системи вимогам, які визначені потребами користувачів у їх робочому середовищі та з реальними сценаріями відповідно до специфікацій.

Важливим є факт, що етап UAT надає останню можливість розробникам виявити та усунути наявні проблеми в програмному продукті, перед тим як його запустять у продакшн. За результатами дослідження компанії Standish Group на етапі впровадження системи чистота програмного коду має становити 99,9 %, оскільки виправлення помилок на цьому етапі є доволі трудомістким, дорогим і шкодить репутації як розробника, так і замовника [8].

Незважаючи на важливість у забезпеченні придатності системи для реалізації запланованих цілей і можливу мінімізацію ризиків невдачі проєкту, розробники не завжди приділяють UAT необхідну увагу та оцінюють його дійсну значимість.

Аналіз останніх досліджень і публікацій. Дослідженням сутності та особливостей UAT, їх класифікації, методів, моделей, інструментів присвячені праці закордонних практиків і учених: P. Hsia, J. Samuel, D. Kung, L. Li, C.T. Hsu, C. Chen, Y. Toyoshima, P. Pandit, S. Tahiliani, H. K.N. Leung, P. W.L. Wong, M. E. Khan, F. Khan та інші. Українська наукова спільнота не займається дослідженням UAT як окремої специфічної галузі тестування програмних продуктів, а розглядає його опосередковано, як рутину частину загального процесу тестування.

Метою статті є змістовний аналіз стратегій і методів приймального користувацького тестування інформаційних систем.

Виклад основного матеріалу дослідження. Приймальне користувацьке тестування надає впевненість розробникам, що їх продукт придатний до використання та досяг своєї мети. Залежно від обраної моделі життєвого циклу проєкту UAT здійснюється перед його звершенням або наприкінці кожної фази, ітерації або спринту. У ньому беруть участь не тільки кінцеві користувачі, але й команда з забезпечення якості, розробники, бізнес-аналітики та керівництво вищого рівня.

До основних типів UAT відносять прийомні випродування користувачів (внутрішні альфа-тести та зовнішні бета-тести), експлуатаційні, нормативні та контракті тести [7].

Часто документ про технічні умови та вимоги є єдиним шаблоном для тестування. Всі дії з розробки системи повинні перевірятися на відповідність специфікаціям і вимогам. Необхідно реалі-

зувати серію тестів, які будуть відображати вимоги до бізнес-процесів, операцій і звітності користувача. Ці тести визначать, чи буде продукт правильно підтримувати бізнес.

Слід, зазначити, що за гнучкою методикою (agile) розробки програмного забезпечення історії користувача (user stories) є альтернативою документу вимог, який складається при застосуванні каскадної моделі (waterfall). Історія користувача — це вимога до будь-якої функціональності або функції, яка записана у кілька рядків [9].

Також необхідно сформувати критерії прийняття, які визначають ступінь виконання user story або вимоги (рис. 1). Найпоширенішими є історії орієнтовані на правила (у вигляді списку) та сценарії. Сценарії активно використовуються agile командами, оскільки вони допомагають переходити до різних вимог, описуючи окремі варіанти використання.

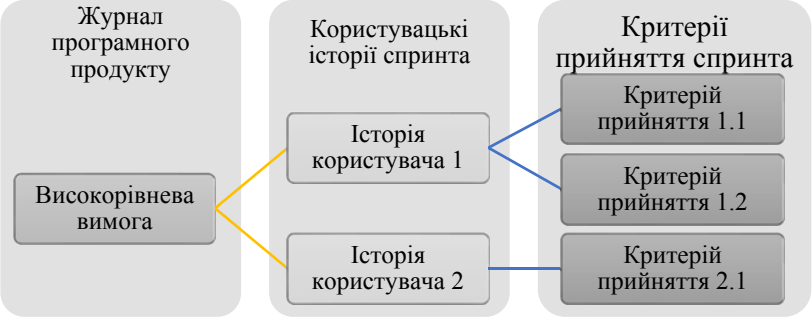

Рис. 1. Взаємозв'язок user stories та критеріїв прийняття у спринті IT-проєкту

Тобто підґрунтям успішного проведення UAT у проєкті є визначення вимог, розроблення критеріїв прийняття та вибір стратегії тестування, серед яких доцільно виділити такі: (1) поведінкове приймальне тестування на основі сценаріїв; (2) тестування на основі чорної скриньки; (3) тестування на основі операцій.

Поведінкове тестування на основі сценаріїв є частиною методології BDD (Behavior Driven Development), за якою першочергово розробляються історії користувача або сценарії, що в подальшому використовуються для документування функцій системи та запуску тестів прийняття [1]. Сценарій — це конкретний приклад використання системи, що складається із впорядкованої послідовності подій, які відображають вимоги користувача або замовника

[3]. Шаблони сценаріїв, що складаються з однакової впорядкованої послідовності типів подій і виконання заданої вимоги до системи, утворюють схеми сценарію.

У межах цієї стратегії на особливу увагу заслуговує спроба формалізації моделі з використанням дерев рішень і теорії автоматів [2, 4]. Модель складається з трьох типів субмоделей, які є набором схем сценаріїв: (1) подання користувача (a user view), (2) користувацький інтерфейс системи (an external system interface) та (3) зовнішнє подання системи (an external system view).

Побудова формалізованої моделі поведінкового UAT складається з кількох кроків: (1) виділення та специфікація сценаріїв користувачів і користувацького інтерфейсу, побудова дерев сценаріїв; (2) кожен користувацький інтерфейс визначається як скінченний автомат (FSM, finite-state machine), кожен з яких згодом об'єднується у складений скінченний автомат (CFSM), що й утворює зовнішнє подання системи; (3) перевірка тестової моделі, де всі згенеровані FSM і CFSM досліджуються з точки зору детермінованості, послідовності, правильності та повноти [4, с. 58].

Дерево сценаріїв позначене як $T(V) = (N, E, L)$ для користувача V складається з скінченних наборів вузлів N, ребер E та міток ребер L. Зокрема, кореневий вузол дерева сценаріїв, позначений <S>, є сприйнятим користувачем початковим станом системи і називається початковим вузлом. Усі кінцеві вузли також позначаються як <S>. Набір вузлів складається з набору станів, які сприймаються користувачем. Для кожного ребра $e \in E$, між будь-якими двома вузлами є мітка $l \in L$, пов'язана з ним, що і є типом події. Позначка $l \in L$ ребра $e \in E$ між вузлами $N1, N2 \in N$ показує, що стан системи змінився з $N1$ на $N2$ через появу типу події l [2, с. 295].

Запропонована модель надає комплексний підхід для виявлення та тестування взаємодіючих сценаріїв, що передбачає побудову матриці для набору пов'язаних FSM, а потім генерування відповідних сценаріїв.

Метою такої формалізації є використання FSM для моделювання схеми сценарію, щоб тестовий сценарій можна було аналізувати, перевіряти та генерувати систематичним способом перед проведенням UAT.

Перевагою розглянутої моделі у межах поведінкової стратегії UAT є надання можливості користувачам відповідно до вимог аналітиків визначати та генерувати тести прийняття за допомогою системного підходу. Проте реалізація моделі є доволі трудомісткою та вимагає технічних знань — *недоліки*.

Стратегія UAT на основі чорної скриньки призначена для вивчення причинно-наслідкових зв'язків між взаємодією користувача з програмним продуктом і результатом (виходом) (рис. 1). Зазвичай, UAT на основі чорної скриньки аналізує лише основні аспекти системи і не має відношення до її внутрішньої логічної структури, а тестувальник не має доступу до вихідного програмного коду [5]. Користувачам-тестувальникам пояснюють, для чого призначений продукт, але як саме він працює вони вивчають самостійно.

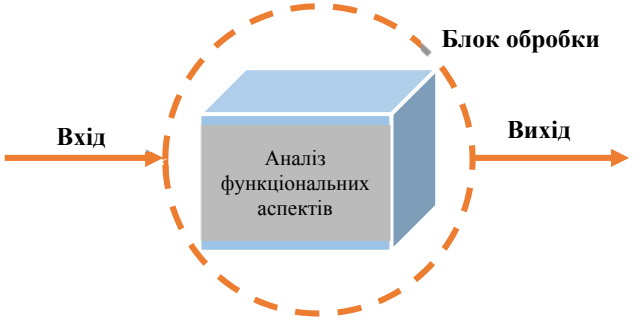

Рис. 2. Тестування за стратегією чорної скриньки

Для приймального користувацького тестування на основі чорної скриньки можна використовувати сукупність методів, які відображено на рис. 3.

1. Еквівалентне розбиття зменшує кількість тестових випадків, оскільки поділяє вхідні дані програмного блоку на розділи даних, з яких можна отримати тестові приклади.
2. Аналіз граничних значень – це метод тестування продукту на крайніх (граничних) значеннях вхідних даних. Він також включає тести, що перевіряють поведінку системи на вхідних даних, що виходять за допустимий діапазон значень.
3. Нечіткі методи тестування застосовуються для пошуку помилок реалізації при введенні деформованих / напівдеформованих даних в автоматизованому або напівавтоматизованому режимі.
4. Граф причинно-наслідкових зв'язків – це метод, за якого тестування починається зі створення графу та встановлення зв'язку між наслідком і його причинами, який описаний логічними операторами: тотожність, заперечення, логічне АБО та логічне І.
5. Тестування ортогонального масиву можна застосовувати для виявлення проблем, при яких вхідний масив даних порівняно невеликий, але занадто великий, щоб забезпечити вичерпне тестування.

6. Попарне тестування передбачає замість перевірки всіх можливих дискретних комбінацій значень усіх параметрів перевіряються тільки комбінації значень кожної пари параметрів.

7. Діаграма переходу стану аналізує поведінку системи при різних умовах входу, метод корисний для тестування стану автомату, перевірки навігації графічним інтерфейсом користувача.

Рис. 3. Методи чорної скриньки для тестування

Тестування на основі чорної скриньки здійснюється відповідно до специфікацій функціональних або зовнішніх вимог системи. Для вибору мінімального набору тестів використовують функціональну матрицю тестів [6].

Стратегія UAT на основі чорної скриньки має низку незаперечних *переваг*: легко сприймається тестувальником; розробка тест кейсів швидка; зменшується можливість проведення неперевірених тестів; тестувальники та розробники не залежать один від одного. *Недоліками* є відсутність чітких специфікацій, що ускладнює розробку тестових прикладів; фактично виконується лише вибрана кількість тестових сценаріїв, як результат, покриття обмежене; критерії прийнятності не чітко визначені.

Стратегія тестування на основі операцій визначає ефективність реалізації процесів в ІС виходячи із використання їх різними групами користувачів. Кожен клас користувачів формує операційний профіль (ОП), що складається із сукупності процесів або операцій і частоти їх ініціації (виклику, появи) (табл. 1).

Таблиця 1

ВИЗНАЧЕННЯ ЙМОВІРНОСТІ ПРОЦЕСІВ В ОП

Код ОП	Ймовірність ініціації (використання) процесів			
	Процес 1	Процес 2	Процес 3	Процес 4
01	0,25	0,25	0,35	0,35
02	0,55	0,15	0,15	0,35
03	0,15	0,45	0,30	0,30

За даної стратегії обсяг тестування процесів ґрунтується на відносній частоті їх використання (ймовірність виникнення). Так, наприклад, операціям, які були ініційовані у системі нещодавно слід призначити більшу кількість випробувань [6].

Також на кількість тестів впливає критичність операцій, яка визначає ступінь тяжкості ефекту за умови виходу системи з ладу. Так критичність кожного ОП визначається окремо. Наприклад, банківська програма може мати 3 ОП користувачів: касир, супервізор і менеджер. ОП касира є найкритичнішим, керівника менш критичним, а менеджера ще найменш критичним з точки зору важливості результатів програми.

Обсяг тестування для кожного ОП пропорційний його критичній вазі:

$$C_i' = \frac{C_i}{\sum_{i=1}^{k} C_i P_i} P_i \ ,$$

$$Q_i = C_i' Q \ ,$$

де C_i' — вага критичності i-го ОП, $i = \overline{1,k}$; k — загальна кількість ОП; C_i — ступінь критичності i-го ОП; P_i — ймовірність ініціації i-го ОП, Q — загальна кількість тестувань, Q_i — кількість тестувань для i-го ОП.

Приклад визначення обсягу тест кейсів на основі зваженої критичності наведено у табл. 2.

Критерієм прийняття у стратегії тестування операцій є такий — «критичних багів не виявлено, а надійність ІС на прийнятному рівні». Виділяють критерії першого та другого прийняття.

1. Критерій першого прийняття

Нехай T — набір тест кейсів, $t_i \in T$ з відображенням $s_i P s_i'$; s_i — показник входу у тест t_i; s_i' — показник виходу застосування s_i до програми P.

Вимога відсутності критичної несправності зображується так: $\neg s_i \in C$, де C — набір неправильних критичних вихідних значень, визначених користувачем.

2. Критерій другого прийняття

(а) Надійність кожного класу ОП є прийнятною за умови: $R_i \geq \Re_i, \forall i$, де R_i — розрахункова надійність i-го ОП, $\Re_i$ — прийнятна надійність i-го ОП.

(б) Надійність усієї системи є прийнятною за умови: $R_0 \geq \Re_0$, де R_0 – передбачувана розрахункова надійність інформаційної системи, $\Re_0$ – прийнятна надійність системи.

Таблиця 2

ПРИКЛАД РОЗРАХУНКУ ОБСЯГУ ТЕСТУВАННЯ

Код ОП, i	Ступінь критичності, C_i	Ймовірність ініціації, P_i	C_iP_i	Вага критичності, C_i'	Кількість тест кейсів
1	1	0,25	0,25	0,097	5
2	4	0,35	1,40	0,538	27
3	3	0,15	0,45	0,173	8
4	2	0,25	0,50	0,192	10
Всього		**100**	**2,6**	**1**	**50**

Після проведення розрахунків виноситься рішення щодо прийняття системи, усунення багів та приведення у відповідність до вимог. Замовник може відмовитися від прийняття через вагомі дефекти як у всій системі, так і в окремих ОП (табл. 3).

Таблиця 3

ПРИКЛАД ВИЗНАЧЕННЯ КРИТЕРІЇВ ПРИЙНЯТТЯ ІС

Варіант	$R_i \geq \Re_i, \forall i$	$R_0 \geq \Re_0$	Рішення щодо прийняття
1	Помилок не виявлено		Прийняти
2	Так	Так	Попереднє приймання
3	Ні	Так	Умовне приймання
4	Так	Ні	Перероблення
5	Ні	Ні	Відмова

Перевагами застосування стратегії UAT на основі операцій є те, що вибір і формування тест кейсів здійснюється для кожного

операційного профілю, а критерії прийнятності чітко визначені. Проте стратегія не позбавлена *недоліків*, її реалізація вимагає проведення додаткового аналізу перед початком тестування, оскільки необхідно виділити різні операційні профілі, а визначення обсягу тестування займає додатковий час.

Порівняльний аналіз стратегій UAT, в основу яких покладено розглянуті моделі, наведено у табл. 4.

Таблиця 4

ПОРІВНЯННЯ СТРАТЕГІЙ UAT

Критерії	Поведінкове UAT на основі сценаріїв	UAT на основі чорної скриньки	UAT на основі операцій
Підхід до тесту	Аналіз сценаріїв, метод скінченних автоматів	Ґрунтується на зовнішньому функціоналі	Створюються тест кейси для кожного ОП користувача. Тести ґрунтуються на ймовірності ініціації операцій
Основа тесту	Специфікація системи	Вимоги користувачів, процедурні інструкції	Вимоги користувачів
Критерії прийняття	Не визначено	Не знайдено головних проблем	Не лишилось критичних багів. Прийнятна надійність
Залученість користувачів	Обмежена	Середня	Середня
Усунення багів	Не визначено	Список помилок	Чітко визначено процедуру
Сильні сторони	Враховує поведінку та містить сценарії	Операційні процедури включені у тестування	Критерії прийняття чітко визначені
Слабкі сторони	Трудомісткий процес, вимагає технічних знань	Відсутність критеріїв прийняття	Попередній аналіз

З огляду на критерії порівняння (табл. 4) опис стратегії UAT на основі операцій є найповнішим. Розроблені моделі для визначення пріоритетних ОП, обсягу тестування кожного ОП та критерії прийняття ОП і системи можуть з легкістю використовуватися на практиці. Безумовно, реалізація такої стратегії значно підвищить ефективність тестування та забезпечить отримання якісного програмного продукту.

Висновки. Багато в чому UAT є одним з найбільш критичних і ризикованих етапів будь-якого ІТ-проєкту. Воно поєднує в собі всі елементи бізнес-процесів у єдину дієву та вимірювану діяльність, яка не тільки гарантує, що користувачі отримають програмний продукт, який задовольнить їхні реальні вимоги, але й забезпечить економічні вигоди бізнесу та успішність ключовим зацікавленим сторонам проєкту.

Слід зазначити, що наразі не існує певного уніфікованого підходу до вибору стратегії UAT або єдиної методології тестування у межах обраної стратегії. Такі рішення приймаються індивідуально, зважаючи на: (1) специфіку кожного окремого проєкту розробки інформаційної системи; (2) рівень кваліфікації тестувальників та команди із забезпечення якості програмного продукту; (3) рівень мотивації зацікавлених сторін; (4) загальний рівень культури проєктного менеджменту.

Вважаємо за доцільне при виборі стратегії UAT та розробці методології тестування використовувати методи системного аналізу та включати у модель оцінювання ризиків, визначення елементів системи, операцій і бізнес-процесів, які найсхильніші до їх впливу, та кількісного визначення наслідків через допущені помилки. Це дозволить ефективно розподілити ресурси та зосередитися на елементах системи, дефекти в яких матимуть найбільший вплив на результат ІТ-проєкту.

Бібліографічні посилання

1. Acceptance Tests In Practice — Behavior Driven Development. URL: https://blog.rapid7.com/2015/02/22/acceptance-tests-in-practice-behavior-driven-development/.
2. Hsia P., Gao J., Samuel J., Kung D., Toyoshima Y. and Chen C. Behavior-based acceptance testing of software systems: a formal scenario approach. *Proceedings of the 18th Annual International Computer Software and Applications Conference (COMPSAC)*. 1994. URL: https://info.computer.org/csdl/pds/api/csdl/proceedings/download-article/12OmNwHQB2o/pdf.
3. Hsia P., Kung D. Software Requirements and Acceptance Testing. Annuals of Software Engineering. 1997. Vol. 3. P. 291–317.
4. Hsia P., Samuel J., Kung D., Li L., Hsu C.T., Chen C., Toyoshima Y. A Usage-Model Based Approach to Test Therac-25. *IFAC Proceedings Volumes*. 1995. Vol. 28. Issue 25. P. 55–63.
5. Khan M.E., Khan F. A Comparative Study of White Box, Black Box and Grey Box Testing Techniques. International Journal of Advanced Computer Science and Applications. 2012. Vol. 3. № 6. P. 12–15.
6. Leung Hareton K.N., Wong Peter W.L. A Study of User Acceptance Tests. Software Quality Journal. 1997. Vol. 6. P. 137-149.

7. Pandit P., Tahiliani S. AgileUAT: A Framework for User Acceptance Testing based on User Stories and Acceptance Criteria. *International Journal of Computer Applications*. 2018. Vol. 120. № 10. P. 16–21.
8. The Standish Group International. Rule of ten. URL: https://www.standishgroup.com/sample_research_files/RuleTen.pdf.
9. What Is User Story And Acceptance Criteria (Examples). URL: https://www.softwaretestinghelp.com/user-story-acceptance-criteria/.
10. Пышкин Е.В. Проблемы автоматизации приемочного тестирования программного обеспечения при использовании подхода «разработка, управляемая описанием поведения». *Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика, телекоммуникации и управление.* 2012. №6 (162). URL: https://cyberleninka.ru/article/n/problemy-avtomatizatsii-priemochnogo-testirovaniya-programmnogo-obespecheniya-pri-ispolzovanii-podhoda-razrabotka-upravlyaemaya.

Статтю подано до редакції 05.11.2020

УДК:519.218 DOI 10.33111/mise.100.10

Круглова Н. В., к.ф.-м.н., доцент,
кафедра математичного аналізу та теорії ймовірностей,
Диховичний О. О., к.ф.-м.н., доцент,
кафедра математичного аналізу та теорії ймовірностей
Дем'яненко О. О., к.ф.-м.н., доцент,
кафедра математичного аналізу та теорії ймовірностей,
НТУУ «КПІ ім. Ігоря Сікорського»

Kruglova N. V., PhD (physical and mathematical sciences),
Associated professor of Mathematical analysis and
Probability Theory Department
Dykhovychnyi O. O., PhD (physical and mathematical sciences),
Associated professor of Mathematical analysis and
Probability Theory Department
Demianenko O. O., PhD (physical and mathematical sciences),
Associated professor of Mathematical analysis and
Probability Theory Department
NTUU «Igor Sicorsky Kyiv Polytechnic Institute»

ПРО ОЦІНКИ ТОЧНОСТІ МОДЕЛЮВАННЯ ЗВУЖЕННЯ БРОУНІВСЬКОГО ЛИСТА НА ЧАСТИНУ КОЛА В ПРОСТОРІ H_{SP}

ON ESTIMATES OF MODELING ACCURACY FOR BROWNIAN SHEET`S RESTRICTION ON THE PART OF A CIRCLE IN SPACE H_{SP}