ЖУРНАЛ ІНЖЕНЕРНИХ НАУК
JOURNAL OF ENGINEERING SCIENCES
ЖУРНАЛ ИНЖЕНЕРНЫХ НАУК
Web site: http://jes.sumdu.edu.ua/
Том 3, № 2 (2016)

УДК 004.415.532

Побудова імітаційної моделі процесу тестування програмного забезпечення в пакеті Simulink

В. М. Дубовой[1)], І. В. Пилипенко[2)]

[1), 2)] *Вінницький національний технічний університет, Хмельницьке шосе, 95, 21000, м. Вінниця, Україна*

Article info:
Paper received: 25 February 2016
The final version of the paper received: 28 July 2016
Paper accepted online: 02 November 2016

Correspondent Author's Address:
[1)] v.m.dubovoy@gmail.com
[2)] innapilipenko17@gmail.com

У роботі вирішується актуальне завдання визначення моделі розроблення рішень тестування програмного забезпечення, що дозволяє прогнозувати як процес загалом, так і його певні етапи. Метою роботи є прогнозування часу тестування програмного продукту та підвищення його якості. Аналіз процесу тестування програмного забезпечення показав, що цей процес можна віднести до розгалуженo-циклічних технологічних процесів, оскільки виконується він циклічно із прийняттям рішень на контрольних операціях. Спираючись на попередні праці авторів, до процесу тестування програмного забезпечення було застосовано методику на основі марковської моделі. Запропонована методика дозволяє виконувати прогнозування для кожного окремого модуля програмного продукту, що призводить до більш якісного прийняття рішень на кожній контрольованій операції всього підпроцесу. Для реалізації та перевірки результатів цієї методики у роботі розроблено імітаційну модель процесу тестування програмного продукту. Результати дослідження набули практичної реалізації у вигляді впровадження на підприємстві.

Ключові слова: розгалужено-циклічний технологічний процес, контрольна операція, прийняття рішення, витрати, оцінювання ризику, прогнозування циклічності, критерій достатності, якість продукту.

1. ВСТУП

Відповідно до поставлених завдань тестування використовуються різні практики імплементації тестування, що в основному впливає на якість самих тестів. Тому актуальним є визначення певної моделі розроблення рішень тестування, яка б дозволяла прогнозувати як процес у цілому, так і його певні етапи [1].

Існує безліч моделей тестування програмного продукту. Цією темою займається багато сучасних дослідників, таких як В. С. Яковина, М. М. Сенів, Я. М. Чабанюк [2]; J. P. Durand, O. Gaudoin [3]; S.Yamada, M. Ohpa, S. Osaki [4] та ін.

Аналіз процесу тестування програмного забезпечення показав, що цей процес можна віднести до розгалужено-циклічних технологічних процесів, оскільки виконується циклічно з прийняттям рішень на контрольних операціях [5].

Важливою складовою кожної моделі тестування програмного продукту є критерій достатності процесу тестування, який дозволяє керівникам проектів приймати обґрунтовані рішення про завершення цього етапу розроблення. Принципова характеристика процесу тестування програмного забезпечення полягає у циклічності процесу, а саме в тому, що в програмі за період виконання кожного циклу тестування виявляються і виправляються помилки, які не були знайдені раніше. У цей час у переважній більшості ІТ-компаній такі показники мають формальний характер і затрати на тестування намагаються зменшити, що призводить до зниження якості продукту. Необхідно спрогнозувати затрати на процес тестування, враховуючи його циклічну повторюваність [6].

В [6] авторами запропонована модель управління циклічними розгалуженими технологічними процесами, що ґрунтується на невизначених циклічних графах і неоднорідних марковських ланцюгах.

Застосована модель [7] розгалужено-циклічного технологічного процесу до процесу тестування програмного забезпечення (ПЗ) дає можливість прогнозувати час на виконання процесу. Складність застосування визначається тим, що існують стани, які неможливо одночасно віднести до бажаних чи небажаних. Кожний модуль, що тестується, має свої характеристики і потребує окремого прогнозування. Запропонована методика дозволяє виконувати прогнозування для кожного окремого модуля програмного продукту, що приводить для більш якісного прийняття рішень на кожній контрольованій операції всього підпроцесу.

Для перевірки застосування розробленої марковської моделі до процесу тестування програмного забезпечення було використано імітаційне моделювання. Застосування імітаційних моделей дає безліч

переваг порівняно із виконанням експериментів над реальною системою і використанням інших методів [8].

Метою роботи є прогнозування часу тестування програмного продукту та підвищення його якості. Відповідно в рамках цієї статті завданням є розроблення імітаційної моделі процесу тестування як розгалужено-циклічного технологічного процесу, яка б дозволила формально описати процес прийняття рішень під час тестування, а також дозволила б спрогнозувати затрати на процес тестування.

2. ОСНОВНА ЧАСТИНА

Для розроблення моделі в середовищі Simulink побудуємо концептуальну модель процесу тестування як циклічного технологічного процесу.

На рис. 1 показано концептуальну модель процесу тестування.

Кожний програмний модуль характеризується такими параметрами:

– розміром;

– вимогами до фунціональності.

Розмір програмного модуля впливає на кількість тестових випадків, які необхідно провести інженеру щодо забезпечення якості (що впливає на коефіцієнт покриття програмного продукту).

Вимоги до функціональності визначають межі показника якості μ, що відповідає за готовність програмного модуля до впровадження та експлуатації.

Тестувальник розробляє набір тестових випадків для перевірки програмного модуля. Такий набір характеризується розміром N (кількістю тестів), та часом на їх виконання t.

Після виконання усіх тестів тестувальник одержує кількість виявлених дефектів у програмному модулі. Кількість дефектів та їх складність визначають показник якості програмного продукту. Фактично, це ймовірність того, що продукт може бути переданий до експлуатації.

На основі показника якості та показника покриття тестувальник приймає рішення:

1) програмний продут відправити на дороблення;

2) програмний продукт вводити в експлуатацію.

Якщо програмний продукт було відправлено на дороблення, він виправляється програмістом і наступним етапом знову надходить на перевірку тестувальнику. Тестувальник перевіряє модуль за новим набором тестів із перевіркою попередніх, які не були виконані. Такий процес тестування відбувається доти, поки продукт не буде відповідати показнику якості та не буде повністю покритий тестами.

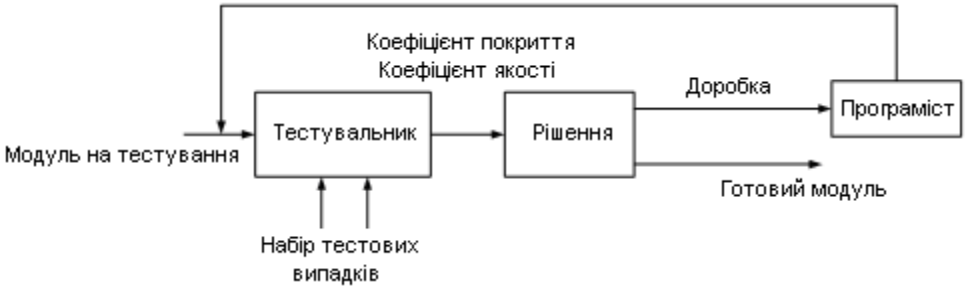


Рисунок 1 – Концептуальна модель процесу тестування за технологією Scrum

Розглянемо блок Тестувальник як сервер, на вхід якого подаються тести з постійною швидкістю із певним періодом. Період визначається відношенням кількості тестів за годину.

Для реалізації блока Рішення необхідно здійснити обробку результатів із попереднього блока та розрахувати показник якості і показник покриття програмного продукту. Розраховані показники необхідно порівняти із заданими. Для цього, знайдемо затримку в часі після виконання тестового циклу та кількість тестів, які не встигли виконатися.

Коефіцієнт покриття знайдемо як відношення виконаних тестів до всіх тестів, які необхідно було виконати. Для розрахунку коефіцієнта покриття було створено підсистему Coverage (рис. 2) із використанням блоків для розрахунку:

– кількості всіх тестів, які можна виконати за відповідний цикл;

– кількості успішно виконаних тестів;

– коефіцієнта покриття.

Коефіцієнт якості знайдемо як відношення кількості знайдених дефектів до кількості успішно виконаних тестів на одному циклі. Для розрахунку коефіцієнта якості створено підсистему Quality (рис. 2) із використанням блоків для розрахунку:

– кількості дефектів;

– коефіцієнта якості.

Після випробовування моделі одержані показники покриття та якості порівнюємо з відомими межами показника якості та покриття тестами і приймаємо рішення щодо подальших дій: уведення в експлуатацію чи повернення на доопрацювання та повторне тестування. Операцію порівняння будемо виконувати на етапі оброблення результатів моделювання.

Циклічність процесу тестування реалізуємо за допомогою розгортання циклу у послідовність однакових підсистем. Тобто циклічну частину будемо виконувати послідовно до того моменту, поки показники не будуть задовольняти задані умови. Таке перетворення ґрунтується на невизначених циклічних графах та показане в [6].

Отже, здійснимо необхідні зв'язки та покажемо комплексну модель процесу тестування з урахуванням підсистем розрахунку показників покриття та якості (рис. 2).

Блок Delay time (рис. 3) показує час очікування кожного тесту в черзі перед тестувальником. Горизонтальна вісь відображає час виконання тестів, а вісь Y – коефіцієнт очікування.

Покажемо приклади коефіцієнтів покриття та якості модуля В на 4 циклах (час моделювання – 360 хв). Аналогічні випробовування здійсненно для модулів А та Б.

Було проведено моделювання процесу тестування ПЗ за технологією Scrum.

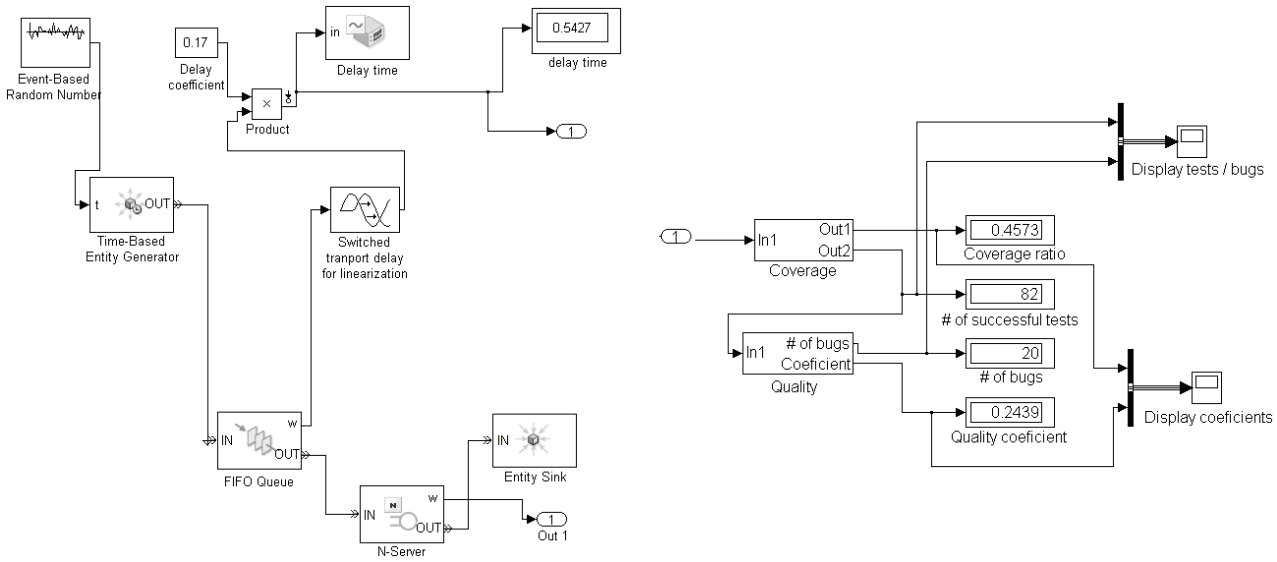
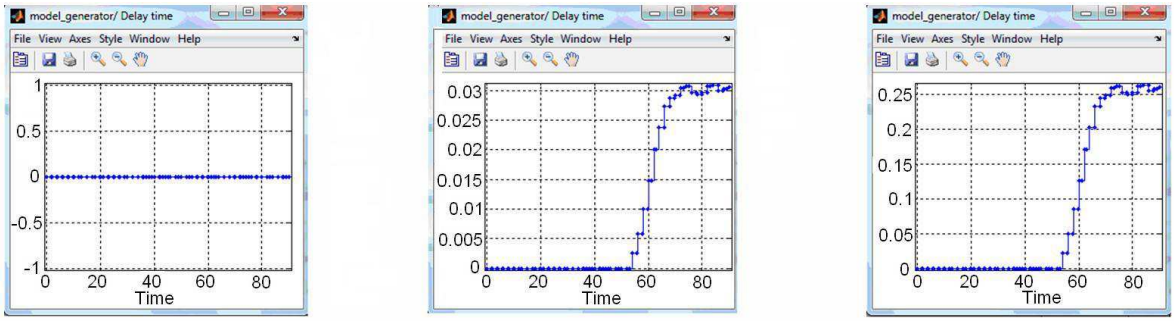


Рисунок 2 – Випробовування моделі процесу тестування для модуля В: коефіцієнт покриття та коефіцієнт якості



а б в

Рисунок 3 – Коефіцієнт очікування тесту на виконання для модулів А (а), Б (б), В (в)

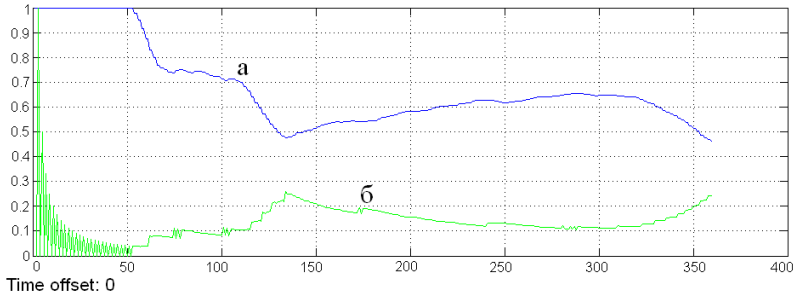


Рисунок 4 – Випробовування моделі процесу тестування для модуля В: коефіцієнт покриття (а) та коефіцієнт якості (б)

Результати моделювання показали такі результати (табл. 1):

1. Для модуля А середній час затримки за один цикл дорівнює 0 – усі тести пройдуть успішно без затримки.

2. Для модуля Б середній час затримки за один цикл дорівнює 0,03059, або 3 % із 90 хв.

3. Для модуля В середній час затримки за один цикл дорівнює 0,26, або 26 % із 90 хв.

Зі збільшенням кількості дефектів (що відбувається при виконанні багатьох циклів) коефіцієнт покриття зменшується, оскільки час тестування витрачається на перевірку вже знайдених дефектів. Проте коефіцієнт якості збільшується, оскільки продукт протестований за дефектами.

Для прогнозування часу виконання тестування продукту необхідно виконати стільки циклів тестування, за яких можна було б стверджувати, що продукт готовий до випуску (готова нова версія). Для цього існує певна ймовірність помилок, з якими продукт можна впроваджувати у використання. При цьому для кожного модуля така ймовірність буде різною, оскільки нова версія продукту спирається на реалізацію або покращання певної функції та допоміжних їй функцій.

У нашому прикладі, вважаємо, що [6]:

Модуль А – головний модуль, який вимагає якісного тестування, при цьому розроблявся кваліфікованою командою.

Модуль Б – суміжний модуль, який покращує роботу системи та буде удосконалюватися в майбутніх версіях.

Таблиця 1 – Результати моделювання

Цикл	Кількість дефектів	Кількість успішно виконаних тестів	Коефіцієнт покриття	Ймовірність помилок
Модуль А				
1	0	45	1	0
2	0	90	1	0
3	0	135	1	0
4	0	180	1	0
5	0	225	1	0
Модуль Б				
1	1	43	0,97	0,10
2	1	85	0,94	0,15
3	2	129	0,95	0,02
4	3	168	0,93	0,02
5	5	204	0,90	0,01
Модуль В				
1	3	33	0,74	0,70
2	9	49	0,54	0,90
3	10	86	0,63	0,35
4	20	82	0,46	0,24
5	35	51	0,23	0,68

Модуль В – новий модуль, розробка якого не має високого приорітету і буде удосконалюватися в наступних версіях.

Отже, для модуля А ймовірність помилок повинна бути мінімальною, прирівняємо її до 0. Ймовірність помилок модуля Б повинна становити не більше 0.4, та ймовірність помилок модуля В повинна становити не менше 0.6.

Отже, аналізуючи таблицю ймовірностей помилки та відомих значень, можна зробити висновок, що найбільш оптимальним буде виконання 3 циклів тестування. Відомо, що 1 цикл тестування для одного модуля займає 90 хв. Отже, на кожний модуль необхідно використати по 270 хв.

Аналогічні розрахунки були проведені за допомогою неоднорідної марковської моделі в [6]. Результати прогнозування співпадають з розрахунковими даними, проведеними за допомогою марковської моделі та імітаційної моделі.

Для оцінки ефективності розробленого програмного забезпечення було проведено ряд експериментів на тестовій вибірці з даними проходження тестових сценаріїв для 50 різних модулів.

Перевірка адекватності модельного комплексу була здійснена за допомогою коефіцієнта розбіжності Тейла [9], який дорівнює нулю за відсутності похибок прогнозу і не має верхньої межі. Для дослідження на адекватність прогнозу були обрані ключові показники моделі (табл. 2). Кількість випробувань одного модуля становить 50 ітерацій. В табл. 2 наведені результати розрахунку коефіцієнта Тейла для одного модуля при різних кількостях випробувань.

Дані табл. 2 показують, що зі збільшенням кількості випробувань, модель є адекватнішою. Згідно з проведеним дослідженням, модель має достатньо високу адекватність: коефіцієнт U знаходиться в межах від 0,05 до 0,18 при кількості випробувань – 50 ітерацій. Також необхідно зазначити, що модель має меншу адекватність при розрахунку показника коефіцієнта покриття. Це пояснюється взаємним впливом знайдених дефектів та проведених тестів. Як зазначалося вище, зі збільшенням кількості дефектів коефіцієнт покриття зменшується, проте коефіцієнт якості збільшується, оскільки продукт протестований за дефектами.

Таблиця 2 – Дослідження адекватності моделі за допомогою коефіцієнта Тейла

Показник моделі	Кількість випробувань n	Коефіцієнт Тейла U
Коефіцієнт покриття	25	0,25
	50	0,18
Коефіцієнт якості	25	0,10
	50	0,06
Коефіцієнт очікування	25	0,09
	50	0,08
Кількість циклів	25	0,09
	50	0,09
Проведені тести	25	0,08
	50	0,07
Знайдені дефекти	25	0,05
	50	0,05
Прогнозований час на виконання тестування	25	0,14
	50	0,13

Надалі результати дослідження практично реалізовані у вигляді програмного забезпечення, впровадженого у ТОВ НВП «Спільна справа» (м. Вінниця) в 2015 році на основі розроблених моделей, алгоритмів та методів. На підприємстві «Спільна справа» для керування процесом тестування програмних продуктів використовується система Squash TM.

Для оцінювання переваг за рахунок застосування розробленого програмного забезпечення інформаційної технології прийняття рішень при керуванні розгалужено-циклічними технологічними процесами було проведено порівняльний аналіз результатів, одержаних за допомогою системи Squash TM без застосування розробленого програмного забезпечення та результатів, одержаних із застосуванням розробленої інформаційної технології. Порівняльний аналіз показано в таблиці 3.

Із таблиці бачимо, що при застосуванні розробленого програмного забезпечення вдається досягти вищого коефіцієнта якості при меншій кількості циклів, а отже, і зменшити витрати на тестування. Ці параметри напряму впливають на загальну ефективність системи. Результати, на основі експериментів з 50 ітерацій для кожного модуля, дозволяють стверджувати, що застосування розробленого програмного забезпечення істотно впливає на характеристики процесу тестування програмного забезпечення. Таким чином, загальний час на надмірне тестування зменшився на 18 %, а показник якості програмного продукту збільшився на 11,5 %.

Таблиця 3 – Порівняльний аналіз результатів експериментів

Результати тестування		Кількість виконаних циклів тестування	Коефіцієнт покриття	Коефіцієнт якості
Система Squash TM з інтеграцією Squash AM	Модуль А	4	1	1
	Модуль Б		0,98	0,869
	Модуль В		0,66	0,575
ІТ ПР РЦТП	Модуль А	3	1	1
	Модуль Б		0,95	0,982
	Модуль В		0,63	0,650

3. ВИСНОВКИ

Розроблено імітаційну модель процесу тестування як розгалужено-циклічного технологічного процесу. За допомогою розробленої моделі здійснено прогнозування часу тестування програмного продукту. Результати дослідження набули практичної реалізації у вигляді програмного забезпечення, яке впроваджене у ТОВ НВП «Спільна справа» (м. Вінниця) в 2015 році. При застосуванні розробленого програмного забезпечення вдається досягти вищого коефіцієнта якості при меншій кількості циклів, а отже, і зменшити витрати на тестування. Результати дозволяють стверджувати, що загальний час на надмірне тестування зменшився на 18 %, а показник якості програмного продукту збільшився на 11,5 %.

Н

Creating a simulation model of software testing using Simulink package

V. M. Dubovoi[1)], I. V. Pylypenko[2)]

[1), 2)] *Vinnytsia National Technical University, 95, Khmelnytsky Highway, 21000, Vinnytsia, Ukraine*

The determination of the solution model of software testing that allows prediction both the whole process and its specific stages is actual for IT-industry. The article focuses on solving this problem. The aim of the article is prediction the time and improvement the quality of software testing. The analysis of the software testing process shows that it can be attributed to the branched cyclic technological processes because it is cyclical with decision-making on control operations. The investigation uses authors' previous works andsoftware testing process method based on Markov model. The proposed method enables execution the prediction for each software module, which leads to better decision-making of each controlled suboperation of all processes. Simulink simulation model shows implementation and verification of results of proposed technique. Results of the research have practically implemented in the IT-industry.

Keywords: branched cyclic process, control operation, decision-making, costs, risk assessment, cyclicity prediction, sufficiency criterion, product quality.

Построение имитационной модели процесса тестирования программного обеспечения в пакете Simulink

В. М. Дубовой[1)], И. В. Пилипенко[2)]

[1), 2)] *Винницкий национальный технический университет, Хмельницкое шоссе, 95, 21000, г. Винница, Украина*

В работе решается актуальная задача определения модели разработки решений тестирования программного обеспечения, позволяющая прогнозировать как процесс в целом, так и его определенные этапы. Целью работы является прогнозирование времени тестирования программного продукта и повышение его качества. Анализ процесса тестирования программного обеспечения показал, что данный процесс можно отнести к разветвленно-циклическим технологическим процессам, поскольку выполняется циклически с принятием решений на контрольных операциях. Опираясь на предыдущие работы авторов, к процессу тестирования программного обеспечения была применена методика на основе марковской модели. Предложенная методика позволяет выполнять прогнозирование для каждого отдельного модуля программного продукта, что приводит к более качественному принятию решений на каждой контролируемой операции всего подпроцесса. Для реализации и проверки результатов данной методики в работе разработана имитационная модель процесса тестирования программного продукта. Результаты исследования получили практическую реализацию в виде внедрения на предприятии.

Ключевые слова: разветвлено-циклический технологический процесс, контрольная операция, принятие решения, затраты, оценка риска, прогнозирование цикличности, критерий достаточности, качество продукта.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ремінний О. А. Модель процесу автоматизованого тестування користувацьких інтерфейсів в умовах багатопродуктових компаній / О. А. Ремінний // Вимірювальна та обчислювальна техніка в технологічних процесах. – 2013. – № 4. – С. 106–113.
2. Критерій достатності процесу тестування програмного забезпечення / В. Яковина, М. Сенів, Я. Чабанюк та ін. // Комп'ютерні науки та інформаційні технології : [збірник наук. праць] / відп. ред. Ю. М. Рашкевич. – Львів : Видавництво НУ «Львівська політехніка», 2010. – С. 346–358.
3. Durand J. P. Software reliapility modelling and prediction with hidden Markov chains / J. P. Durand, O. Gaudoin // Statistical Modelling (5). – 2005. – P. 75–93.
4. Yamada S. S-shaped reliapility growth modelling for software error detection / S. Yamada, M. Ohpa, S. Osaki // IEEE Transactionson Reliapility. – 1983. – Vol. R-32, No. 5. – P. 475–478.
5. Дубовой В. М. Прогнозування доцільної кількості повторень циклічного технологічного процесу / В. М. Дубовой, І. В. Пилипенко // Вісник ВПІ. – № 1. – Вінниця : УНІВЕРСУМ Вінниця, 2015. – С. 86–91.
6. Дубовой В. М. Моделювання процесу тестування програмного забезпечення як розгалужено-циклічного технологічного процесу / В. М. Дубовой, І. В. Пилипенко // Автоматизація технологічних і бізнес-процесів. – 2015. – № 24 – Ч. 7. – С. 55–64.
7. Дубовой В. М. Застосування марковської моделі для аналізу впливу циклічності на управління розгалуженим технологічним процесом [Електронний ресурс] / В. М. Дубовой, І. В. Пилипенко, Р. С. Стець // Наукові праці ВНТУ. – 2014. – № 4. – С. 21–27. – Режим доступу : http://praci.vntu.edu.ua/article/view/3827/5583.
8. Комп'ютерне моделювання систем та процесів. Методи обчислень. Частина 1 : навчальний посібник / Р. Н. Квєтний, О. Р. Бойко, О. Ю. Софина, О. М. Шушура ; за заг. ред. Р. Н. Квєтного. – Вінниця : ВНТУ, 2012. – 193 с.
9. Нейлор Т. Машинные имитационные эксперименты с моделями экономических систем / Т. Нейлор ; [пер. с англ.]. – Москва : Мир, 1975. – 502 с.

REFERENCES

1. Reminnyi, O. A. (2013). Model protsesu avtomatyzovanoho testuvannia korystuvatskyh interfeisiv v umovah bahatoproduktovyh kompanii [Model of automated testing process of user interfaces in conditions of many-production companies]. *Vymiriuvalna ta obchysliuvalna tehnika v tehnolohichnyh protsesah – Measuring and computing in technological processes*, 4, 106–113 [in Ukrainian].
2. Yakovyna, V., Seniv, M., Chabanyuk, D., Himka, U. (2010). The criterion of adequacy of software testing [Kryterii dostatnosti protsesy testuvannia prohramnoho zabezpechennia]. *Lvivska politehnika*, 346–358 [in Ukrainian].
3. Durand, J. P., Gaudoin, O. (2005) Software reliapility modelling and prediction with hidden Markov chains. *Statistical Modelling* (5), 75–93.
4. Yamada S., Ohpa M., Osaki S. (1983). S-shaped reliapility growth modelling for software error detection. *IEEE Transactionson Reliapility. Vol. R-32, No. 5*, 475–478.
5. Dubovoi, V. M., Pylypenko, I. V. (2015). Prohnozuvannia dotsilnoi kilkosti povtoren tsyklichnoho tehnolohichnoho protsesy [Prediction of expedient quantity of repetitions if the cyclic technological process]. Visnyk VPI – *News of VPI, 1, 86–91* [in Ukrainian].
6. Dubovoi, V. M., Pylypenko, I. V. (2015). Modeliuvania protsesy testyvannia prohramnoho zabezpechennia yak rozhaluaheno-tsyklichnoho tehnolohichnoho protsesy [Simulation of software testing as branched-cyclic technological process]. Avtomatyzatsia technolohichnyh i biznez-protsesiv – *Automation of technological i business processes,* Vol. 7, 24, 4, 55–54 [in Ukrainian].
7. Dubovoi, V. M., Pylypenko, I. V., Stets, R. S. (2014). Zastosuvania markovskoi modeli dlia analizu vplylu tsyklichnosti na upravlinnia rozhalushenym tehnolohichnym protsesom [Using of Markov model to analyze the impact of cyclicity on management of branched technological process]. *Naukovi pratsi VNTU – Scientific proceedings of VNTU*, 4, 21–27. Retrieved from http://praci.vntu.edu.ua/article/view/3827/5583.
8. Kvetnyi, R. N., Bohach, I. V., Boiko, O. R., Sofina, O. Yu., Shushura, O. M. (2012). *Kompiuterne modeliuvannia system ta protsesiv. Metody obchyslen [Computer simulation of systems and processes. Methods of computation]*. R. N. Kvetnyi (Ed). (Vol. 1). Vinnytsia: VNTU [in Ukrainian].
9. Neylor T. Computer simulation experiments with models of economic systems (1975). Willey, New York.