



ТЕХНІЧНІ ЗАСОБИ ТЕХНОЛОГІЇ В СИСТЕМАХ УПРАВЛІННЯ

УДК 519.24:681.515

МОДЕЛЮВАННЯ ПРОЦЕСУ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЯК РОЗГАЛУЖЕНО-ЦИКЛІЧНОГО ТЕХНОЛОГІЧНОГО ПРОЦЕСУ

Simulation of software testing as branched-cyclic technological process

Дубовой В.М.¹, Пилипенко І.В.¹ (Dubovoy V.M., Pylypenko I.V.)¹Вінницький національний технічний університет, м. Вінниця,¹E-mail: innapilipenko17@gmail.com

Copyright © 2014 by author and the journal "Automation technological and business - processes".

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>DOI: [10.15673/2312-3125.24/2015.56333](https://doi.org/10.15673/2312-3125.24/2015.56333)

Анотація

В статті розглянуто процес тестування програмного забезпечення як розгалужено-циклічний технологічний процес. 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. Авторами запропоновано побудувати модель процесу тестування, щоб спрогнозувати затрати на процес тестування, враховуючи його циклічну повторюваність. Побудовано найпростішу схему процесу тестування ПЗ у вигляді графа. Застосовано алгоритм перетворення циклічного графа процесу тестування на ациклічний. Побудовано марковську модель РТП для процесу тестування ПЗ, враховуючи перетворення циклічного графа на еквівалентний йому ациклічний. За основу взято неоднорідну марковську модель. Застосовано розроблену модель на реальних даних. Розглянуто технологію управління проектами Scrum, яка використовується для гнучкої розробки ПЗ. Прогнозування часу здійснено за допомогою необхідної кількості циклів до досягнення заданої ймовірності відсутності дефектів в програмному продукті. Для визначення ймовірності станів використано відомі результати проходження кожного з модулів на основі 50 ітерацій. Використано визначення бажаних та небажаних станів, в яких може бути операція після виконання одного циклу. Розроблена модель дала змогу спрогнозувати час виконання процесу тестування ПЗ для трьох різних модулів. Запропонований підхід дозволяє здійснювати прогнозування процесу тестування для кожного окремого модуля програмного продукту, що дозволяє прийняти обґрунтовані рішення на кожній контрольній операції всього підпроцесу.

Annotation

In the paper the process of software testing is described as branched-cyclical technological process. The principal characteristic of the software testing process is cyclical process, namely that the program for the period of execution of each test cycle detected and corrected errors that were not discovered earlier. The authors proposed to build a model of the test to predict the cost of the testing process given its cyclical recurrence. We constructed a simple circuit of software testing process as a graph. We applied the algorithm of conversion the cycle graph of testing process to acyclic. We constructed Markov model for software testing process, including conversion cycle graph at the equivalent acyclic. The basis is taken heterogeneous Markov model. Authors applied the designed model on real data. We described Scrum technology of project management, that used for software development. The prediction is done through number of cycles required to achieve a given probability of absence of defects in software. To determine the probability of the states we used the results known passage of each module based on 50 iterations. We used preferred and undesirable conditions which can be operation after one cycle.

**ТЕХНІЧНІ ЗАСОБИ ТЕХНОЛОГІЇ В СИСТЕМАХ УПРАВЛІННЯ**

The model made it possible to predict the performance of the process of software testing for three different modules. The proposed approach allows the prediction of the testing process for each software module that allows you to make informed decisions on each control operations in whole subprocess.

Ключові слова

Розгалужено-циклічний технологічний процес, тестування програмного забезпечення, ациклічний граф, неоднорідна марковська модель, ймовірність стану.

Вступ

З невідомим розвитком та поширенням інформаційних технологій у всіх сферах людської життєдіяльності з'являються нові та складніші завдання перед розробниками програмного забезпечення. Програмні продукти стають багатокомпонентними і вимагають спеціалізованого підходу для їх розроблення та перевірки. У зв'язку з цим підвищуються вимоги до надійності програм, виникає потреба у скороченні витрат на тестування та у прогнозуванні надійності розроблюваного програмного забезпечення (ПЗ) [1].

Відомо, що для прогнозування надійності програмного забезпечення широко використовують тип моделей, які описують характеристики ПЗ на основі кількості виявлених помилок у програмному продукті [2]. Тому, для оцінки та прогнозування надійності ПЗ необхідно досліджувати, вдосконалювати та аналізувати моделі та методи тестування.

Актуальність

Існує безліч моделей тестування програмного продукту. Даною темою займається багато сучасних дослідників, таких як Яковина В.С., Сенів М.М., Чабанюк Я.М. [3]; Durand J.P., Gaudoin O. [4]; Yamada S., Ohra M., Osaki S. [5] та ін.

Важливою складовою кожної моделі тестування програмного продукту є критерій достатності процесу тестування, який дозволяє керівникам проектів приймати обґрунтовані рішення про завершення даного етапу розробки. 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.

В даний час у переважній більшості ІТ компаній такі показники носять формальний характер та затрати на тестування намагаються зменшити, що призводить до зниження якості продукту. Необхідно будувати модель процесу тестування, щоб спрогнозувати затрати на процес тестування, враховуючи його циклічну повторюваність. Саме це і обґрунтовує актуальність проблеми моделювання процесу тестування як розгалужено-циклічного технологічного процесу.

Мета роботи

Прогнозування часу тестування програмного продукту як розгалужено-циклічного технологічного процесу.

Задачі дослідження

1. Аналіз процесу тестування програмного забезпечення як розгалужено-циклічного технологічного процесу.
2. Побудова моделі процесу тестування як розгалужено-циклічного технологічного процесу.
3. Застосування розробленої моделі для прогнозування часу тестування програмного продукту.

Процес тестування як розгалужено-циклічний технологічний процес

Найпростіша схема процесу тестування зображена у вигляді графа на рис. 1.

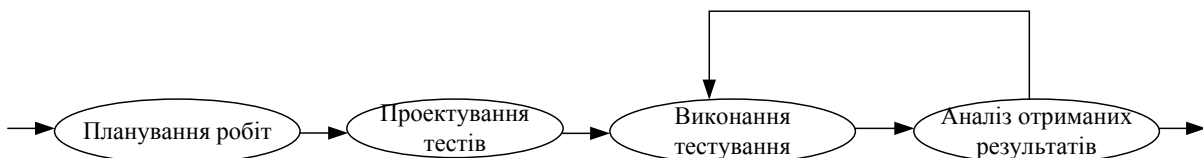


Рис. 1. Процес тестування ПЗ: циклічний граф процесу

Тестування - це одна з технік контролю якості, що включає:

- Планування робіт (TestManagement);
- Проектування тестів (TestDesign);
- Виконання тестування (TestExecution);
- Аналіз отриманих результатів (TestAnalysis) [7].

Ці чотири етапи є операціями процесу тестування ПЗ. Кожна операція містить стани, які варіюються в



ТЕХНІЧНІ ЗАСОБИ ТЕХНОЛОГІЇ В СИСТЕМАХ УПРАВЛІННЯ

залежності від вхідних параметрів та параметрів виконання попередньої операції. Зокрема, «Планування робіт (TestManagement)» містить наступні параметри, які впливають на стан операції:

- інформація про структуру ПЗ або системи в документації («біла скринька»);
- тестові набори даних для перевірки правильності роботи компонентів і системи в цілому без знання їх структури («чорна скринька»);
- граничні значення, таблиці прийняття рішень, потоки даних, статистика відмов і ін.;
- блок-схеми побудови програм і набори тестів для покриття системи цими тестами та ін.

Підпроцес «Виконання тестування» - «Аналіз отриманих результатів» виконується циклічно доки виявлення дефектів стане мінімальним. При цьому, після першого циклу виконання підпроцесу на операцію «Виконання тестування» буде впливати результат попереднього циклу, тобто результат операції «Аналізу». При цьому вхідні дані для виконання наступного циклу будуть варіюватися, оскільки вони залежать від результатів виконання процесу на попередньому циклі.

В залежності від того, яким чином буде знайдено дефект, стан операції знаходження дефекту буде варіюватися:

- ми дізнаємося (або вже знаємо) очікуваний результат;
- ми дізнаємося (або вже знаємо) фактичний результат;

Стан операції також буде визначатися тестовими даними, які використовуються для перевірки роботи системи і складаються різними способами: генератором тестових даних, проектною групою на основі документів або наявних файлів, користувачем з специфікації вимог та ін.

Побудова моделі процесу тестування як розгалужено-циклічного технологічного процесу

В [6] авторами запропонована модель управління циклічними розгалуженими технологічними процесами, яка ґрунтується на невизначених циклічних графах і неоднорідних марковських ланцюгах. Побудуємо таку модель для процесу тестування ПЗ.

В [7] було запропоновано алгоритм перетворення циклічного графа процесу тестування на ациклічний, що зображено на рис. 2.

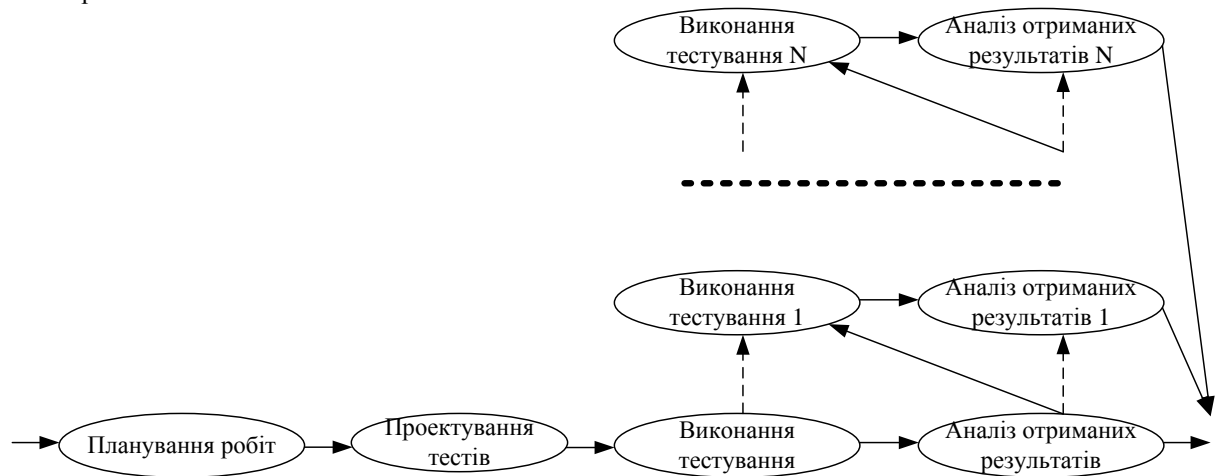


Рис. 2. Процес тестування ПЗ:перетворення циклічного графа в ациклічну форму

Застосуємо марковську модель РТП до процесу тестування ПЗ, враховуючи перетворення циклічного графа на еквівалентний йому ациклічний. За основу візьмемо неоднорідну марковську модель [6]:

$$\tilde{p}_v(X_j) = \sum_{i=1}^m \sum_{h=1}^{v+1} \left\{ \sum_{r=0}^{(h-1)} \dots \sum_{r=0}^{h-1} c_1 c_2^{h-1} \prod_{r=0}^{h-1} \tilde{p}_{v-h}(X_i) \right\}$$

де $c_v^{00}(X_i, X_j) = c_{v-1}^{00}(X_i, X_j) = c_1$ - коефіцієнт впливу v -го підпроцесу з i -го в j -й стан без урахування впливу інших підпроцесів; $c_v^{v-1,h}(X_i, X_j) = c_{v-1}^{v-2,h}(X_i, X_j) = c_2$ - коефіцієнт впливу v -го підпроцесу з i -го в j -й стан з урахуванням попереднього $v-2$ -го підпроцесу з h -м станом; X_j - номер стану операції; $\tilde{p}_v(X_j)$ - вірогідність стану; m - кількість станів; v - кількість під процесів; h - номер стану; p_{v-h} - ймовірність $v-2$ -го підпроцесу з h -м станом.

Проілюструємо методику отримання параметрів марковської моделі на рис. 3.



ТЕХНІЧНІ ЗАСОБИ ТЕХНОЛОГІЇ В СИСТЕМАХ УПРАВЛІННЯ

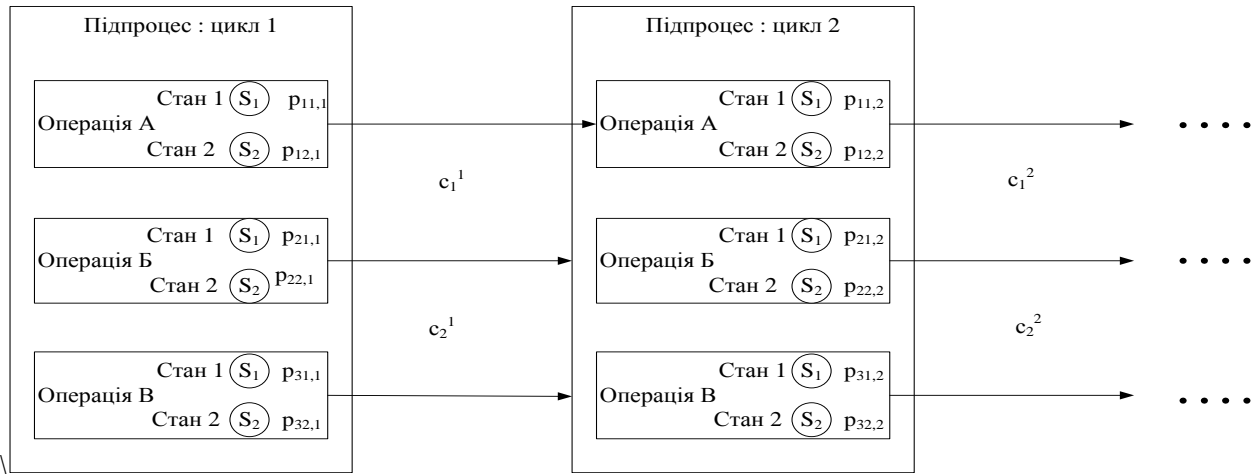


Рис. 3. Визначення параметрів марковської моделі

Наприклад, щоб отримати ймовірності станів операцій на 3 циклі виконання підпроцесу, з відомими аналогічними ймовірностями на перших двох циклах необхідно врахувати коефіцієнти впливу 2-го підпроцесу з 1-го в 2-ий стан без врахування впливу інших підпроцесів та з урахуванням попереднього 1-го підпроцесу з його відомим станом.

Застосування розробленої моделі для прогнозування часу тестування програмного продукту

Розглянемо процес тестування ПЗ на практиці. Будемо розглядати технологію управління проектами Scrum, яка використовується для гнучкої розробки ПЗ. Особливістю цієї технології є акцент на тестуванні програмного продукту в кінці кожної ітерації [8].

Для прогнозування часу тестування ПЗ візьмемо реальні дані з відомої ІТ-компанії у м. Вінниця. Для полегшення опису даних оберемо для прикладу одного інженера з контролю якості програмного продукту (тестувальника).

1. Робочий день тестувальника: всього на тестування витрачається приблизно 4,5 год в день з 8-ми годинним робочим днем. Інший час – це мітинги (переговори), пошта, чати, обід (перерви) тощо.

2. Ціль тестування – якнайбільше провести тестів (тестових випадків). В середньому на проходження одного автоматизованого тестового випадку необхідно витратити 2 хв. За одну сесію можна виконати таких 45 тестів.

3. Якщо тестувальник знайшов помилку (дефект) – потрібно її записати в систему управління проектом. В середньому на опис дефекту тестувальнику необхідно 10 хвилин.

Оберемо 3 модуля для тестування. Розробники модуля А – команда з високими професійними якостями; розробники модуля Б – команда з середніми професійними якостями; розробники модуля В – команда з низькими професійними якостями. Складемо таблицю витраченого часу на виконання тестування кожного з модулів (таблиці 1-2).

Таблиця 1 – День 1-й

| Модуль | Час на оформлення дефекту (тести, які знайшли дефекти) | Час на виконання тесту (тести, які не знайшли дефекти) | Всього тестів |
|--------|--|--|---------------|
| А | 0 хвилин (немає дефектів) | 90 хвилин (45 тестів) | 45 тестів |
| Б | 10 хвилин (1 дефект) | 80 хвилин (40 тестів) | 41 тест |
| В | 80 хвилин (8 дефектів) | 10 хвилин (5 тестів) | 13 тестів |

Після проведення тестування за представленим сценарієм можна зробити наступні висновки:

1. Модуль А відмінно протестований по всіх тестах.
2. Модуль В відмінно протестований по дефектах.
3. Модуль В відстає в 3 рази.

Таблиця 2 – День 2-й

**ТЕХНІЧНІ ЗАСОБИ ТЕХНОЛОГІЇ В СИСТЕМАХ УПРАВЛІННЯ**

| Модуль | Час на перевірку виправлень | Час на оформлення дефекту (тести, які знайшли дефекти) | Час на виконання тесту (тести, які не знайшли дефекти) | Нових тестів | Всього тестів за 2 дні |
|--------|-----------------------------|--|--|--------------|------------------------|
| А | 0 хвилин | 0 хвилин (немає дефектів) | 90 хвилин (45 тестів) | 45 | 90 |
| Б | 6 хвилин (1 дефект) | 10 хвилин (1 дефект) | 74 хвилин (37 тестів) | 38 | 79 |
| В | 48 хвилин (8 дефектів) | 40 хвилин (4 дефекта) | 2 хвилини (1 тестів) | 5 | 18 |

1. Модуль В за 2 дні відстає в 5 разів.
2. Модуль Б відстає на 10 %.

Отже:

- 1) знаходження дефектів сьогодні потребує перевірку їх завтра, що затримує тестування;
- 2) якість продукту великою мірою впливає на швидкість тестування.

Застосуємо запропоновану модель для прогнозування часу тестування ПЗ.

Прогнозувати час будемо по необхідній кількості циклів до досягнення заданої ймовірності відсутності дефектів.

На стан поточної ітерації впливає стан попередньої ітерації – залежить від того чи було знайдено дефекти і їхня кількість.

Отже, маємо 2 цикли одного підпроцесу з трьома операціями (А, Б, В). За один цикл (1 ітерація) кожна з операцій може мати 2 стани:

- 1) протестований по тестах – мінімум дефектів S_1 .
- 2) протестований по дефектах – максимум дефектів S_2 .

Для визначення ймовірності станів використаємо відомі результати проходження кожного з модулів на основі 50 ітерацій. Так, наприклад маємо вибірку з даними проходження тестових сценаріїв для модуля А за 50 ітерацій (таблиця 3).

Таблиця 3 - вибірка з даними проходження тестових сценаріїв для модуля А

| Ітерація | К-сть тестів, що пройшли успішно (без виявлення дефектів) | К-сть тестів, під час проходження яких було виявлено дефекти | Ітерація | К-сть тестів, що пройшли успішно (без виявлення дефектів) | К-сть тестів, під час проходження яких було виявлено дефекти |
|----------|---|--|----------|---|--|
| 1 | 44 | 7 | 26 | 41 | 4 |
| 2 | 42 | 3 | 27 | 43 | 2 |
| 3 | 41 | 4 | 28 | 43 | 2 |
| 4 | 45 | 0 | 29 | 41 | 4 |
| 5 | 40 | 5 | 30 | 42 | 3 |
| 6 | 41 | 4 | 31 | 43 | 2 |
| 7 | 43 | 2 | 32 | 40 | 5 |
| 8 | 43 | 3 | 33 | 39 | 6 |
| 9 | 41 | 5 | 34 | 41 | 4 |
| 10 | 42 | 3 | 35 | 45 | 0 |
| 11 | 43 | 2 | 36 | 37 | 8 |
| 12 | 40 | 5 | 37 | 40 | 5 |
| 13 | 39 | 6 | 38 | 43 | 2 |



| | | | | | |
|----|----|---|----|----|---|
| 14 | 41 | 4 | 39 | 40 | 5 |
| 15 | 45 | 0 | 40 | 41 | 4 |
| 16 | 39 | 8 | 41 | 44 | 1 |
| 17 | 40 | 5 | 42 | 42 | 3 |
| 18 | 43 | 2 | 43 | 41 | 4 |
| 19 | 40 | 5 | 44 | 45 | 0 |
| 20 | 41 | 4 | 45 | 40 | 5 |
| 21 | 44 | 1 | 46 | 41 | 4 |
| 22 | 42 | 3 | 47 | 43 | 2 |
| 23 | 41 | 4 | 48 | 43 | 1 |
| 24 | 48 | 0 | 49 | 41 | 4 |
| 25 | 40 | 5 | 50 | 42 | 3 |

На основі експериментальних даних визначимо коефіцієнт кореляції Спірмана.

Для автоматизації розрахунку було написано додаток на мові програмування JavaScript. Робоче вікно з результатами обчислення показано на рис. 4.

| № | X | Y | |
|----|----|---|-----|
| 1 | 1 | 7 | Del |
| 2 | 2 | 3 | Del |
| 3 | 3 | 4 | Del |
| 4 | 4 | 0 | Del |
| 5 | 5 | 5 | Del |
| 6 | 6 | 4 | Del |
| 7 | 7 | 2 | Del |
| 8 | 8 | 3 | Del |
| 9 | 9 | 5 | Del |
| 10 | 10 | 3 | Del |
| 11 | 11 | 2 | Del |
| 12 | 12 | 5 | Del |
| 13 | 13 | 6 | Del |
| 14 | 14 | 4 | Del |
| 15 | 15 | 0 | Del |

| | | | |
|-----------|----|----------|-----|
| 16 | 16 | 8 | Del |
| 17 | 17 | 5 | Del |
| 18 | 18 | 2 | Del |
| 19 | 19 | 5 | Del |
| 20 | 20 | 4 | Del |
| 21 | 21 | 1 | Del |
| 22 | 22 | 3 | Del |
| 23 | 23 | 4 | Del |
| 24 | 24 | 0 | Del |
| 25 | 25 | 5 | Del |
| Добавить | | Очистить | |
| Вычислить | | | |

$r=0.17, p>0,05$

Рис. 4. Приклад робочого вікна обробки експериментальних даних

На основі експериментальних даних отримуємо статистичні оцінки. За допомогою критерія Стьюдента перевіряємо, чи можемо отримані статистичні оцінки (відносні частоти) розглядати як ймовірності.

Вплив попереднього циклу на поточний визначається тим, що після проходження одного циклу операція В почала відставати в 3 рази, після проходження другого циклу – в 5 разів, тобто було виконано лише 18 тестів з 90 необхідних. Коефіцієнти впливу будемо визначати за допомогою статистичної оцінки. Так, наприклад, для модуля складемо табл. 4:

Таблиця 4 – Розрахунок коефіцієнту впливу

| Номер експерименту, в якому на попередньому циклі дефект не виявлено | Номер експерименту, в якому на попередньому циклі дефект виявлено | | | | Загальна кількість виявлених дефектів /загальна кількість тестів |
|--|---|-----|-----|-----|--|
| | 1 | ... | 50 | | |
| 1 | + | | 0 | | 3/50 |
| ... | ... | ... | ... | ... | ... |
| 50 | ... | ... | + | | 10/50 |
| Середній коефіцієнт впливу | | | | | 0,33 |



ТЕХНІЧНІ ЗАСОБИ ТЕХНОЛОГІЇ В СИСТЕМАХ УПРАВЛІННЯ

Аналогічні таблиці було складено для модулів Б та В.

Складемо таблицю ймовірностей двох циклів процесу тестування – табл. 5.

Таблиця 5 – Ймовірності станів операцій після 2 циклів тестування

| Цикл | Операція | Стан | Вірогідність стану | Вплив |
|----------------|----------|----------------|--------------------------|---|
| P ₁ | А | S ₁ | p _{11,1} = 1 | c ₁ ¹ = 1 c ₂ ¹ = 0,33 |
| | | S ₂ | p _{12,1} = 0 | |
| | Б | S ₁ | p _{21,1} = 0,88 | |
| | | S ₂ | p _{22,1} = 0,12 | |
| | В | S ₁ | p _{31,1} = 0,28 | |
| | | S ₂ | p _{32,1} = 0,72 | |
| P ₂ | А | S ₁ | p _{11,2} = 1 | c ₁ ² = 1 c ₂ ² = 0,2 |
| | | S ₂ | p _{12,2} = 0 | |
| | Б | S ₁ | p _{21,2} = 0,84 | |
| | | S ₂ | p _{22,2} = 0,16 | |
| | В | S ₁ | p _{31,2} = 0,11 | |
| | | S ₂ | p _{32,2} = 0,89 | |

Розрахуємо ймовірності того, що операція А на 3-му підпроцесі (цикл P₃) буде знаходитися в першому S₁ стані.

$$\begin{aligned}\tilde{p}_{3,A}(X_1) &= c_1^1 \cdot c_2^1 \cdot p_{11,1} \cdot p_{11,2} + c_1^2 \cdot c_2^2 \cdot p_{11,1} \cdot p_{11,2}, \\ \tilde{p}_{3,A}(X_1) &= 1 \cdot 0,33 \cdot 1 \cdot 1 + 1 \cdot 0,2 \cdot 1 \cdot 1 = 0,53.\end{aligned}$$

Вірогідність того що в S₂ стані:

$$\begin{aligned}\tilde{p}_{3,A}(X_2) &= c_1^1 \cdot c_2^1 \cdot p_{12,1} \cdot p_{12,2} + c_1^2 \cdot c_2^2 \cdot p_{12,1} \cdot p_{12,2}; \\ \tilde{p}_{3,A}(X_2) &= 0.\end{aligned}$$

Аналогічно розрахуємо ймовірності для операцій Б та В:

$$\tilde{p}_{3,B}(X_1) = 0,39; \quad \tilde{p}_{3,B}(X_2) = 0,01; \quad \tilde{p}_{3,B}(X_1) = 0,016; \quad \tilde{p}_{3,B}(X_2) = 0,34.$$

В [7] подано визначення бажаних та небажаних станів для операції. Узагальнивши, підсумуємо ймовірності станів після виконання 3 циклів.

Таблиця 6 – Ймовірності станів операцій після виконання 3 циклів

| Операція | Стан | Ймовірність станів | | | Визначення стану |
|----------|-----------------|--------------------|--------|--------|------------------|
| | | Цикл 1 | Цикл 2 | Цикл 3 | |
| А | S _{1А} | 1.00 | 1.00 | 0.53 | бажаний стан |
| | S _{2А} | 0.00 | 0.00 | 0 | небажаний стан |
| Б | S _{1Б} | 0.88 | 0.84 | 0.39 | бажаний стан |
| | S _{2Б} | 0.12 | 0.16 | 0.01 | небажаний стан |
| В | S _{1В} | 0.28 | 0.11 | 0.016 | бажаний стан |
| | S _{2В} | 0.72 | 0.89 | 0.34 | небажаний стан |



ТЕХНІЧНІ ЗАСОБИ ТЕХНОЛОГІЇ В СИСТЕМАХ УПРАВЛІННЯ

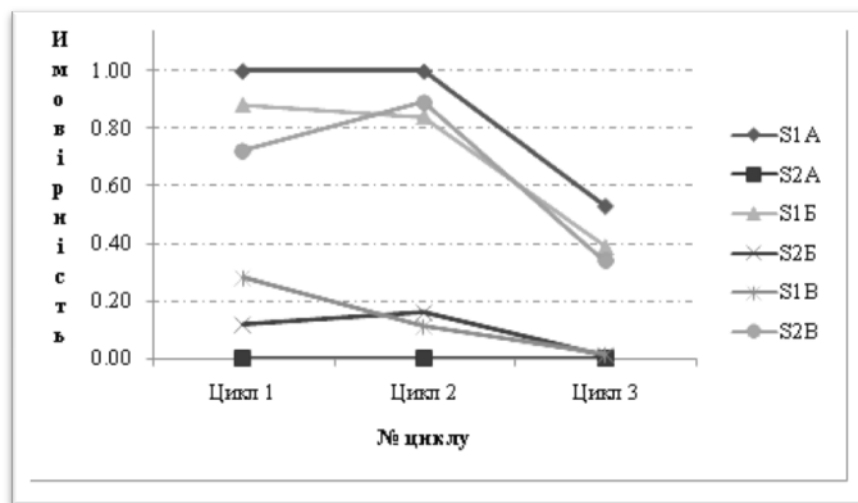


Рис. 5. Ймовірності станів операцій після виконання 3 циклів

Даний випадок має стани які одночасно є бажаними та небажаними, наприклад

- 1) модуль А відмінно протестований по всіх тестах, але не по дефектах
- 2) модуль В відмінно протестований по дефектах, але не по тестах.

Тому на графіку ми бачимо зниження вірогідності обох станів для кожної операції після збільшення циклів тестування. Це пояснюється тим, що при збільшенні циклів тестування тестувальник проведе більше тестів, при чому для модуля А (модуль без дефектів) – це зайве тестування (проходження одних і тих же тестів). Для модуля Б та В – це знаходження більшої кількості дефектів, перевірка їх та продовження проходження всіх тестів.

Тому можна зробити висновки, що:

1) час на виконання тестування модуля А буде мінімальним, оскільки небажаний стан є незмінним та дорівнює 0.

2) час на виконання тестування модуля В буде найбільшим, оскільки необхідно витратити час на перевірку дефектів, продовжити проводити нові тести і вносити записи про нові знайдені дефекти. Зі збільшенням циклів тестування небажаний стан зменшується, що говорить про те, що необхідно провести більше циклів для стабілізації якості модуля В, а отже, і зросте час на тестування.

Для прогнозування часу виконання тестування продукту необхідно виконати стільки циклів тестування, при 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 опирається на реалізацію або покращення певної функції та допоміжних їй функцій.

У нашому прикладі, вважається, що:

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 версія

Модуль В – новий модуль, розробка якого не має високого пріоритету 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, нанесемо на графік (рис. 6) задані ймовірності та спрогнозуємо необхідну кількість циклів.



ТЕХНІЧНІ ЗАСОБИ ТЕХНОЛОГІЇ В СИСТЕМАХ УПРАВЛІННЯ

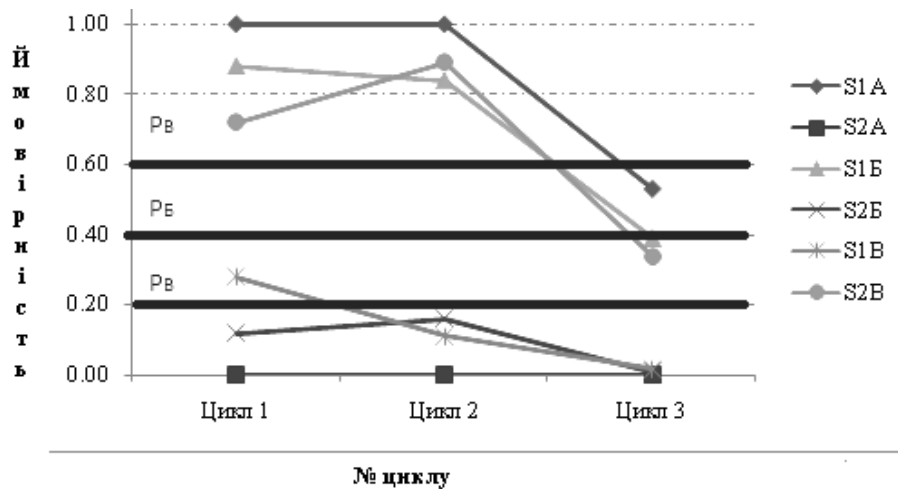


Рис. 6. Задані ймовірності та прогнозування необхідної кількості циклів

Отже, аналізуючи таблицю ймовірностей бажаних та небажаних станів та графік заданих ймовірностей помилки, 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.

Результати прогнозування співпадають з даними практичного управління процесом тестування в ІТ-компанії м. Вінниці.

Висновки

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

Застосована модель розгалужено-циклічного технологічного процесу до процесу тестування ПЗ дає змогу прогнозувати час на виконання процесу. Складність застосування визначається тим, що існують стани, котрі не можливо одночасно віднести до бажаних чи небажаних.

Запропонований підхід на основі марковської моделі РЦТП дозволяє здійснювати прогнозування процесу тестування для кожного окремого модуля програмного продукту, що дозволяє прийняти обґрунтовані рішення на кожній контрольній операції всього підпроцесу.

Література

- [1] Процедура стохастичної оптимізації для моделі тестування з напівмарковськими переключеннями [Електронний ресурс]. Режим доступу: http://irpis-npuv.gov.ua/cgi-bin/irpis_npuv/cgiirpis_64.exe?C21COM=2&I21DPN=UJRN&P21DPN=UJRN&IMAGE_FILE_DOWNLOAD=1&Image_file_name=PDF/Mtkm_fiz_mat_2014_10_13.pdf;
- [2] Моделі і методи тестування програмних систем [Електронний ресурс]. Режим доступу: http://dSPACE.npuv.gov.ua/pitstream/handle/123456789/289/%D0%9A%D0%PE%D1%80%D0%PE%D1%82%D1%83%D0%PD_%231.pdf?sequence=1;
- [3] Критерій достатності процесу тестування програмного забезпечення / В. Яковина, М. Сенів, Я. Чабанюк, Д. Федасюк, У. Хімка // Комп'ютерні науки та інформаційні технології : [зб. наук. пр.] / відп. ред. Ю.М. Рашкевич. — Л. : Вид-во Львів. політехніки, 2010. — С. 346–358;
- [4] Durand J.P., Gaudoin O. Software reliability modelling and prediction with hidden Markov chains // Statistical Modelling (5) — 2005. — pp. 75–93;
- [5] Yamada S. S-shaped reliability growth modelling for software error detection / S. Yamada, M. Ohpa, S. Osaki // IEEE Transactions on Reliability. — 1983. — Vol. R-32, No. 5. — pp. 475–478;
- [6] Прийняття рішень в управлінні розгалуженими технологічними процесами : монографія / В. М. Дубовой, Г. Ю. Дерман, І. В. Пилипенко, М. М. Байас. — Вінниця : ВНТУ, 2014. — 216 с;
- [7] Дубовой В.М., Пилипенко І.В., Стець Р.С. Застосування марковської моделі для аналізу впливу циклічності на управління розгалуженим технологічним процесом. Наукові праці ВНТУ, 2014, № 4 [Електронний ресурс] – Режим доступу: <http://praci.vntu.edu.ua/article/view/3827/5583>;



ТЕХНИЧНИ ЗАСОБИ ТЕХНОЛОГІЇ В СИСТЕМАХ УПРАВЛІННЯ

- [8] Scrum и XP: заметки с передовой [Elektronnyi resurs]. Rezhim dostupu: http://scrum.org.ua/wp-content/uploads/2008/12/scrum_xp-from-the-trenches-rus-final.pdf.

References

- [1] Protsehura stohastychnoi optmizatsii dlia modeli testuvannia z napivmarkovskumu pereklučeniamu [Elektronnyi resurs]. Reshym dostupu: http://irpis-npuv.gov.ua/cgi-bin/irpis_npuv/cgiirpis_64.exe?C21COM=2&I21DPN=UJRN&P21DPN=UJRN&IMAGE_FILE_DOWNLOAD=1&Image_file_name=PDF/Mtkm_fiz_mat_2014_10_13.pdf;
- [2] Modeli i metody testuvannia prohramnyh system [Elektronnyi resurs]. Reshym dostupu: http://dspace.npuv.gov.ua/pitstream/handle/123456789/289/%D0%9A%D0%PE%D1%80%D0%PE%D1%82%D1%83%D0%PD_%231.pdf?sequence=1;
- [3] Kryterii dostatnosti protsesu testuvannia prohramnoho zabezpechennia / V. Iakovyna, M. Seniv, IA. Chabanuk, D. Fedasuk, U. Himka // Komputerni nauky ta informatsiini tehnologii : [zb. nauk. pr.] / vidp. red. U. M. Rashkevych. — L. : Vyd-v Lviv. politehniky, 2010. — С. 346–358;
- [4] Durand J.P., Gaudoin O. Software reliability modelling and prediction with hidden Markov chains // Statistical Modelling (5) — 2005. — pp. 75–93;
- [5] Yamada S. S-shaped reliability growth modelling for software error detection / S. Yamada, M. Ohpa, S. Osaki // IEEE Transactionson Reliability. — 1983. — Vol. R-32, No. 5. — pp. 475–478;
- [6] Pryiniattia rishen v upravlinni rozhaluzhenymy tehnologichnymy protsesamy : monografia / V. M. Dubovoy, G. U. Derman, I. V. Pylypenko, M. M. Baias. — Vinnytsia : VNTU, 2014. — 216 s;
- [7] Dubovoy V. M., Pylypenko I. V., Stets R. S. Zastosuvannia markovskoi modeli dlia analizu vplyvu tsyklichnosti na upravlinnia rozhaluzhenym tehnologichnym protsesom. Naukovi pratsi VNTU, 2014, № 4 [Elektronnyi resurs]. Reshym dostupu: <http://praci.vntu.edu.ua/article/view/3827/5583>;
- [8] Scrum i XP: zametki s peredovoi [Elektronnyi resurs]. Reshym dostupu: http://scrum.org.ua/wp-content/uploads/2008/12/scrum_xp-from-the-trenches-rus-final.pdf.

Отримано в редакцію: 10.11.2015 р./ Прийнято до друку: 11.12.2015 р./ Received by edition: 10.11.2015. Approved for the press: 11.12.2015

УДК 621.039.56

МОДЕЛИРОВАНИЕ ИЗМЕНЕНИЯ СТРУКТУРЫ ТЕХНИЧЕСКИХ СРЕДСТВ АВТОМАТИЗАЦИИ ПРИ РАБОТЕ АЭС С ВВЭР-1000 В МАНЕВРЕННОМ РЕЖИМЕ.

Simulation of changes in the structure of technical means of automation at the VVER-1000 while maneuvering

Плахотнюк А.А.¹, Коккол Е.А.¹, Максимов М.В.² (Plakhotnuk A.A., Kokolo E.A., Maksimov M.V.)

^{1,2} Одесский национальный политехнический университет

² E-mail: prof.maksimov@gmail.com

Copyright © 2014 by author and the journal “Automation technological and business - processes”.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



DOI: [10.15673/2312-3125.24/2015.56334](https://doi.org/10.15673/2312-3125.24/2015.56334)

Аннотация

На атомных электростанциях станциях (АЭС) выдвигаются высокие требования к надежности и безопасности энергоблока. Особое внимание этому уделяется при проведении так называемого маневра мощностью. При эксплуатации энергоблока для более эффективной работы используются три различные