

РИЗИКО-ОПЕРАЦІЙНИЙ ПІДХІД ДО ВИРІШЕННЯ ПРОБЛЕМИ ОПТИМАЛЬНОГО ВИПУСКУ ПРОГРАМНИХ СИСТЕМ

Г.Б. Мороз, Т.М. Коротун

Київський національний університет внутрішніх справ України
02121, Київ, Колекторна, 4,
тел.: (044) 561 1813;
Інститут програмних систем НАН України
03187, Київ, проспект Академіка Глушкова, 40,
тел.: (044) 526 4579

Запропоновано підхід до вирішення задачі оптимального випуску програмних систем, базований на аналізі ризику відмов програмних компонентів і врахуванні операційного профілю функціонування системи при формуванні критерію оптимізації. Розроблено метод оцінки ризику відмов програмних компонентів та модель для визначення оптимального часу тестування програмних модулів, яка враховує ризики відмов модулів під час експлуатації.

The approach to the decision of task of optimal software systems release is represented. It is based on the software failure risk analysis and of operational profile of system functioning for building the optimization criteria. The method of modules risks evaluation has been developed. A model for the determination of the optimum time of the system's modules testing has been developed. It takes into account risks of module failures during the system operation.

Вступ

Однією з найважливіших проблем, які стоять перед розробниками програмних систем (ПС), є обґрунтоване ухвалення рішення про момент випуску ПС, тобто про припинення тестування і передачу користувачеві. Відомо, що чим довше триває тестування, тим вище якість, зокрема надійність, і менше вартість супроводу ПС. Однак затримка випуску ПС може викликати значні штрафи й суттєво збільшити вартість тестування. Водночас наслідки постачання ПС низької якості – це завжди збитки користувачів: не лише матеріальні та фінансові, але і падіння престижу. Як зазначає В.В. Липаєв [1] втрати користувачів через відмови у ПС можуть значно перевищувати користь від їх використання.

Ці проблеми, в свою чергу, негативно відображаються на конкурентоспроможності організації-розробника програмних продуктів [2].

Звичайно, тестування дозволяє знизити ризик відмов шляхом пошуку дефектів, а також невизначеність щодо ризику, надаючи інформацію для керування розробкою. З огляду на обмеженість часу тестування – завжди постає питання знаходження оптимального моменту випуску ПС.

Очевидно, що оптимальний час випуску ПС суттєво залежить від ступеня досягнутого компромісу між такими чинниками як надійність, вартість та час. Для вирішення цієї проблеми запропоновано низку підходів, що базуються на різних припущеннях, цільових критеріях та умовах експлуатації ПС [3–20].

Ґрунтуючись на аналізі літературних джерел зупинимося на деяких моментах, пов'язаних з даною проблемою.

1. У задачах оптимального випуску ПС їх, зазвичай, розглядають як чорний ящик, тобто не враховують розмір, складність, структуру, умови експлуатації ПС тощо [3–12].

2. У залежності від обраного критерію оптимізації і необхідних обмежень розрізняють пряму та обернену задачі оптимального випуску ПС [8–10].

Пряму задачу звичайно формують наступним чином: знайти такий момент часу T^* , щоб

$$C(T^*) = \min_{0 < T < T_L} \{C(T) : R(x|T) \geq R_0\},$$

де $C(T)$ – загальна очікувана вартість розробки ПС або середній загальний прибуток від реалізації ПС;

$R(x|T)$ – функція експлуатаційної надійності ПС, яка тестувалася протягом часу T ;

R_0 – необхідна надійність ПС; T_L – сумарний час тестування та експлуатації ПС.

Обернена задача оптимального випуску ПС може бути поставлена так: знайти такий момент часу T^* , щоб

$$R(x|T^*) = \min_{0 < T < T_L} \{R(x|T) : C(T) \leq C_0\},$$

де C_0 – верхня межа допустимої вартості розробки ПС.

Огляд основних типів критеріїв цієї групи, базованих на використанні моделей надійності можна знайти в [13].

3. Сучасні складні ПС складаються з багатьох розподілених програмних застосувань (компонентів, модулів), які розроблюються незалежними групами в різних умовах та середовищах. Для таких систем проблема визначення оптимального випуску може формулюватися як комбінаторна проблема оптимізації з відомими вартістю, надійністю, трудовитратами та іншими атрибутами компонентів системи.

Найбільш відомі підходи – максимізація надійності ПС, або мінімізація вартості (ресурсів). Така задача відома для вбудованих систем і вирішується методами динамічного програмування [14–17].

Запропоновано підходи, також, до оптимального розподілу надійності по компонентах ПС до початку тестування, з метою максимізації задоволення користувача при відомих технічних та вартісних обмеженнях [16–20]. Зокрема, в [20] наведена постановка задачі оптимального розподілу надійності для ПС з кількома застосуваннями у вигляді двох зв'язаних проблем:

- з зафіксованою цільовою інтенсивністю відмов;
- з зафіксованим часом тестування.

Інтенсивності відмов компонентів, які використовуються для побудови застосування, пов'язуються з вартістю тестування за допомогою моделей зростання надійності. Запропоновані рішення проблеми для одного застосування ПС. У випадку багатьох застосувань рішення не наведені, але вони можуть бути отримані з використанням інструментів нелінійного програмування.

В усіх розглянутих підходах не враховується серйозність наслідків відмов, а лише їхня ймовірність.

У даній роботі запропоновано так званий ризико-операційний підхід до розв'язання задачі оптимального випуску ПС, базований на диференційованому розподілі часу тестування між компонентами ПС з урахуванням ризику їх відмов під час використання ПС.

Розроблено метод оцінювання ризику відмов, в основу якого покладено поняття операційного профілю і загроз відмов. Перевага даного підходу над розглянутими полягає у тому, що він краще дозволяє мінімізувати експлуатаційні ризики відмов ПС, оскільки в значній мірі враховує як її структуру, так і майбутнє використання.

1. Операційний профіль та оцінювання ризику відмов програмної системи

Адекватність тестування є головним припущенням усіх моделей зростання надійності. Вона полягає у тому, що тестові дані та функції ПС вибираються відповідно до частоти їх очікуваного використання, а середовище тестування – максимально наближене до умов експлуатації. В інженерії надійності забезпечення адекватності тестування пов'язується з побудовою “операційного профілю” [21].

Операційний профіль використовується для обчислення частотного розподілу очікуваного використання окремих складових ПС (функцій, модулів, операцій). Це дозволяє диференціювати складові ПС за частотою використання та будувати на цій основі ефективні сценарії тестування.

Згодом, концепція операційного профілю була узагальнена та отримала назву *концепції профілів* – системи поглядів (користувачів, розробників) на процес використання ПС з позицій визначення кількісних характеристик елементів цього процесу, облік яких дозволяє підвищити ефективність тестування ПС [22].

Існують різні підходи до визначення і опису профілів ПС, але їх об'єднує погляд на специфікацію дій у процесі експлуатації ПС як на сукупність двох аспектів опису використання, а саме, сценаріїв використання та ймовірності здійснення цих сценаріїв [23].

Визначення операційного профілю ПС є складовою запропонованого в даній роботі методу оцінки ризику відмов ПС та стратегії тестування, яка полягає у визначенні програмних компонентів (ПК), які потребують більш тривалого тестування.

Стосовно тестування кожного ПК вводиться поняття його корисності. Тестування вважається прибутковим, якщо його вартість, а також вартість усунення дефектів, не перевищують очікуваних втрат користувача.

В основу здійснення стратегії тестування покладається:

- аналіз ризику відмов ПС;
- визначення внеску кожного компонента у загрози ПС;
- оцінювання очікуваної кількості відмов, які можуть виникнути у ПК при його використанні;
- оцінювання ризиків відмов ПК.
- визначення моменту оптимального випуску ПС, який максимізує функцію корисності.

1.1. Метод оцінювання ризику відмов. Введемо позначення:

$H = \{H_i, i=1,2,\dots,r\}$ – множина всіх потенційних загроз ПС, обумовлених відмовами через дефекти в її програмних компонентах;

$S = \{S_i, i=1,2,\dots,n\}$ – множина всіх можливих сценаріїв функціонування ПС;

t_0 – час виконання компонента в період експлуатації ПС;

t_e – час тестування;

C_m – внесок програмного компонента у ризик відмов ПС;

$R(t_0)$ – величина ризику відмови компонента за час t_0 ;

$\mu(t)$ – функція зростання надійності;

$P(S_i)$ – ймовірність того, що при реалізації сценарію $S_i (i=1,2,\dots,n)$ буде виконуватися даний компонент;

$P(H_j/S_i)$ – умовна ймовірність того, що при реалізації сценарію S_i причиною виникнення загрози H_j буде відмова саме даного компонента;

C_j – вартість наслідків реалізації загрози H_j ($j=1,2,\dots,r$).

Метод оцінювання ризику відмов ПК полягає у послідовному вирішенні задач на наступних етапах.

1.1.1. Перший етап. Вирішується задача ідентифікації можливих загроз ПК та оцінки вартості наслідків через здійснення загроз C_j , $j=1,2,\dots,r$. У ході аналізу визначаються:

- сценарії роботи користувачів в ПК;
- можливі сценарії подій, які можуть призвести до відмов ПК;
- типи і серйозність відмов ПК для користувача, які відповідають подіям у цих сценаріях;
- можливі наслідки для користувача від відмов ПК.

Припустимо, в ході аналізу сценаріїв роботи користувачів були виявлені загрози відмов ПК $H_1\dots H_r$. Тоді для кожної загрози H_j , розробляються відповідні сценарії подій у роботі ПК, які можуть призвести до відмови. Для кожної події в операційному сценарії розглядаються умови виникнення та серйозність відмов функцій ПК, які реалізуються компонентами ПК.

Серйозність наслідків відмов розглядається в контексті очікуваних збитків користувача через відмови системи при її експлуатації.

1.1.2. Другий етап. Визначається внесок кожного програмного компонента в ідентифіковані загрози для ПК зв'язування можливих відмов компонентів з зовнішніми zagrożами H_j та їхніми наслідками. Для цього будуються операційні сценарії роботи ПК за участю функцій, які реалізовані програмними компонентами, та визначається, на які зовнішні загрози можуть впливати можливі відмови компонента у кожному операційному сценарії S_i . Ймовірність використання компонента m у сценарії S_i визначається за формулою: $P(S_i) = kn$, де n – загальна кількість сценаріїв у роботі ПК, а k – кількість сценаріїв, у яких використовується компонент.

Після цього виконується співвіднесення загроз з компонентами з огляду на частоту використання кожного компонента в сценарії S_i . Після того, як встановлено, що компонент відноситься до загрози H_j при i -му використанні, робиться розподіл умовних ймовірностей $P(H_j/S_i)$ для всіх загроз, які можуть мати місце при заданому сценарії використання S_i .

Тоді, внесок кожного компонента C_m у всі ідентифіковані загрози обчислюється за формулою:

$$C_m = \sum_{i=1}^n \left[P(S_i) \sum_{j=1}^r P(H_j / S_i) \cdot C_j \right]. \quad (1)$$

1.1.3. Третій етап. Виконується оцінка очікуваної кількості відмов компонента, які можуть виникнути при його виконанні за час t_0 , за допомогою функції зростання надійності $\mu(t_0)$, конкретна форма якої визначається обраною моделлю. Параметри моделі можуть визначатися шляхом прогнозування або оцінювання [24].

При використанні прогнозування параметри моделі надійності визначаються з огляду на структуру, складність та час використання ПК з наступним уточненням за результатами тестування. Застосування оцінювання параметрів моделей надійності потребує даних про результати тестування (моменти відмов чи інтервалів часу між відмовами). У цьому разі оцінка параметрів виконується статистичними методами (найменших квадратів чи максимальної правдоподібності).

1.1.4. Четвертий етап. Для кожного програмного компонента ризик оцінюється з використанням відповідних оцінок внеску та очікуваної кількості відмов $\mu(t_0)$ в період T за формулою: $R(t_0) = C_m \mu(t_0)$.

З початком тестування фіксується час відмов та виконується пошук і усунення дефектів. Зібрані дані про відмови модуля та час виконання модуля t_0 при експлуатації використовуються для вибору та оцінки параметрів моделі надійності.

Результати цієї оцінки можуть спричинити зміну виконаного попереднього розподілу модулів за критичністю (модулі з невеликим внеском можуть виявитися не надійними).

Впродовж циклів *тестування/усунення дефектів* на основі аналізу нових даних про відмови виконується повторне оцінювання параметрів моделі надійності (або вибір іншої моделі).

Оновлені параметри використовуються для отримання нових оцінок параметрів функції $\mu(t_0)$.

Зміни в проекті ПК можуть спричинити зміну оцінок внеску компонента. Зміни середовища можуть призвести до потреби у зміні оцінок зовнішнього ризику.

Таким чином, оцінювання ризику розподілено в часі та може застосовуватися як механізм керування ризиком під час тестування. Це дозволяє виявляти схильні до відмов компоненти, та класифікувати їх за критичністю, а отже ретельніше спланувати та виконувати тестування.

2. Визначення оптимального моменту випуску ПК

Нехай:

$\lambda(t) = d\mu(t)/dt$ – інтенсивність відмов модуля;

$C(t_e)$ – повна вартість тестування протягом часу t_e ;

c_1 – вартість одиниці часу тестування;

c_2 – вартість усунення дефекту, що призвів до відмови в процесі тестування;

$\Delta R(t_0/t_e)$ – функція зниження ризику відмови компонента за час його виконання t_0 , за умови, що компонент тестувався час t_e ;

$K(t_0/t_e)$ – прибуток за час експлуатації ПС, за умови, що компонент тестувався час t_e ;

Тоді задачу можна сформулювати таким чином: знайти такий момент випуску ПС t_e^* , щоб прибуток був максимальним, тобто

$$t_e^* = \{t_e: K(t_0/t_e^*) \geq K(t_0/t_e), 0 \leq t_e \leq t_{\text{доп}}\},$$

де $t_{\text{доп}}$ – максимально допустимий час тестування.

Для розв'язання цієї задачі перш за все знайдемо функцію зниження ризику відмови компонента за час експлуатації ПС за умови, що компонент тестувався час t_e .

Нехай $\mu(t_0/t_e)$ – кількість відмов компонента за час t_0 його виконання у процесі експлуатації ПС, за умови, що компонент тестувався час t_e . Тоді, враховуючи, що $\mu(t_0/0) = \mu(t_0)$, функція зниження ризику відмови компонента за час t_0 , за умови, що компонент тестувався час t_e , дорівнює:

$$\Delta R(t_0/t_e) = C_m \mu(t_0/0) - C_m \mu(t_0/t_e) = C_m (\mu(t_0/0) - (\mu(t_0 + t_e) - \mu(t_e))) = C_m (\mu(t_0) - \mu(t_0 + t_e) + \mu(t_e)),$$

де C_m – внесок даного компонента в загальний ризик ПС, який обчислюється за формулою (1).

Загальні витрати на тестування можна визначити так: $C(t_e) = c_1 t_e + c_2 \mu(t_e)$. Тоді прибуток від тестування буде дорівнювати:

$$K(t_0/t_e) = \Delta R(t_0/t_e) - C(t_e) = C_m (\mu(t_0) - \mu(t_0 + t_e) + \mu(t_e)) - c_1 t_e - c_2 \mu(t_e). \quad (2)$$

Похідна функції $K(t_0/t_e)$ по t_e дорівнює

$$K'(t_0/t_e) = C_m (\lambda(t_e) - \lambda(t_0 + t_e)) - c_1 - c_2 \lambda(t_e). \quad (3)$$

Значення оптимального часу t_e^* отримуємо як розв'язок рівняння $K'(t_0/t_e) = 0$ залежно від конкретної функції $\lambda(t)$.

Так у випадку експоненційної моделі зростання надійності $\lambda(t) = \nu \beta \exp(-\beta \cdot t)$, тоді з (3) маємо:

$$K'(t_0/t_e) = C_m (\nu \beta \exp(-\beta t_e) - \nu \beta \exp(-\beta(t_0 + t_e))) - c_1 - c_2 \nu \beta \exp(-\beta t_e).$$

З рівняння

$$C_m (\nu \beta \exp(-\beta t_e) - \nu \beta \exp(-\beta(t_0 + t_e))) - c_1 - c_2 \nu \beta \exp(-\beta t_e) = 0$$

знаходимо єдину точку екстремуму функції $K(t_0/t_e)$, яку позначимо t_e^* .

$$t_e^* = -\frac{1}{\beta} \ln \left(\frac{c_1}{\beta \nu (C_m (1 - \exp(-\beta t_0)) - c_2)} \right). \quad (4)$$

Друга похідна функції $K(t_0/t_e)$ по t_e дорівнює

$$K''(t_0/t_e) = \beta^2 \nu \exp(-\beta t_e) (C_m (\exp(-\beta t_e) - 1) + c_2).$$

Оскільки для даної моделі надійності $\beta > 0$ і $c_1 > 0$, то

$K''(t_0/t_e^*) = -\beta c_1 < 0$. Отже точка t_e^* , якщо вона тільки існує, – є точкою максимуму функції $K(t_0/t_e)$. Неважко переконатися, що t_e^* існує тоді й тільки тоді, коли виконується умова:

$$1 - \exp(-\beta t_0) > \frac{c_1}{C_m \beta \nu} + \frac{c_2}{C_m}. \quad (5)$$

Дійсно, оскільки $\beta > 0$, то для того щоб логарифм в (4) існував і при цьому значення часу тестування не було від'ємним, параметри моделі повинні задовольняти системі нерівностей

$$C_m (1 - \exp(-\beta t_0)) > c_2,$$

$$\beta \nu (C_m (1 - \exp(-\beta t_0)) - c_2) > c_1.$$

Звідки легко одержуємо (5). З цієї умови можна також оцінити величину часу використання модуля t_0 :

$$t_0 > -\frac{1}{\beta} \ln \left(1 - \frac{1}{C_m} \left(c_2 + \frac{c_1}{\beta \cdot \nu} \right) \right).$$

Для того, щоб $t_0 > 0$ має виконуватися умова

$$0 < 1 - \frac{1}{C_m} \left(c_2 + \frac{c_1}{\beta \cdot \nu} \right) < 1, \quad \text{звідки випливає, що } C_m > c_2 + \frac{c_1}{\beta \cdot \nu}.$$

Нажаль, це єдиний випадок, коли можна легко здійснити дослідження функції прибутку $K(t_0/t_e)$ у загальному вигляді.

Так, наприклад, при використанні логарифмічної моделі зростання надійності рівняння для знаходження критичних точок функції прибутку $K(t_0/t_e) = 0$ має вигляд:

$$C_m \left(\frac{v\beta}{\beta(t_0 + t_e) + 1} - \frac{v\beta}{\beta \cdot t_e + 1} \right) + c_1 + c_2 \frac{v\beta}{\beta \cdot t_e + 1} = 0.$$

В інтервалі $(0, \infty)$ при виконанні певних умов це рівняння має один розв'язок відносно t_e :

$$t_e^* = \frac{-c_1(2 + \beta t_0) - c_2 v\beta + \beta \sqrt{D}}{2c_1 \beta},$$

де $D = c_1^2 t_0^2 + 4C_m c_1 t_0 v - 2c_1 c_2 t_0 v + c_2^2 v^2$.

Для моделі S-подібного зростання надійності з перегинами можна також знайти в загальному вигляді аналітичний розв'язок рівняння $K(t_0/t_e) = 0$ відносно t_e , проте він дуже громіздкий і тому не наводиться.

Для моделі S-подібного сповільненого зростання надійності розв'язок рівняння $K(t_0/t_e) = 0$ можна отримати чисельними методами.

4. Приклад застосування моделі

Запропонована модель була випробувана в інтегрованій ПС, яка складалася з семи програмних застосувань, кожне з яких, у свою чергу складалося з кількох програмних компонентів (модулів) та призначалася для функціонування у розподіленому середовищі Oracle RunTime, Oracle Express та MS Office (Word, Excel).

На етапі проектування було проведено аналіз загроз та оцінені збитки від їх виникнення, відповідно було визначено внески окремих програмних компонентів (модулів) у ці загрози.

Тестування виконувалося на тестовому стенді, який моделював умови використання ПС. Для реєстрації часу виконання t та моментів відмов у програмний код були вбудовані функції визначення моментів початку та завершення (нормального або аварійного) роботи компонентів (модулів) та їх реєстрації у базі даних.

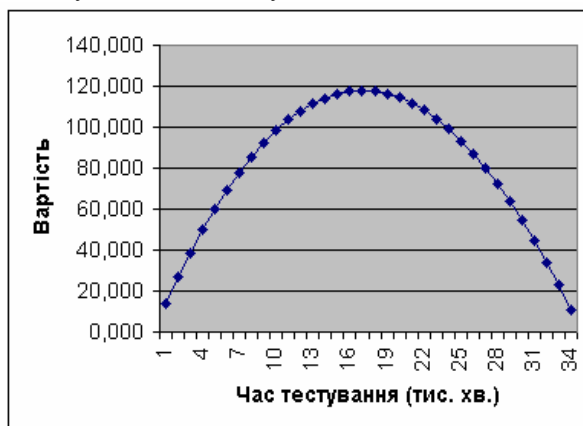
За результатами тестування були отримані дані про відмови модулів та оцінені параметри моделі надійності.

В таблиці наведено отримані оцінки часу використання, внеску, параметрів моделі надійності, ризику та оптимального часу тестування для на прикладі п'яти модулів.

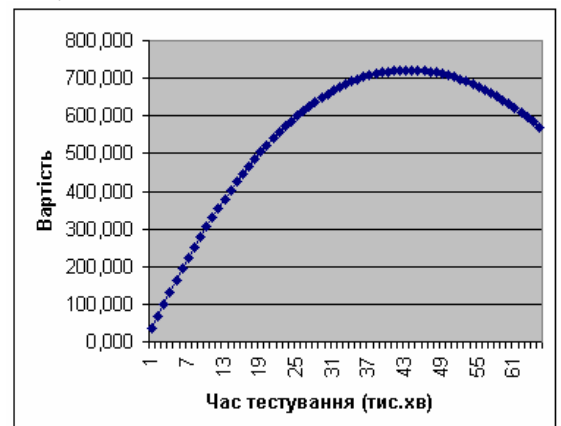
Таблиця

Модуль	Час (тис.хв.) t_0	Внесок (грн) $C_m(t_0)$	Параметри моделі надійності		Очікувана кількість відмов $\mu(t_e)$	Ризик модуля $R(t_0)$	Оцінка t^*
			ν	β			
M1	160	25000	42.8	0.000082	0.58625	14656.14	1699.54
M2	100	5000	43.1	0.000322	1.367	6835.33	1668.99
M3	160	29000	56.7	0.000076	0.685296	19872.61	4341.8
M4	160	10000	46.6	0.00083	1.14874	11487.4	3381.91
M5	100	16000	39.7	0.00017	0.74306	7440.63	488.246

На рисунку (а, б) зображені графіки функції $K(t_e)$ для модулів M1 та M3. Видно, що для модуля M1 $K(t_e)$ зростає, досягаючи в т. $t_e = 1699.54$ хв. максимуму, після чого, корисність від подальшого тестування починає зменшуватися. Для модуля M3 оптимальний час досягається в т. $t_e = 4341.8$ хв.



а



б

Рисунок. Поведінка графіка $K(t_e)$ для двох модулів: а – M1, б – M3

Застосування запропонованої моделі та методу дозволило визначити:

- програмні застосування, відмови яких становлять загрози відмови ПС;
- модулі, які роблять найбільший внесок у загрозу цілісності ПС та є критичними для вирішення задач користувачів ПС (модулі М1-М5);
- ризики відмов модулів;
- оптимальний час тестування кожного модуля з позицій зниження ризику його відмови під час експлуатації.

Випробування запропонованого методу оцінювання ризику та моделі визначення оптимального моменту випуску ПС показали її практичну корисність для побудови стратегії тестування ПС, спрямованої на зниження ризику відмов.

Висновки

Запропоновано ризико-операційний підхід до вирішення задачі оптимального випуску ПС. Використання цього підходу в інтегрованих ПС з модульною структурою дозволяє раціонально розподіляти час тестування між окремими програмними компонентами (застосуваннями, модулями), відбиває погляд користувача на якість використання ПС.

1. *Лунаев В.В.* Обеспечение качества программных средств. – М.: СИНЕРГ, 2001. – 380 с.
2. *Основы инженерии качества программных систем / Ф.И. Андон, Г.И. Коваль, Т.М. Коротун, В.Ю. Суслов // Под ред. И.В. Сергиенко. – Киев.: Академперіодика, 2002. – 504 с.*
3. *Kapur P., Garg R.* Optimum Software release policies for Software reliability growth model under imperfect debugging // *Rech. oper.* – 1990. 24, N 3. – P. 295–305.
4. *Musa J., Ackerman A.* Quantifying Software Validation: When to stop testing? // *IEEE Software* May. 1989. –P. 19–27.
5. *Kapur P., Garg R.* Optimum release policy for an inflection S-shaped Software reliability growth model // *Microelectron. Reliab.* –1991. –31, N 1. –P. 39–41.
6. *Ohtera H., Yamada S.* Optimum software-release time considering an error-detection phenomenon during operation // *IEEE Trans. Reliab.* – 1990. – R-39, –N 5. –P. 596 - 599.
7. *Yamada S., Osaki S.* Optimal software release policies with simultaneous cost and reliability requirements // *Eur. J. Operational Res.* –1987. –31, N 1. - P. 46–51.
8. *Yamada S., Osaki S.* Cost-reliability optimal release policies for software systems // *IEEE Trans. Reliab.*, –1985. - R–34, N5. – P. 422–424.
9. *Koch H., Kubat P.* Optimum release time of computer software // *IEEE Trans. Software Eng.*, –1983. - SE-6, N 3. –P. 323-327.
10. *Leung Y.* Optimum software release time with a given cost budget // *J. System. Software*, –1992. –17, –N 3. –P. 233-242.
11. *Musa J., Okumoto K.* Logarithmic Poisson Time Model for Software Reliability Measurement // *Proc. Sevent International Conference on Software Engineering*. – Orlando, Florida. – 1984. –P. 230–238.
12. *Yamada S., Ohtera H., Nanhisa H.* A testing-effort dependent software reliability model and Us application // *Microelectron. Rehab.* –1987. –27, N 3. – P. 507–522.
13. *Мороз Г.Б.* Пуассоновские модели роста надежности программного обеспечения и их приложение. Аналитический обзор // *УСИМ.* – 1996. – № 1-2. –С. 69 - 85.
14. *Berman O., Ashrafi N.* Optimization models for reliability of modular software systems // *IEEE Trans. Software Reliability.* -1993, vol. 19. – P. 1119 – 1123.
15. *Cai, M., Lyu M., Wong K, Ko R.* Component-based software engineering: Technologies, development frameworks, and quality assurance schemes // *Proc. Asia-Pacific Software Engineering Conf.*, Dec. –2000. – P. 372–379.
16. *Fyffe D., Hines W., Lee N.* System reliability allocation and a computational algorithm // *IEEE Trans. Reliability.* – 1968. vol. R-17. – P. 64–69.
17. *Tillman F.A., Hwang C.L., Kuo W.* Determining component reliability and redundancy for optimum system reliability // *IEEE Trans. Reliab.*, – 1977. vol. R-26. – P. 162–165.
18. *Zahedi F. and Ashrafi N.* Software reliability allocation based on structure, utility, price, and cost // *IEEE Trans. Software Eng.*, –Apr. –1991. vol. 17. – P. 345–355.
19. *Yacoub S. M., Cucik B., and Ammar H. H.* A component-based approach to reliability analysis of distributed systems // *Proc. 18th IEEE Symp. Reliable Distributed Syst.*, – 1999. – P. 158–167.
20. *Lyu M. R., Rangarajan S. et.* Optimal Allocation of Test Resources for Software Reliability Growth Modeling in Software Development // *IEEE Trans. on Reliab.* – 2002., JUNE Vol. 51, – N. 2
21. *Musa J.D.* Operational Profiles in Software Reliability Engineering // *IEEE Software.* – 1993. – Vol.10. – N 2, –P. 14-32.
22. *Коваль Г.И., Мороз Г.Б., Коротун Т.М.* Концепция профилей в инженерии надежности программных систем // *Математичні машини і системи.* – 2004. – № 1. – С. 166-184.
23. *Мороз Г.Б., Коваль Г.И., Коротун Т.М.* Применение концепции операционных профилей в инженерии надежности программных систем // *УкрПрог* –2004. – С. 188-197.
24. *Модели роста надежности ПО. Г.Б. Мороз, Е.М. Лаврищева* – Киев: 1992. – 25 с. - (Препр. НАН Украины. Ин-т кибернетики им. В.М. Глушкова; 92-38).