

УДК 004.413.5; 004.416.6

М. М. Сенів, Д. В. Федасюк, Ю. І. Парфенюк,
В. С. Яковина Я. М. Чабанюк

СИСТЕМА ОЦІНЮВАННЯ ТА ПРОГНОЗУВАННЯ НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

In this paper the architecture and implementation of the software system for software reliability evaluation and prediction based on mathematical reliability model with dynamic index of the software project size have been presented. The system gives the opportunity to computerize the experimental data processing and to evaluate the adequacy of the software testing process.

Keywords: *software reliability, software life-cycle, reliability model, software engineering.*

Архітектура та реалізація програмної системи для оцінювання та прогнозування надійності програмного забезпечення на основі математичної моделі надійності з динамічним показником величини програмного проекту. Система дає змогу автоматизувати обробку експериментальних даних та оцінити достатність процесу тестування програмного забезпечення.

Ключові слова: *надійність програмного забезпечення, життєвий цикл програмного забезпечення, модель надійності, програмна інженерія.*

В умовах бурхливого розвитку програмного ринку гостро стоїть проблема визначення надійності програмного забезпечення (ПЗ) та її кількісної оцінки на всіх етапах життєвого циклу. Для вирішення задач оцінки та прогнозування надійності в наш час використовують моделі надійності ПЗ [1]. Модель надійності ПЗ – це математична модель побудована для оцінки залежності надійності програмного забезпечення від певних параметрів.

Тепер є багато математичних моделей прогнозування надійності [3], кожна з них має свої переваги та недоліки, але всі відомі моделі оцінки та прогнозування надійності ПЗ побудовані на основі певних припущень і спрощень, а тому не достатньо повно відображають процес тестування, та їх результати не завжди збігаються з практичними. Тому виникає потреба у вдосконаленні та розробці нових моделей прогнозування надійності ПЗ, а також у виділенні кількісних параметрів та критеріїв оцінки надійності ПЗ. З огляду на це було запропоновано нову математичну модель оцінювання надійності ПЗ з динамічним показником величини програмного проекту [5,7]. Форми кривих функції кумулятивної кількості несправностей та інтенсивності появи помилок цієї моделі більш точно відповідають практичним результатам тестування ПЗ. На основі цієї моделі було описано та формалізовано критерій достатності процесу тестування ПЗ, який базується на поведінці залежності показника величини проекту від циклів тестування програмного продукту [6] та розроблено метод оцінювання та прогнозування надійності програмного забезпечення [2].

Мета цієї роботи полягає у проектуванні та розробці програмної системи для оцінювання та прогнозування надійності ПЗ на основі вищенаведених моделі та методу, яка б давала змогу автоматизувати обробку результатів процесу тестування ПЗ та давала практичні рекомендації щодо його достатності.

Короткий опис моделі з динамічним показником величини програмного проекту. Припустимо, що час виявлення помилок у моделі оцінювання та прогнозування надійності ПЗ розподілений за законом Пуассона. Крім того, вважаємо, що індекс величини проекту є параметром моделі та визначається на основі експериментальних даних і набуває значення з дійсного діапазону і завжди біль-

© М. М. Сенів, Д. В. Федасюк, Ю. І. Парфенюк, В. С. Яковина Я.М. Чабанюк, 2010

ший від нуля. Пропонуємо такий вигляд функції інтенсивності виявлення несправностей:

$$\lambda(t) = \alpha \beta^{s+1} t^s \exp(-\beta t), \quad (1)$$

де α – коефіцієнт, що визначає загальну кількість помилок в ПЗ; β – коефіцієнт, що характеризує загальну тривалість процесу виявлення помилок; s – індекс величини проекту, що узагальнює S -подібну модель.

Для інтенсивності (1) функція кумулятивної кількості несправностей має такий вигляд:

$$\mu(t) = \int_0^t \lambda(\tau) d\tau = \alpha [-\beta^s t^s e^{-\beta t} + s \Gamma_{\beta t}(s)], \quad (2)$$

де $\Gamma_z(p) = \int_0^z t^{p-1} e^{-t} dt$, ($\text{Re } p > 0$), – неповна гама-функція. Зауважимо, що при

$s = 1$ функція інтенсивності виявлення несправностей (1) та кумулятивна функція (2) збігаються з виглядом відповідних функцій S -подібної моделі.

Загальна кількість помилок в ПЗ визначається кумулятивною функцією, якщо $t \rightarrow \infty$, таким чином:

$$\mu(\infty) = \alpha s \Gamma(s), \quad (3)$$

де $\Gamma(s)$ – гама-функція.

Отже, аналітичний вигляд побудованої моделі дає змогу узагальнити вираз для загальної кількості помилок у системі (3), яка залежить від величини та складності проекту і визначається параметрами моделі. Крім того, частковий випадок при $s = 1$ (S -подібна модель) з урахуванням того, що $\Gamma(1) = 1$, дає значення $\mu(\infty) = \alpha$, що відповідає S -подібній моделі. Рівняння (1) та (2) будемо називати моделлю з динамічним показником величини проекту.

Проектування та реалізація програмної системи для кількісної оцінки та прогнозування надійності ПЗ. *Архітектура системи.* Під час проектування системи, опираючись на попередні розробки [4], було розроблено інформаційну модель системи [8] та визначено основні її функціональні можливості та частини, з яких буде складатися система. Основні етапи взаємодії системи із користувачем та її функціонування продемонстровано на діаграмі випадків використання (рис. 1).

Основні випадки використання:

1. *Введення початкових даних тестування проекту.* На цьому етапі здійснюється введення даних про тестування проекту, а саме:

- кількість проведених тестів (наприклад, 1200);
- кількість інтервалів або розмір інтервалу тестування;
- кількість помилок, які потрапляють в певні інтервали.

2. *Опрацювання введених даних про проект.* На цьому етапі готують дані для опрацювання їх певною моделлю надійності програмного забезпечення (ПЗ). Відбувається формування необхідних даних із введених даних користувачем, а саме:

– визначення інтервалів тестування та кількості помилок на відповідних інтервалах (масив T та M).

- визначається загальна кількість помилок, виявлених у процесі тестування (K).
- кількість ітерацій для цього етапу, яка відповідає кількості інтервалів (N).

3. *Пошук коефіцієнтів відповідно до моделі.* На цьому етапі здійснюється розв'язання системи нелінійних рівняння, яка реалізовує власне модель, а також пошук необхідних коефіцієнтів L , B , S для подальшого аналізу.

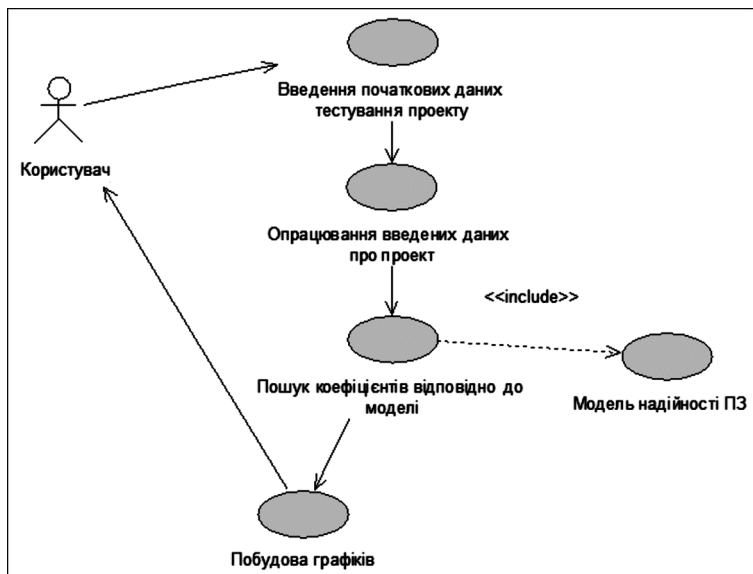


Рис. 1. Діаграма випадків використання.

4. *Модель надійності ПЗ.* Реалізовано модель, відповідно до якої здійснюється аналіз тестування проектів.

5. *Побудова графіків.* Цей етап є завершальним в роботі програми, здійснюється відображення результатів у графічній формі для більш зрозумілого аналізу достатності тестування проекту.

Для відображення послідовності дій для випадку побудови графіків результатів запропонованої моделі надійності програмного забезпечення використано UML діаграму дій (рис. 2).

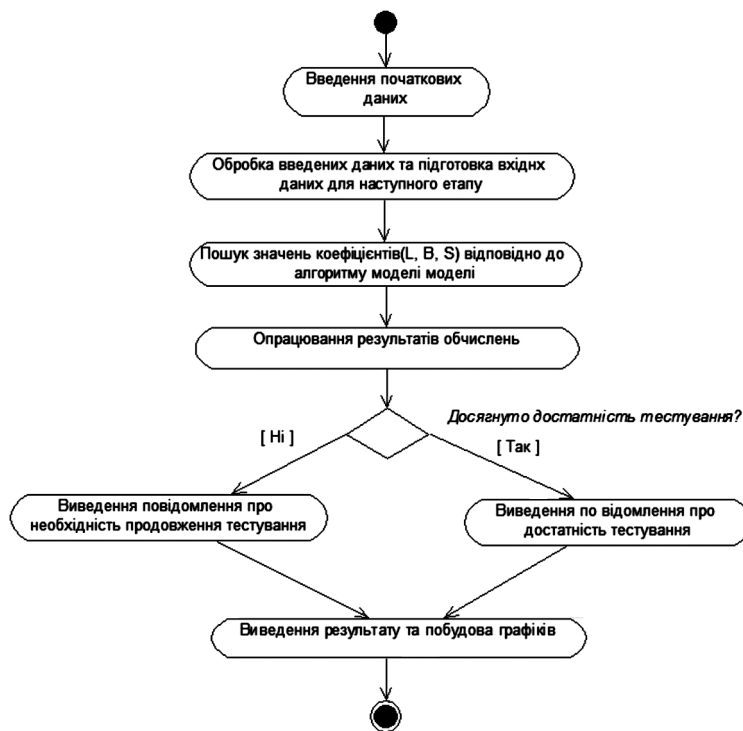


Рис. 2. Діаграма дій.

Отже, для отримання результатів визначення надійності програмного забезпечення відповідно до запропонованої моделі необхідно виконати такий потік дій:

- введення початкових даних. На цьому етапі вводиться кількість проведених тестів та поділ на відповідні інтервали, визначається час тестування, вводиться помилки, які знайдено на відповідних інтервалах тестування;
- обробка введених даних та підготовка до наступного етапу. На цьому етапі здійснюється опрацювання введених даних користувачем для подальшого використання в моделі визначення надійності програмного забезпечення;
- пошук значень коефіцієнтів відповідно до алгоритму запропонованої моделі. Здійснюється обчислення математичної моделі з введеними користувачем даними;
- на наступному етапі відбувається розгалуження на два потоки, які основані на знайдених результатах достатності тестування;
- на кінцевому етапі відбувається виведення результатів та побудова графіків.

Систему розроблено відповідно до об'єктно-орієнтованого підходу створення програмного забезпечення. Цей підхід дає можливість легко вдосконалювати та змінювати систему. Відповідно розроблено основні класи системи та відношення між ними. Нижче відображено діаграму класів системи (рис. 3). Діаграма послідовності дій відображає обчислення параметрів математичної моделі та збереження результатів.

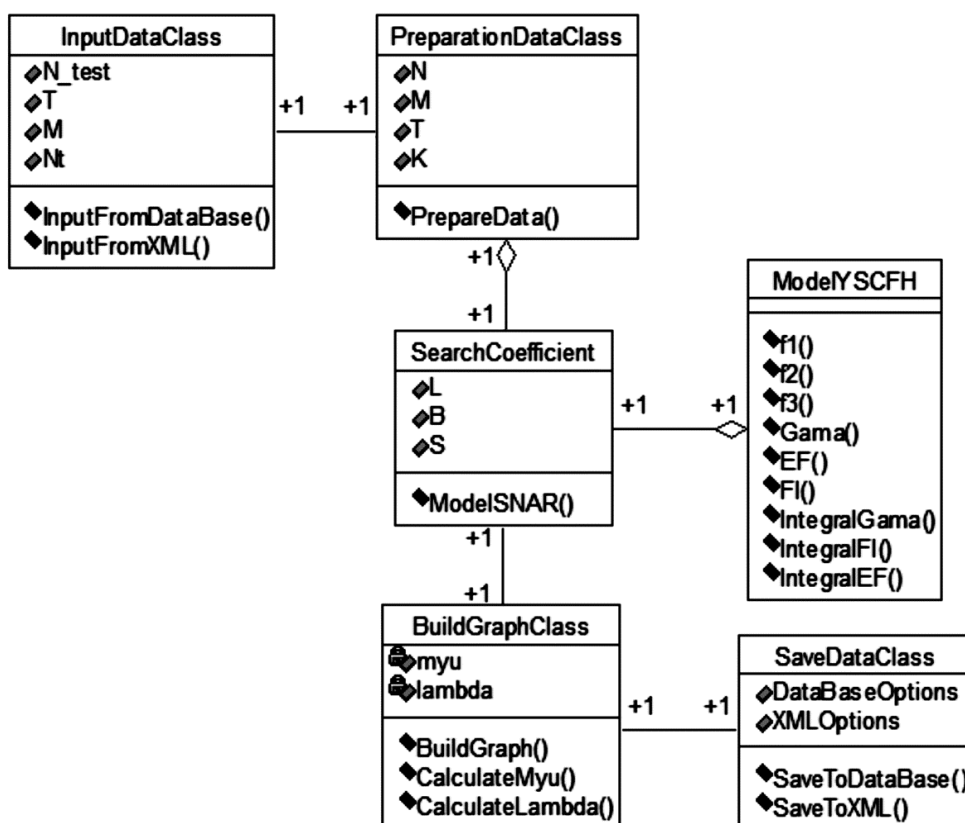


Рис. 3. Діаграма класів.

На основі вищевказаного було розроблено архітектуру (рис. 4) та формат подання даних (рис. 5) програмної системи для оцінювання та прогнозування надійності програмного забезпечення на основі математичної моделі надійності з динамічним показником величини програмного проекту.

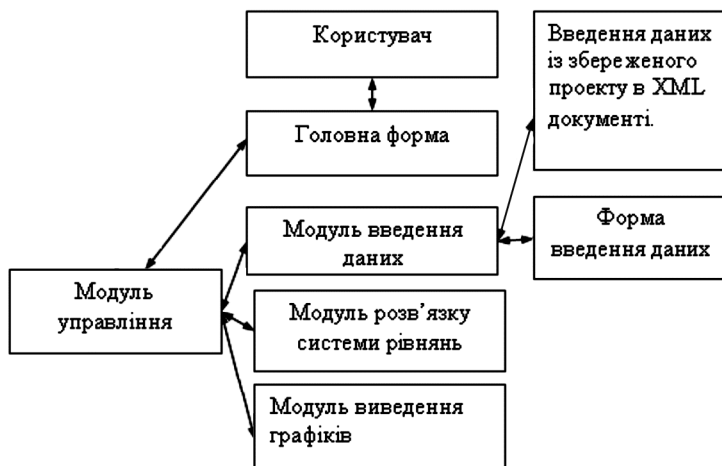


Рис. 4. Архітектура програмної системи оцінювання та прогнозування надійності програмного забезпечення.

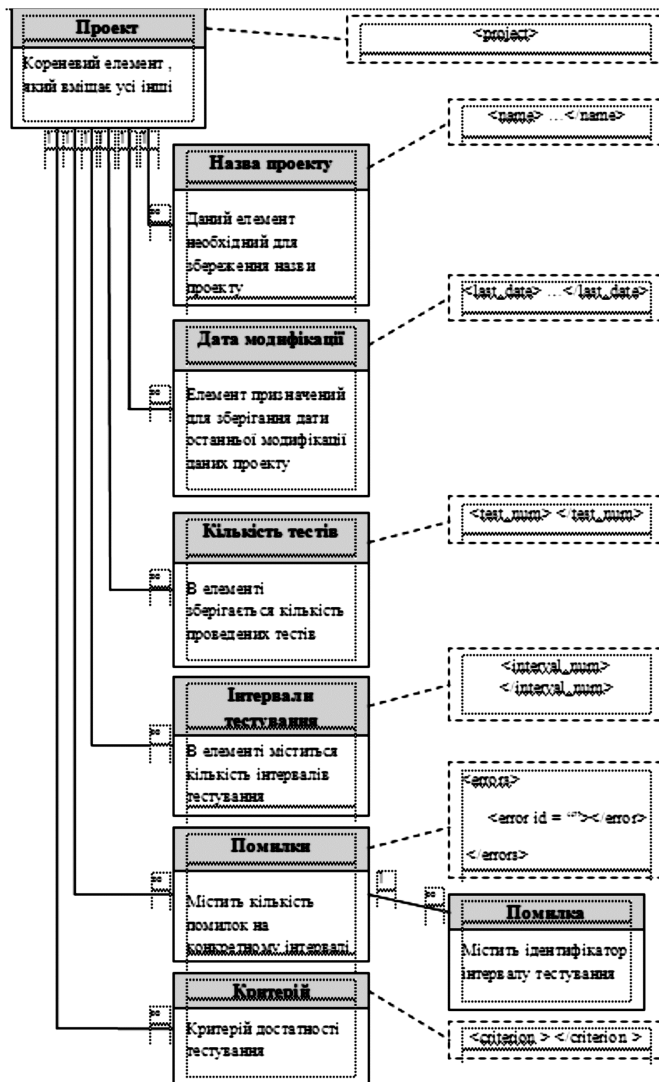


Рис. 5. Формат подання даних програмної системи.

Реалізація системи та приклад її застосування. Для створення системи було обрано мову програмування С#, яка є об'єктно-орієнтованою та входить до складу IDE середовища Microsoft Visual Studio 2008. Також для використання системи в середовищі Windows було обрано .NET Framework 3.5, який містить сучасні компоненти для полегшення процесу створення програмних систем.

Створена програмна система є WindowsFormsApplication і надає користувачеві можливість взаємодіяти із системою в звичному візуальному інтерфейсі. Для одночасної роботи з декількома документами систему реалізовано у вигляді Multi Document Interface.

Для створення нового проекту використовують "Майстер створення проекту". У процесі створення нового проекту необхідно вказати кількість проведених тестів та кількість рівномірних інтервалів, на які буде розбито всю кількість проведених тестів.

На наступному кроці створюється форма, в яку необхідно ввести кількість помилок, які припадають на певні інтервали.

Після введення кількості помилок відбувається обчислення необхідних коефіцієнтів відповідно до запропонованої моделі. Для побудови графіків результатів обчислень необхідно натиснути на відповідну кнопку на формі. Після цього буде обчислено значення функцій Λ та M і побудовано графіки.

Нехай виконано тестування певного програмного продукту. Відповідно проведено певну кількість тестів та отримано результати у вигляді виявлених помилок. Розіб'ємо всю кількість проведених тестів на частини, які складаються з 200, 500 та 1200 проведених тестів та розбиті на інтервали по 10 тестів.

Під час проведення 200 тестів бачимо, що графіки функцій не набули необхідного вигляду та гладкості (рис. 6).

За результатами проведених обчислень 200 тестів не є достатніми для досягнення критерію достатності тестування та припинення тестування.

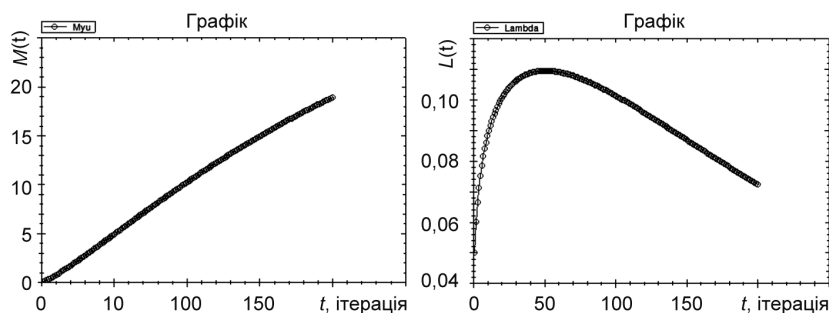


Рис. 6. Графіки функції M та Λ для 200 тестів.

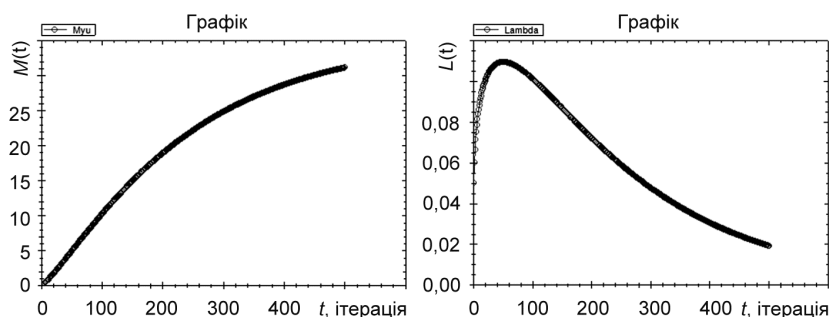


Рис. 7. Графіки функції M та Λ для 500 тестів.

Під час проведення 500 тестів графіки функцій набувають гладкішого вигляду (рис. 7) та монотонності. Але ця кількість проведених тестів не забезпечує до-

сягнення критерію достатності.

Під час виконання 1200 значення функцій набувають необхідного значення (рис. 8). Графік функції $M(t)$ досягає максимуму та залишається на цьому рівні, а графік функції $Lambda$ є близьким до нуля і не змінює свого вигляду. Також задовольняються усі умови критерію достатності тестування програмного продукту. На цьому етапі виводиться вікно повідомлення про можливість припинення тестування програмного продукту. Незважаючи на те, що критерій достатності тестування програмних продуктів досягнуто, програма дозволяє продовжувати дослідження процесу тестування.

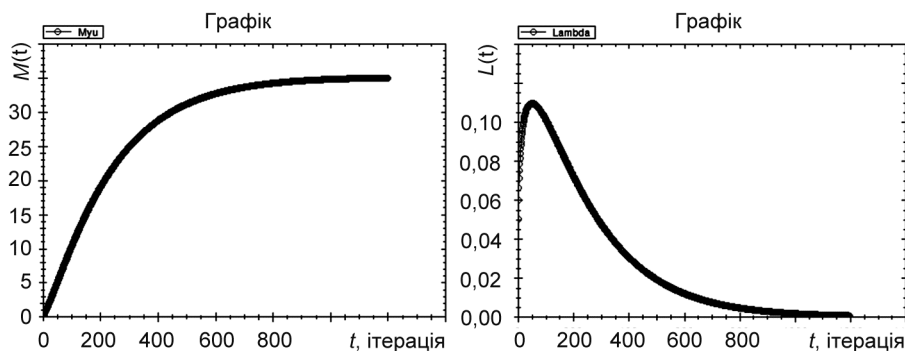


Рис. 8. Графіки функції $M(t)$ та $Lambda$ для 1200 тестів.

ВИСНОВКИ

Розроблено архітектуру та інформаційну модель програмної системи для оцінювання та прогнозування надійності ПЗ. Створена на їх основі програмна система реалізує запропоновану математичну модель надійності програмного забезпечення з динамічним показником величини програмного проекту. Система дає змогу автоматизувати обробку експериментальних даних та оцінювати достатність процесу тестування програмних продуктів, що дає можливість надати практичні рекомендації керівникам проекту стосовно розподілу виробничих ресурсів між етапами життєвого циклу розробки ПЗ.

1. *Липаев В. В.* Надежность программных средств. – М.: СИНТЕГ, 1998. – 232 с.
2. *Метод оцінювання та прогнозування надійності програмного забезпечення на основі моделі з динамічним показником величини проекту / М. М. Сенів, В. С. Яковина, Я. М. Чабанюк, Д. В. Федасюк // Комп'ютеринг. – 2010 (подано до друку).*
3. *Тейер Т., Липов М., Нельсон Э.* Надежность программного обеспечения / Пер. с англ. – М.: Мир, 1981. – 323 с.
4. *Програмний модуль прогнозування надійності системи теплового проектування / Д. Федасюк, М. М. Сенів, П. Сердюк, Н. Мамроха // Вісник Національного ун-ту "Львівська політехніка" Комп'ютерні науки та інформаційні технології. – 2008. – № 629. – С. 161–165.*
5. *Побудова і дослідження моделі надійності програмного забезпечення з індексом величини проекту / Я. М. Чабанюк, В. С. Яковина, Д. В. Федасюк, М. М. Сенів, У. Т. Хімка // Інженерія програмного забезпечення. – 2010 (в друці).*
6. *Критерій достатності процесу тестування програмного забезпечення / В. С. Яковина, М. М. Сенів, Я. М. Чабанюк, Д. В. Федасюк, У. Т. Хімка // Вісник Національного ун-ту "Львівська політехніка" Комп'ютерні науки та інформаційні технології. – 2010 (в друці).*
7. *The Estimation and Prediction Model of the Software Reliability with the Project Size Index / D. Fedasyuk, M. Seniv, Y. Chabanyuk, U. Khimka // Proc. Xth Int. Conf. "Modern problems of radio engineering, telecommunications, and computer science". – Lviv-Slavske, Ukraine, 2010. – P. 209–210.*
8. *Information model of data representation for software reliability estimation framework / D. V. Fedasyuk, M. M. Seniv, V. S. Yakovyna, N. M. Mamrokhа // Proc. Xth Int. Conf. "The Experience of Designing and Application of CAD Systems in Microelectronics CADSM 2009". – Lviv-Polyana, Ukraine. – 2009. – P. 287–291.*