

**В. Яковина, В. Смірнов**  
Національний університет “Львівська політехніка”,  
кафедра програмного забезпечення

## ОГЛЯД ОСНОВНИХ ПІДХОДІВ ДО АНАЛІЗУ НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

© Яковина В., Смірнов В., 2011

**Здійснено огляд і аналіз основних підходів до оцінювання надійності програмного забезпечення. Розглянуто моделі на основі статистичного опису поведінки відмов на основі архітектурного підходу та непараметричні моделі.**

**Ключові слова:** програмне забезпечення, надійність, архітектура, модель, оцінювання.

**The survey and analysis of the main approaches to assessing software reliability have been performed. The models based on statistical description of the failures behavior, the architectural approach models as well as non-parametric models have been reviewed.**

**Keywords:** software, reliability, architecture, model, estimation.

### Вступ

У багатьох дослідженнях окремо виділяють поняття надійності програмного забезпечення (ПЗ) [1–3], тому що із застосуванням понять надійності до програмних засобів варто враховувати особливості і відмінності цих об'єктів від традиційних технічних систем [4], для яких спочатку розроблялася теорія надійності:

- домінуючими факторами, що визначають надійність програм, є дефекти і помилки під час проектування і розробки;
- відносно рідкісне руйнування програмних компонентів і необхідність їхньої фізичної заміни приводить до принципової зміни понять збою і відмови програм;
- для підвищення надійності програмних комплексів особливого значення набувають методи автоматичного скорочення тривалості відновлення і перетворення відмов на короткочасні збої шляхом введення в програмні засоби часової, програмної та інформаційної надлишковості;
- непередбачуваність місця, часу та імовірності прояву дефектів, а також їхнє нечасте виявлення під час реальної експлуатації достатньо надійного ПЗ не дає змоги ефективно використовувати традиційні методи апріорного розрахунку показників надійності;
- традиційні методи форсованих випробувань надійності систем шляхом фізичного впливу на їхні компоненти не можна застосувати до програмних засобів.

З урахуванням зазначених особливостей застосування основних понять теорії надійності складних систем до життєвого циклу і оцінювання якості ПЗ можна адаптувати і розвивати цю теорію в особливому напрямку – надійність програмного забезпечення [4]. Сучасні підходи до оцінювання надійності ПЗ засновані на паралелях з надійністю апаратного забезпечення з урахуванням принципових відмінностей між програмним та апаратним забезпеченням [5].

Загальне визначення надійності ПЗ наведено в [5]. Нехай  $F$  – довільно визначений клас помилок, а  $T$  – відносна міра часу, одиниці вимірювання якої визначаються конкретною програмою, надійність якої оцінюється. Тоді надійністю програмного пакета відносно класу помилок  $F$  і відносно метрики часу  $T$  є імовірність того, що жодної помилки цього класу не виникне протягом виконання програми за наперед заданий відносний період часу.

### Оцінювання надійності ПЗ на основі статистичного моделювання поведінки відмов

За останні десятиліття було запропоновано багато аналітичних моделей для вирішення проблеми вимірювання надійності ПЗ. Ці підходи ґрунтуються переважно на історії спостереження помилок ПЗ і можуть бути класифіковані згідно з процесом дослідження помилок так [5]: моделі на основі часу між помилками, моделі на основі кількості помилок, моделі на основі висівання помилок, моделі на основі області вхідних даних.

У класі моделей на основі часу між помилками предметом дослідження є час між виявленнями помилок (відмовами). Найпоширенішим припущенням є те, що час між  $(i-1)$ -ю та  $i$ -ю помилками підлягає певному виду розподілу, параметри якого залежать від кількості помилок, що залишаються в програмі протягом цього інтервалу. Оцінки параметрів отримують на основі спостережених часових інтервалів між помилками, а оцінки надійності ПЗ (середній час до наступної помилки тощо) потім отримують на основі моделі з визначеними параметрами. Інший підхід трактує час між помилками як реалізацію стохастичного процесу і використовує відповідні моделі часових рядів для опису процесу виявлення помилок, що покладено в основу цього процесу.

Моделі цього класу є одними з найперших, запропонованих для оцінювання надійності ПЗ. Під час моделювання розподілу часу між помилками вважають, що час до наступної відмови буде більшим, оскільки дефекти вилучаються з програмної системи. Для певного набору експериментальних даних це може не повністю справджуватись, оскільки слід враховувати, що час між відмовами є випадковою величиною, і спостережувані значення мають статистичні флуктуації. Існує велика кількість моделей цього класу, серед яких найпоширенішими є деевтрофікаційна модель Jelinski–Moranda [6], модель Schick–Wolverton [7], модель недосконалого відлагодження Goel–Okumoto [8], байєсівська модель Littlewood–Verrall [9], в якій автори застосували інший підхід до розроблення моделі визначення часу між помилками.

На практиці широко використовують моделі на основі кількості помилок [1, 5]. Предметом дослідження моделей цього класу є кількість помилок у визначеному часовому інтервалі, а не час між помилками. У міру усунення дефектів з системи очікується, що виявлена кількість помилок в одиницю часу спадатиме. Вважають, що кількість помилок відповідає відомому стохастичному процесу з дискретною або неперервною інтенсивністю виявлення помилок, що залежить від часу. Параметри функції інтенсивності потоку відмов можна оцінити на основі спостережених значень кількості помилок чи за часом появи помилок.

Було запропоновано багато моделей, що описують такі явища. В основу більшості з цих моделей покладено розподіл Пуассона, параметри якого мають різний вигляд для різних моделей, оскільки використання такого розподілу випадкових величин добре зарекомендувало себе в багатьох областях, де основна зацікавленість полягає в кількості подій [5]. До таких моделей належать моделі: Shooman [10], Jelinski–Moranda [6], Schick–Wolverton [7], Musa [11], Schneidewind [12], Goel–Okumoto [13], S-подібна модель зростання надійності [14] тощо.

Основним підходом моделей на основі висівання помилок є “висівання” відомої кількості помилок в програму, яка, як вважають, має невідому кількість власних помилок. Після цього програма тестується і підраховується виявлена кількість висіяних та власних помилок. На основі цього з використанням методів комбінаторики та максимальної правдоподібності отримують оцінку кількості помилок в програмі до висівання, яка використовується для отримання оцінок надійності ПЗ та інших відповідних характеристик. Найпопулярнішою та найбільш фундаментальною моделлю цього класу є гіпергеометрична модель Mills [15].

Основним підходом моделей на основі області вхідних даних є генерування набору тестових прикладів з вхідного розподілу, який, як вважають, репрезентує цільове використання програми. Через складність отримання такого розподілу область вхідних даних розподіляють на набір класів еквівалентності, кожен з яких зазвичай пов'язують з одним зі шляхів виконання програми. Оцінку надійності ПЗ отримують на основі помилок, виявлених шляхом фізичного чи символічного виконання тестових прикладів, взятих з області вхідних даних. До моделей цього класу належать модель Nelson [16] та Ramamoorthy–Bastani [17], в якій автори розглядали надійність критичного ПЗ, ПЗ реального часу та ПЗ автоматизованого управління.

Оскільки в багатьох випадках помилки ПЗ виявляються корельованими випадковими величинами, адекватнішим засобом опису надійності ПЗ можуть стати неоднорідні марковські ланцюги [18]. Крім того, в моделях на основі пуассонового розподілу припускають, що інтенсивність виявлення помилок є неперервною функцією часу, тоді як процес усунення помилок вносить розриви до інтенсивності виявлення помилок.

Для усунення такого роду спрощень і припущень [19], а також для моделювання надійності програм з багатьма потоками виконання [20] використовують формалізм дискретних ланцюгів Маркова. Основною перевагою таких моделей є врахування наявності неоднорідних періодів в процесі усунення помилок. В сенсі прогнозування надійності моделі на основі ланцюгів Маркова виявляють в середньому кращі результати порівняно зі звичайними моделями, особливо у випадках, коли надійність зростає нерегулярно. Використання напівмарковських процесів дасть змогу адекватніше описувати поведінку реального ПЗ, де закон час виправлення помилки може залежати від типу помилки і не завжди описуватись експоненційною функцією.

### **Аналіз надійності ПЗ на основі архітектурного підходу**

Описаний у попередньому підрозділі клас моделей належить до моделей на основі “чорної скриньки”, оскільки в них не враховано структури досліджуваного програмного засобу, а опрацьовують тільки вхідні і вихідні дані тестування ПЗ.

На противагу такому підходу, моделі “білої скриньки” можуть адекватніше описати надійність програмного забезпечення, адже вони оцінюють внутрішню будову програмного забезпечення, тобто його архітектуру, тому цей тип моделей називають ще моделями, побудованими за архітектурним підходом. За таким підходом надійність системи розглядається як функція надійності її компонент.

Своєю чергою, моделі, побудовані за архітектурним підходом [21], поділяються на адитивні моделі, моделі на основі шляхів виконання програми, моделі на основі компонентного підходу.

В адитивних моделях для обчислення надійності системи використано тільки надійність кожної компоненти (модуля) програмного продукту, причому надійність системи є сумою надійності її модулів. Найвідомішою моделлю цього класу є модель Xie–Wohlin [22].

Для прогнозування надійності програмного продукту в підході на основі шляхів виконання програми використано інформацію про послідовність виконання компонент ПЗ, отриману емпіричним шляхом. Представниками моделей цього класу є моделі Shooman [23] та Krishnamurthy–Mathur [24].

У моделях надійності на основі компонентного підходу для опису архітектури ПЗ використовують граф потоку керування. Вважають, що передача контролю між компонентами має властивості марковського процесу. За цим підходом архітектуру ПЗ можна змоделювати як ланцюг Маркова з дискретним часом чи неперервним часом, а також напівмарковським процесом. Пізніше кожну з моделей можна класифікувати як поглинальну (містить поглинальний стан – стан, з якого система вийти не може) та непоглинальна (не містить поглинальних станів).

Окрім того, в цьому класі моделей можна виділити композиційні та ієрархічні моделі [25]. До композиційних належать моделі, які одночасно комбінують архітектуру програмного продукту та характер його помилок для обчислення надійності ПЗ. В ієрархічних моделях спочатку розв’язується архітектурна модель, а потім поведінка помилок системи додається до існуючого результату для прогнозування надійності ПЗ.

**Використання штучних нейронних мереж для аналізу надійності ПЗ.** Традиційні параметричні моделі зростання надійності ПЗ, такі як неоднорідний пуассонів процес успішно використовуються в практиці інженерії надійності ПЗ. Однак, сьогодні не існує єдиної параметричної моделі, яка дала б змогу точно прогнозувати надійність для усіх класів ПЗ. На додачу до параметричних моделей, непараметричні моделі, наприклад, такі як штучні нейронні мережі (НМ), можуть бути альтернативним методом прогнозування надійності ПЗ [26]. Так, за

останнє десятиліття доведено, що нейронні мережі можуть бути універсальним апроксиматором для будь-якої нелінійної неперервної функції з довільною точністю [26].

Більшість нейронних мереж, що використовуються для моделювання надійності ПЗ, можуть бути поділені на два класи [26]. Перший використовує кумулятивний час виконання як вхідні дані і відповідну кумулятивну кількість помилок як бажаний вихід. Цей клас використовує для моделювання різні архітектури нейронних мереж, такі як рекурентні НМ, мережі Елмана тощо. Другий клас моделює надійність ПЗ на основі НМ типу “множинний вхід із затримкою – одиничний вихід”. Так, наприклад, було використано значення останніх 50 часів між помилками в якості входів із затримкою для прогнозування часу виявлення наступної помилки [27].

Дослідження використання НМ для прогнозування надійності ПЗ йдуть у різних напрямках. Так, було показано, що з використанням комбінованих НМ можна істотно покращити прогнозування надійності ПЗ [28]. Інші дослідження використовують підхід на основі НМ для побудови динамічно зваженої комбінаційної моделі надійності ПЗ, яка дає змогу суттєво покращити результати прогнозування надійності ПЗ [26]. Ще один підхід заснований на використанні самогенерованих НМ з ітераційним навчанням (Iteration Learning Selfgenerating Neural Networks – ISGNN). З використанням такого типу НМ можна зменшити час навчання мережі до 1/6 від часу навчання мережі зворотного поширення похибки, а середня абсолютна похибка, як і середня квадратична похибка такої мережі зменшуються на 3–15 % порівняно з мережею зворотного поширення похибки [29].

### Висновки

Проаналізовано основні підходи до оцінювання надійності програмного забезпечення. Показано, що в основу більшості статистичних моделей покладено розподіл Пуассона, оскільки використання такого розподілу випадкових величин добре зарекомендувало себе в багатьох галузях, де основна зацікавленість полягає в кількості подій.

Для усунення спрощень і припущень статистичних моделей надійності ПЗ, а також для моделювання надійності програм на основі їх архітектури використовують формалізм дискретних ланцюгів Маркова. Основною перевагою таких моделей є врахування наявності неоднорідних періодів в процесі усунення помилок. Використання напівмарковських процесів дасть змогу адекватніше описувати поведінку реального ПЗ, де закон час виправлення помилки може залежати від типу помилки і не завжди описуватись експоненційною функцією.

Разом з тим сьогодні не існує єдиної параметричної моделі, яка б дала змогу точно прогнозувати надійність програмного забезпечення різних класів. Перспективним засобом розв’язання цієї задачі є непараметричні моделі, зокрема штучні нейронні мережі, які можуть бути альтернативним засобом прогнозування надійності ПЗ. Оптимальним вибором архітектури, структури та параметрів НМ можна досягти заданої точності апроксимації параметра потоку відмов, тобто аналізу та прогнозування показників надійності ПЗ.

1. Половко А.М., Гуров С.В. *Основы теории надежности*. – СПб.: БХВ-Петербург, 2008. – 704 с.
2. Тейер Т., Липов М., Нельсон Э. *Надежность программного обеспечения: Пер. с англ.* – М.: Мир, 1981. – 323 с.
3. О.В., Казарин. *Теория и практика защиты программ*. – М.: МГУИ, 2004. 450 с.
4. В.В., Лунаев. *Надежность программных средств*. – М.: СИНТЕГ, 1998. 232 с.
5. Goel, A.L. *Software reliability models: assumptions, limitations, and applicability* // *IEEE Transactions on software engineering*. 1985, Vol. SE-11, No 12, pp. 1411–1423.
6. Z. Jelinski and P. Moranda *Software reliability research* // *Statistical Computer Performance Evaluation*. – W. Freiberger, Ed. – New York: Academic. – 1972. – P. 465–484.
7. G.J. Schick and R.W. Wolverson *Assessment of software reliability* // *Proc. Oper. Res. – Physica-Verlag. – Wirzberg-Wien*. – 1973. – P. 395–422.
8. A. L. Goel and K. Okumoto *An analysis of recurrent software failures in a real-time control system* // *Proc. ACM Annu. Tech. Conf., ACM, Washington, DC, 1978, pp. 496–500*.
9. B. Littlewood and J. L. Verrall *A Bayesian reliability growth model for computer software* // *Appl. Statist., Vol. 22 (1973), pp. 332–346*.
10. M.L. Shooman *Probabilistic models for software reliability prediction* // *Statistical Computer Performance Evaluation*. – W.

Freiberger, Ed. – New York: Academic. – 1972. – P. 485–502. 11. J.D. Musa *A theory of software reliability and its application* // *IEEE Transactions on Software Engineering*. – SE-1(3). – 1975. – P. 312–327. 12. N.F. Schneidewind *Analysis of Error Process in Computer Software* // *Sigplan Note*. – Vol. 10. – No.6. – 1975. – P.337–346. 13. A.L. Goel, K. Okumoto *Time-Dependent Error-Detection Rate Model for Software and other Performance Measures* // *IEEE Transactions on Reliability*. – Vol. R-28. – No. 3. – 1979. – P. 206–211. 14. S. Yamada, M. Ohba, S. Osaki *S-shaped reliability growth modeling for software error detection* // *IEEE Transactions on Reliability*. – Vol. R-32. – No.5. – 1983. – P. 475–478. 15. H. D. Mills *On the statistical validation of computer programs* // *IBM Federal Syst. Div., Gaithersburg, MD, Rep. 72-6015*, 1972. 16. E. Nelson *Estimating software reliability from test data* // *Microelectron. Rel.*, Vol. 17 (1978), pp. 67–74. 17. C. V. Ramamoorthy and F. B. Bastani *Software reliability: Status and perspectives* // *IEEE Trans. Software Eng.*, Vol. SE-8 (1982), pp. 359–371. 18. K.-Y. Cai, D.-B. Hu, C.-G. Bai, H. Hu, T. Jing. *Does software reliability growth behavior follow a non-homogeneous Poisson process.* // *Information and Software Technology*. 2008, Vol. 50, pp. 1232–1247. 19. Durand J.B., Gaudoin O. *Software reliability modelling and prediction with hidden Markov chains* // *Statistical Modelling*. – Vol. 5 (1). – 2005. – P. 75–93. 20. Y.-M. Kwon, G. Agha. *A Markov Reward Model for Software Reliability* // *The Next Generation Software (NGS) Workshop at International Parallel and Distributed Processing Symposium (IPDPS)*, 2007, pp. 1–6. 21. S. Gokhale, W.E. Wong, K.S. Trivedi, and J.R. Horgan *An Analytical Approach to Architecture-Based Software Reliability Prediction* // *Performance Evaluation*, Vol 28 (2005), No. 1, pp. 51–64. 22. M. Xie, C. Wohlin *An additive reliability model for the analysis of modular software failure data* // *Proceedings of the Sixth International Symposium on Software Reliability Engineering*, 1995, pp. 188–194. 23. M. Shooman *Structural models for software reliability prediction* // *Proceedings of the Second International Conference on Software Engineering*, 1976, pp. 268–280. 24. S. Krishnamurthy, A.P. Mathur *On the estimation of reliability of a software system using reliabilities of its components* // *Proceedings of the Eighth International Symposium on Software Reliability Engineering*, 1997, pp. 146–155. 25. S. Gokhale, K.S. Trivedi *A Time/Structure Based Software Reliability Model* // *Annals of Software Engineering*, Vol. 8 (1999), pp. 85–121. 26. Y.-S. Su, C.-Y. Huang. *Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models.* // *The Journal of Systems and Software*. 2007, Vol. 80, pp. 606–615. 27. K.Y. Cai, L. Cai, W.D. Wang, Z.Y. Yu, D. Zhang. *On the neural network approach in software reliability modeling.* // *The Journal of Systems and Software*. 2001, pp. 47–62. 28. Zheng, J. *Predicting software reliability with neural network ensembles.* // *Expert Systems with Applications*, Vol. 36 (2009), Issue 2, Part 1, P. 2116–2122. 29. A. Li, D. Qiu, Z.H. Li. *Predicting Time between Software Failures Using ISGNN.* // *International Journal of Computer Science and Network Security*. 2006, Vol. 6, No 6, pp. 116-118.