

УДК 004.891

Бабій І.М. (ХмНУ)
Лєнков О.С. (НАУ)
к.т.н., доц. Огнєвий О.В. (ХмНУ)

АНАЛІЗ ПРОГРАМНО-АЛГОРИТМІЧНИХ ЗАСОБІВ ПРОВЕДЕННЯ ОЦІНКИ НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ,

У статті проведено аналіз програмно-алгоритмічних засобів проведення оцінки надійності програмного забезпечення (ПЗ), що дозволяє проводити розрахунок характеристик надійності ПЗ (таких як, час напрацювання до відмови, коефіцієнт готовності, ймовірність відмови.

Розглядаються методи тестування, що дозволяють підвищити надійність ПЗ. Описані основні складові функціональної надійності програмних систем. Наведені результати експериментальних досліджень характеристик виявлення помилок в складних програмах.

Визначені основні фактори, що впливають на надійність ПЗ. Запропоновані способи забезпечення і підвищення надійності ПЗ.

Ключові слова: корпоративні інформаційні системи, програмне забезпечення, надійність.

Вступ. Архітектура сучасних корпоративних інформаційних систем (КІС) є, як правило, функціонально розподіленою. Вона характеризується багатопотоковою організацією обчислень, при якій запити реалізуються паралельно і розподіляються по декількох процесорах (серверах). Основним засобом реалізації функцій обробки інформації та управління в таких системах є програмне забезпечення (ПЗ). Суттєвою особливістю КІС є безперервність процесів введення і обробки інформації, циклічний характер обчислювальних процесів. У зв'язку з цим найважливішою проблемою, яка виникає при створенні КІС, є забезпечення високого рівня надійності їх функціонування. У розподілених системах, архітектура яких забезпечує можливість повного або часткового резервування апаратних засобів, основним фактором, що визначає надійність функціонування, є програмне забезпечення.

Численні наукові публікації і накопичений досвід розробки програмних систем в Росії і за кордоном свідчать про те, що досить впевнено прогнозувати рівень надійності функціонування ПЗ важко. Проблема полягає в тому, що існуючі методи і моделі прогнозування надійності ПЗ не в повній мірі придатні для практичного застосування.

Постановка задачі. В даний час в області машинної обробки інформації існують дві взаємопов'язані проблеми: вартість обробки інформації і ненадійність програмного забезпечення, що організовує і виконує процес обробки інформації. При цьому перша проблема знаходиться в залежності від другої, так як висока вартість проектування, тестування і супроводу програм обробки інформації визначається насамперед ненадійністю ПЗ [2].

Необхідність підвищення надійності програмного забезпечення обумовлена ще й тим, що в даний час ПЗ несе значно більше функціональне навантаження в вирішенні задач управління, ніж технічні засоби.

Тому метою даної статті є аналіз програмно-алгоритмічних засобів проведення оцінки надійності програмного забезпечення, що дозволяє проводити розрахунок характеристик надійності ПЗ (таких як, час напрацювання до відмови, коефіцієнт готовності, ймовірність відмови) і на основі аналізу побудова в подальшому моделі прогнозування змін цих характеристик у часі.

Аналіз відомих досліджень і публікацій. У характеристиках і метриках якості програмного забезпечення дається визначення основних характеристик якості ПЗ, а також наводяться рекомендації щодо їх виміру. Зокрема, визначена номенклатура показників

надійності ПЗ та вводиться шість характеристик якості, в тому числі для оцінки надійності: завершеність, стійкість до помилок, відновленість, узгодженість, правильність роботи, своєчасність. Основні показники якості ПЗ відображені в таблиці 1.

Таблиця 1

Показники якості ПЗ

Показник	Опис
Зручність супроводу	ПЗ повинно бути таким, щоб існувала можливість його удосконалення у відповідь на зміни вимог замовника або користувача
Надійність	Визначається рядом характеристик, таких як безвідмовність, захищеність і безпека
Ефективність	ПЗ повинно розумно витратити ресурси і володіти достатньо швидкісними часовими характеристиками
Зручність у використанні	ПЗ має бути зручним в експлуатації і бути розрахованим на технічний рівень експлуатуючого персоналу, мати відповідний користувацький інтерфейс і документацію

Дані показники не впливають безпосередньо з того, які дії може виконувати програмний продукт. Вони характеризують поведінку програми при виконанні цих дій.

Надійність - один з найважливіших факторів, що визначає загальну продуктивність і ефективність систем. У зв'язку з цим вже на стадії проектування системи питанню надійності повинна приділятися значна увага. У період, коли встановлюється початкова взаємозалежність між характеристиками системи, витратами і графіком виконання робіт, повинні бути сформульовані і вимоги до надійності, так як саме вони значною мірою визначають можливість реалізації проекту і вартість майбутньої системи.

У КІС комп'ютер, як частина системи, зазвичай виконує функції управління і повинен працювати в режимі реального часу. Тому при розробці ПЗ необхідно враховувати апаратні засоби, засоби взаємодії з користувачем і середовище оточення. Оскільки багато властивостей ПЗ складної системи проявляють себе тільки тоді, коли вони зібрана повністю і проявляються в робочому режимі, то не врахування цих факторів в сукупності може привести до побудови ненадійного ПЗ. Графік співвідношення надійності ПЗ і апаратури показаний на рис. 1.



Рис. 1. Співвідношення надійності програми і апаратури

Можна виділити три типи системних (програмно-апаратних) компонентів, схильних до відмов:

1) апаратні засоби системи, що відмовляють або через помилки конструювання, або через помилки виготовлення, або через зношування (старіння), або через експлуатацію у важких неприпустимих по ТУ умовах;

2) ПЗ системи, яке може відмовити через помилки в специфікаціях, в архітектурі, в програмному коді;

3) людський фактор, який своїми діями порушує заплановану роботу системи або ініціює незаплановані в ПЗ дії.

Аналіз програмного забезпечення показує, що висока вартість ПЗ це результат його низької надійності.

Типовий розподіл вартості ПЗ слідує: програмування - 10%, проектування – 15%, тестування – 25%, супровід – 50%.

Звідси випливає висновок, що найкращий шлях скорочення вартості ПЗ - в зменшенні вартості його тестування і, головне, супроводу, тобто в підвищенні надійності.

Теорія надійності як наука отримала розвиток стосовно складних технічних систем. Необхідність і корисність контролю технічних компонент систем і систем в цілому, з метою перевірки відповідності їх поточних характеристик заданим, доведені практикою. У цьому плані проведено значну кількість робіт по надійності технічних систем, розроблено безліч моделей забезпечення розумними методами надійності складних систем і їх технічної готовності.

Ці моделі в ряді випадків дозволяють не тільки оцінювати показники надійності і готовності технічних систем і їх компонентів, але і дають можливість прогнозувати значення цих показників на основі накопиченого досвіду. Крім того, ряд моделей дозволяє на основі накопичених даних висловлювати припущення щодо режимів роботи, при яких найбільш часто виявляються відхилення від нормального функціонування, а також про оптимальний підхід до відновлення (ремонт) системи або її компонентів після збою.

Під системою в теорії надійності прийнято розуміти сукупність підсистем або елементів, функціонально об'єднаних відповідно до деякого алгоритму взаємодії при виконанні конкретного завдання в процесі застосування за призначенням. Під це визначення системи повністю підходить програмне забезпечення.

До основних проблем досліджень надійності ПЗ відносяться:

- розробка методів оцінки та прогнозування надійності ПЗ;
- визначення основних факторів, що впливають на надійність ПЗ;
- розробка методів, що забезпечують досягнення заданого рівня надійності ПЗ;
- вдосконалення методів підвищення надійності ПЗ в процесі проектування і експлуатації.

Основна причина помилок в ПЗ - це його складність. Для боротьби зі складністю виділяються дві концепції: незалежність та ієрархічна структура.

В роботі [2] наводиться правило " $n \pm 1$ ". Перевірка правильності фази n проекту повинна здійснюватися проектувальниками (виконавцями) фаз $(n + 1)$ і $(n-1)$. Крім того, в [1] наводиться обґрунтування необхідності якомога більш раннього виявлення помилок проектування ПЗ. Воно полягає в тому, що вартість виправлення помилки з часом зростає, а ймовірність правильно виправити помилку - падає

При цьому ймовірність правильно виправити помилку знаходиться в протиріччі з ймовірністю виявити помилку. Ймовірність виявити помилку зростає з часом при уточненні вимог замовника і під час дослідної експлуатації. У зв'язку з цим важливо вирішити завдання оптимізації часу виявлення помилки при мінімальних витратах на її виправлення

Процес виконання програми з наміром знайти помилку називається тестуванням. Для ПЗ важливим є спроба знайти помилку виконуючи програму в заданому реальному середовищі або при випробуванні (валідації)

Частоти прояву помилок в ПЗ за типами помилок представлені в табл. 2.

Таблиця 2

Частота появи помилок в ПЗ

Тип помилки	Частота появи, %
Не повна або помилкова специфікація	28
Відхилення від специфікації	12
Нехтування правилами програмування	10
Помилкова вибірка даних	10
Помилкова логіка або послідовність операцій	12
Помилкові арифметичні операції	9
Брак часу для вирішення	4
Помилка при обробці переривань	4
Помилка у вихідних даних	3
Неточний запис	8

Як видно з таблиці 2, основна кількість помилок робиться через невірну специфікацію або ТЗ. Ці помилки, в свою чергу, можуть бути розділені на наступні категорії (таблиця 3):

Таблиця 3

Категорії помилок в ПЗ

Причина помилки	Частота появи, %
Помилки в числових значеннях	12
Недостатні вимоги до точності	4
Помилкові символи або знаки	2
Помилки оформлення	15
Неправильний опис або вимога до апаратури	2
Вихідні дані для розробки неповні, неточні або помилкові	52
Двозначність вимог	13

З цих таблиць стає зрозумілим, на що потрібно звертати особливу увагу при проведенні валідації та верифікації ПЗ.

Тестування програми ведеться до тих пір, поки інтенсивність програмних помилок не зменшиться до заздалегідь заданого рівня. Орієнтовно можна виходити з того, що інтенсивність програмних помилок на етапі випробувань повинна бути не більше інтенсивності апаратних відмов.

Програмні і апаратні відмови мають загальні ознаки:

- об'єкт не виконує заданої функції;
- часи до відмов і часи усунення відмов носять випадковий характер;
- методи обробки статистичних даних однакові.

І відмінності:

– апаратна відмова залежить або від часу, або від обсягу виконаної роботи, а програмна відмова - від тієї функції, яку виконує пристрій під керуванням програми (тобто з якою ймовірністю програма вийде на ділянку, яка містить помилку);

– виявлення та усунення апаратної відмови не означає, що така відмова не повториться, а виявлення і усунення програмної помилки означає, що така відмова більше не повториться (але можуть з'явитися нові помилки);

– програмна відмова може ніколи не реалізуватися за даних умов експлуатації програми;

- апаратні відмови поділяють на раптові і поступові.

Програмні відмови виникають, як правило, раптово і по природі своїй не збігаються з раптовими апаратними відмовами, тому що ймовірність їх виникнення не пов'язана з тривалістю роботи системи. Вона пов'язана з умовною ймовірністю того, що програма містить помилку в даній частині програми і ймовірністю того, що пристрій буде працювати під управлінням цієї частини програми.

Якщо апаратна частина жорстко задана і інтенсивність відмов її не змінюється (тільки збільшується в результаті старіння), то ПЗ в процесі експлуатації повинно мати ряд модифікацій які би приводили (в ідеалі) до зменшення інтенсивності відмов. Слід мати на увазі, що ПЗ в складних технічних системах містить найбільшу кількість помилок, а з ростом надійності технічних засобів складе 90% відмов від загального числа

Виділяють 4 групи принципів забезпечення надійності: запобігання помилок, виявлення помилок, виправлення помилок, забезпечення стійкості до помилок.

Для підвищення надійності програмних комплексів необхідно застосовувати різноманітність. Цей метод передбачає реалізацію однієї і тієї ж функції різними алгоритмами і з застосуванням різних засобів розробки. Також пропонується застосовувати глибоко ешелонований захист. Цей метод передбачає застосування багаторівневого захисту з перекриттям захистів різних рівнів. Пропонується застосовувати також пом'якшену деградацію систем, тобто при виході з ладу однієї частини системи, інша частково бере на себе виконання її функцій.

На мінімізацію помилок і збоїв можливі також і інші дії:

- запобігання помилок за рахунок структурного програмування;
- приховування інформації або дозований доступ до даних з боку програмних засобів і об'єктів в об'єктно-орієнтованому програмуванні;
- налагодження;
- стійкість до збоїв;
- обробка виняткових ситуацій (перехоплення помилок, наприклад, розподіл на нуль) і локалізація помилок і збоїв;
- відновлення програми після збою;
- верифікація та валідація (верифікація відповідає на питання, чи правильно і якісно створена програма, а валідація (або атестація) - на питання чи правильно працює програма).

Підвищення надійності ПЗ можливе також за допомогою введення надмірності. Для цього при розробці ПЗ необхідно використовувати метод резервування. Розробляються дві або більше різних по алгоритмам версій програми для вирішення однієї і тієї ж задачі. В даному випадку добре підходить метод, коли одну і ту ж програму пишуть дві незалежні групи програмістів, навіть якщо при цьому вони реалізують один і той же алгоритм. Це дуже ресурсомісткий метод і тому рідко використовується на практиці. Таке ПЗ паралельно виконується в процесі експлуатації.

Надійність ПЗ підвищується також за допомогою застосування різних методів тестування. Повне тестування ПЗ об'єктивно неможливо, тому зазвичай застосовують такі види тестування як:

1. Тестування гілок.
2. Математичний доказ правильності алгоритму розв'язання задачі (в деяких роботах саме в цьому сенсі вживається слово верифікація). Доказ правильності програми за допомогою обчислення предикатів першого порядку не виключає помилки в програмі, так як відноситься до доведення правильності внутрішньої специфікації на конкретному модулі.
3. Символічне тестування (за допомогою спеціально підібраних тестових наборів), ще називається статичним тестуванням. Зручно при локалізації помилки, прояв якої виявлено при конкретному вузькому або строго заданому діапазоні вхідних значень.
4. Динамічне тестування (за допомогою динамічно генеруємих вхідних даних), що зручно при швидкому тестуванні в усьому широкому діапазоні вхідних параметрів.
5. Тестування шляхів виконання програми.
6. Функціональне тестування.

7. Перевірки за часом виконання програми.
8. Перевірка по використанню ресурсів і стресове тестування.

Існує 4 основні складові функціональної надійності програмних систем:

- 1) безвідмовність - властивість програми виконувати свої функції під час експлуатації;
- 2) працездатність - властивість програми коректно (так як очікує користувач) працювати весь заданий період експлуатації;
- 3) безпека - властивість програми бути безпечною для людей і оточуючих систем;
- 4) захищеність - властивість програми протистояти випадковим або навмисним вторгненням в неї.

В цьому випадку високий рівень функціональної надійності може бути досягнутий тільки за рахунок зменшення ефективності роботи програми. Чим інтенсивніше використання ПЗ, тим швидше виявляються в ньому помилки.

Якщо інтенсивне тестування програм протягом досить тривалого часу не приводить до виявлення дефектів або помилок, то створюється відчуття марності подальшого тестування, і програма передається на експлуатацію. Експериментальні дослідження характеристик виявлення помилок в складних програмах дозволило оцінити темп виявлення помилок, при якому складні комплекси програм передаються в регулярну експлуатацію: 0,02 - 0,05 помилок в день на людину, тобто фахівці виявляють лише близько однієї помилки кожні два місяці.

Інтенсивність виявлення помилок нижче 0,001 помилок в день на людину, тобто менше однієї помилки в рік на 3-4 фахівців, може служити еталоном високої якості налагодження і надійності для ПЗ обробки інформації і відповідає дуже високому рівню напруження на відмову $\approx 5 - 10$ тисяч годин.

Висновки. На основі зробленого огляду можна констатувати: на сьогоднішній день відсутнє загальне рішення проблеми надійності ПЗ і є багато приватних рішень, які не враховують такі істотні фактори як інтенсивність внесення і усунення помилок в програмі, час розробки ПЗ.

До числа основних факторів, що впливають на надійність ПЗ віднесені:

- 1) взаємодія ПЗ з зовнішнім середовищем (програмно-апаратні засоби, транслятори, ОС). Цей фактор вносить найменший внесок в надійність ПЗ при сучасному рівні надійності апаратури, ОС і компіляторів;
- 2) взаємодія з людиною (розробником і користувачем) (метрика Холстеда);
- 3) організація ПЗ (проектування, постановка задачі і способи їх досягнення і реалізації) і якість його розробки. Цей фактор робить найбільший внесок в надійність;
- 4) тестування.

Відповідно до цього способу забезпечення і підвищення надійності ПЗ можуть бути наступними:

- 1) Удосконалення технології програмування (наприклад, формальний опис етапів програмування за допомогою мови UML).
- 2) Вибір алгоритмів, які не чутливі до різного роду порушень обчислювального процесу (використання алгоритмічної надмірності).
- 3) Резервування програм - N-версійність програмування.
- 4) Верифікація та валідація програм з подальшим коригуванням.

Жодний з способів не може вважатися достатнім для оцінки надійності. Таким чином, сьогоднішній рівень розуміння проблеми надійності, в основному якісний, але він дозволяє нам розглядати програму як чорний ящик, на вхід якого поступають певні дані із зовнішніми впливами, а на виході формується потік помилок, які усуваються з певним успіхом. Тому на сьогоднішній день актуальним є завдання побудови більш досконалих моделей.

ЛІТЕРАТУРА:

1. А.В. Антонов, А.С. Степанянц. Методы анализа надежности (безошибочности) программного обеспечения программно-технических средств // Труды II Междунар. конф. "Идентификация систем и задачи управления" (SICPRO-2003). – С.924-942.

2. Ханджян А.О. Анализ современного состояния разработки надежного программного обеспечения //Естественные и технические науки. – М., 2005. – №2. – С. 220 – 227.

3. Лисс В.А. Математические модели надежности программного обеспечения распределенных систем //Известия СПбГТУ "ЛЭТИ". Сер. "Информатика, управление и компьютерные технологии". – 2005. – Вып.2. – С.26-32.

REFERENCES:

1. A.V. Antonov, A.S. Stepanyanc. Metody analiza nadezhnosti (bezoshibochnosti) programmno obespecheniya programmno-tekhnicheskikh sredstv // Trudy II Mezhdunar. konf. "Identifikaciya sistem i zadachi upravleniya" (SICPRO-2003). – S.924-942.

2. Handzhyan A.O. Analiz sovremennogo sostoyaniya razrabotki nadezhnogo programmno obespecheniya //Estestvennye i tekhnicheskie nauki. – М., 2005. – №2. – S. 220 – 227.

3. Liss V.A. Matematicheskie modeli nadezhnosti programmno obespecheniya raspredelennyh sistem //Izvestiya SPbGTU "LEHTI". Ser. "Informatika, upravlenie i komp'yuternye tekhnologii". – 2005. – Выр.2. – S.26-32.

Рецензент: д.т.н., доц. Оксіюк О.Г., Київський національний університет імені Тараса Шевченка

Бабий И.Н., Ленков А.С., к.т.н., доц. Огневой А.В.
**АНАЛИЗ ПРОГРАММНО-АЛГОРИТМИЧЕСКИХ СРЕДСТВ ПРОВЕДЕНИЯ ОЦЕНКИ
НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

В статье проведен анализ программно-алгоритмических средств проведения оценки надежности программного обеспечения (ПО), что позволяет проводить расчет характеристик надежности ПО (таких как, время наработки до отказа, коэффициент готовности, вероятность отказа. Рассматриваются методы тестирования, позволяющие повысить надежность ПЗ. Описаны основные составляющие функциональной надежности программных систем. Приведены результаты экспериментальных исследований характеристик обнаружения ошибок в сложных программах . Определены основные факторы, влияющие на надежность ПЗ. Предложены способы обеспечения и повышения надежности ПО.

Ключевые слова: корпоративные информационные системы, программное обеспечение, надежность.

Babiy I.M., Lenkov O.S., Ph.D. Ognjevuj O.V.
**ANALYSIS SOFTWARE AND ALGORITHMIC MEANS OF ASSESSING THE
RELIABILITY OF SOFTWARE**

The article analyzes the software and algorithmic means of assessing the reliability of software (software) that allows you to calculate the characteristics of software reliability (such as a time to failure, availability factor, the probability of failure. Methods of testing that improve reliability PZ. Opysani main components of functional reliability of software systems. The results of experimental studies the characteristics of detecting errors in complex programs. Osnovnyi identified factors affecting the reliability PZ. Zaproponovani ways to ensure and improve the reliability of software.

Keywords: corporate information systems, software reliability.