

АНАЛІЗ НАДІЙНОСТІ ФУНКЦІОНУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧИХ СИСТЕМ

В роботі розглядаються якість і надійність програмного забезпечення. Наводяться показники надійності програмного забезпечення, описуються деякі підходи до оцінки надійності програмного забезпечення і аналізу видів, наслідків та критичності помилок у програмному забезпеченні. Пропонується методика розрахунку коефіцієнта готовності та середнього напрацювання на відмову програмного забезпечення інформаційно-управляючих систем. Методика використовує теорію марковських процесів. Робота може бути застосована для проектної оцінки та аналізу надійності інформаційно-управляючих систем.

Ключові слова показники якості, коефіцієнт готовності, якість програмного забезпечення, напрацювання на відмову, марковський процес, математична модель.

Y.B. KOVALENKO, L.P. RYBALKA, M. V. BURLAKA

National Aviation University

ANALYSIS OF THE RELIABILITY OF THE FUNCTIONING OF THE SOFTWARE OF INFORMATION AND MANAGEMENT SYSTEMS

Abstract – The paper considers the quality and reliability of software. Leading Different indicators of software reliability, describes some approaches to assessing the reliability of software and analysis of the types, consequences and criticality of errors in the software. The method of calculating the availability factor and the mean time between failures of the software of information-control systems is proposed. The technique uses the theory of Markov processes. The work can be applied for the project evaluation and analysis of the reliability of information management systems. The analysis of the function of the software for the protection of information and control systems have been performed, and the function of the intensity has been shown to demonstrate the pardon and assessment of the parameters for the observant denim. The approaches to the excellent hope of the software analysis are determined, the consequences that the criticality of the software on providing true-control systems. Method of developing software for the preparation of middle gates for information and control software systems has been hammered, and it has given the team the opportunity to prepare a medium-sized channel for the software support of information management systems.

Keywords: quality indicators, availability factor, software quality, time between failures, markov process, mathematical model.

Вступ

Надійність програмного забезпечення інформаційно-управляючих систем (ІУС) є однією з основних характеристик його якості [1, 3]. Саме від надійності функціонування програмного забезпечення (ПЗ) ІУС залежить повнота виконання завдань. Свідченням цього є значне число фатальних ситуацій як вітчизняних, так і зарубіжних об'єктів підвищеної небезпеки. В міру ускладнення об'єктів управління, збільшення їх функцій проблема забезпечення необхідного рівня надійності ПЗ стає актуальною.

Відомо чотири основних напрямки забезпечення надійності ПЗ [1, 2]. Перший напрямок – попередження помилок – засновано на мінімізації або виключенні помилок. Друге – виявлення помилок – реалізується за рахунок здатності ПЗ самостійно виявляти помилки. Третє – виправлення помилок і їх наслідків. Четверте – забезпечення стійкості до помилок – засновано на унікальній властивості ПЗ, що полягає в здатності ПЗ функціонувати при наявності помилки, тобто його відмовостійкості.

Одним з найпоширеніших підходів до реалізації програмної відмовостійкості є методологія надмірності. У теорії надійності відомі декілька форм надмірності: апаратна; інформаційна, навантажувальна, функціональна, часова, комбінована.

Для більшості ІУС перехід у критичний стан, не допустимий для їх функціонування у режимах зберігання або транспортування, не передбачено. Тому зупинимося на оцінці двох властивостей надійності: безвідмовність і ремонтпридатність. Комплексною характеристикою безвідмовності і ремонтпридатності системи є коефіцієнт готовності. Аналітичний підхід до оцінки показників надійності базується на незалежному обліку потоку відмов за рахунок різних чинників, що впливають на функціонування ІУС. На початкових етапах розвитку теорії надійності основна увага приділялася розробці методик, що регламентують порядок збору та обробки статистичних даних про відмови виробів.

При цьому удосконалювалися ймовірно-статистичні методи дослідження, в тому числі, визначення адекватних законів розподілу показників безвідмовності, довговічності, ремонтпридатності і зберігання. Пізніше почали розроблятися методи розрахунку і випробувань виробів з урахуванням випадкового характеру відмов.

Основними організаторами робіт з розвитку теорії надійності, в тому числі прикладної, стали академік А.І. Берг, Я.М. Сорін, Б.Р. Левін і Я.Б. Шор. Найбільший внесок в даній області був зроблений Б.В. Гнеденко, А.Д. Соловйовим, Ю. К. Беляєвим, Х.Б. Кордонським, Н.А. Шішоноковим, В.Ф. Репкін, Л.Л. Барвінським, М.А. Ястребенецьким, Т.А. Голінкевічем, І.А. Ушаковим.

Проблеми надійності сучасних інформаційно-обчислювальних систем, автоматизованих систем управління розглядаються в роботах А.Є. Олександровича, Ю.В. Бородакія, В.О. Чаканова, Ю.Н. Федорова та ін.

Слід зазначити, що питання надійності інтенсивно досліджувалися і за кордоном, наприклад, такими фахівцями, як Д. Нейман, А. Пірс, К. Барлоу, С. Прошан та ін.

Висока надійність технічних систем, в тому числі і систем управління, визначається не тільки проектуванням та виготовленням включаються в них технічних засобів, а й раціональною експлуатацією, тому питань експлуатаційної надійності, проведення випробувань та обробки інформації про експлуатацію пристроїв і систем, завжди приділялася велика увага.

Отже, при проектуванні сучасних ІУС важливе місце займають проектна оцінка та аналіз їх надійності. Визначивши необхідні аспекти надійності ІУС та її стійкості у відношенні відмов, на етапі проектування можна підвищити рівень живучості системи та зменшити витрати на її розгортання.

Постановка завдання

Проведені дослідження ставили за мету визначити якість і надійність функціонування програмного забезпечення інформаційно-управляючих систем. А також виокремлено підходи до оцінки надійності програмного забезпечення і аналізу видів, наслідків та критичності помилок у ПЗ ІУС. Що дало змогу розрахувати коефіцієнт готовності та середнього напрацювання на відмову програмного забезпечення інформаційно-управляючих систем.

Результати дослідження

Розглянули концепцію мультіверсійного програмування, яка була запропонована А. Авіженісом, як одного із способів програмної надмірності [7, 8]. Він визначив мультіверсійне програмування як незалежну генерацію функціонально еквівалентних програм, що означає ізольоване створення версій програм незалежними фахівцями з використанням різних специфікацій, алгоритмів і мов програмування [5]. Дана концепція дозволяє змінювати або замінювати програму, що відмовила під час роботи, іншою без шкоди для ІУС [4].

Якщо в ПЗ передбачається система діагностування стану програм, то, як правило, власне її надійність вважається ідеальною. Як показник надійності ПЗ приймається ймовірність знаходження системи в працездатному стані – ймовірність безвідмовної роботи [6].

При побудові математичних моделей надійності функціонування ПЗ ІУС були прийняті наступні допущення:

- відмови програм виявляються системою контролю миттєво і достовірно;
- потоки відмов і відновлень Пуассоновські;
- під час перемикавання резервна програма не відмовляє;
- перемикач програм з основною на резервну не ідеальний;
- відновлення основної програми починається тільки після підключення резервної;
- система з відмовного стану не відновлюється.

Оцінка якості програмного забезпечення може проводитися з двох позицій: з позиції позитивної ефективності і безпосередньої адекватності їх характеристик за призначенням, цілям створення і застосування, а також з негативної позиції можливого при цьому збитку – ризику від використання ПЗ або системи.

Показники якості переважно відображають позитивний ефект від застосування програмного забезпечення і основне завдання при розробці проекту полягає в забезпеченні високих значень якості. Ризики характеризують можливі негативні наслідки, що проявилися в ході експлуатації помилок, або збиток для користувача при застосуванні та функціонуванні ПЗ. Згідно [11] якість програмного забезпечення це весь обсяг ознак і характеристик програмного забезпечення, яке відноситься до її здатності задовольняти встановленим або передбачуваним потребам. Якість ПЗ оцінюється наступними характеристиками:

- Функціональні можливості. Набір атрибутів, що відносяться до суті набору функцій та їх конкретних властивостей. Функціями є ті, які реалізують встановлені або передбачувані потреби.
- Надійність. Набір атрибутів, що відносяться до здатності ПЗ зберігати свій рівень якості функціонування при встановлених умовах за встановлений період часу.
- Практичність. Набір атрибутів, які відносяться до обсягу робіт, необхідних для використання та індивідуальної оцінки такого використання певним і передбачуваним колом користувачів.
- Ефективність. Набір атрибутів, що відносяться до співвідношення між рівнем якості функціонування ПЗ та обсягом використовуваних ресурсів при встановлених умовах.
- Підтримка. Набір атрибутів, які відносяться до обсягу робіт, необхідних для проведення конкретних модифікацій.
- Мобільність. Набір атрибутів, що відносяться до здатності ПЗ бути перенесеним з одного оточення в інше.

За визначенням, встановленим в [10], надійність – властивість об'єкта виконувати задані функції, зберігаючи в часі значення встановлених експлуатаційних показників у заданих межах, які відповідають заданим режимам та умовам використання, технічного обслуговування, ремонту, зберігання і транспортування [14].

При цьому надійність є комплексною властивістю, яка в залежності від призначення об'єкта та умов його застосування може включати безвідмовність, довговічність, ремонтпридатність і збереженість або певні поєднання цих властивостей (рис. 1).

Оскільки ПЗ в процесі експлуатації не зношується, його поломка і ремонт в загальноприйнятому сенсі не проводиться, то надійність програмного забезпечення має сенс характеризувати тільки з точки зору безвідмовності його функціонування і можливості відновлення функціонування після відмов викликаних проявами помилок [15].

В [11] надійність ПЗ пропонується характеризувати за допомогою наступних характеристик: стабільність, стійкість і відновлюваність.

В цьому випадку стабільність і стійкість характеризують безвідмовність ПЗ, а відновлюваність – можливість відновлення функціонування ПЗ після його відмови. Для кількісної оцінки надійності ПЗ необхідно визначити показники надійності для кожної властивості і методику їх визначення. Для оцінки стабільності ПЗ можливе використання показників, що характеризують безвідмовність технічних пристроїв [10] (рис. 2).

У більшості випадків потік програмних помилок може бути описаний негомогенним процесом Пуассона [12]. Це означає, що програмні помилки відбуваються в статистично незалежні моменти часу, напрацювання підкоряються експоненціальному розподілу, а інтенсивність прояву помилок змінюється в часі. Зазвичай використовують спадну інтенсивність прояву помилок. Це означає, що помилки, як тільки вони виявлені, ефективно усуваються без введення нових помилок.

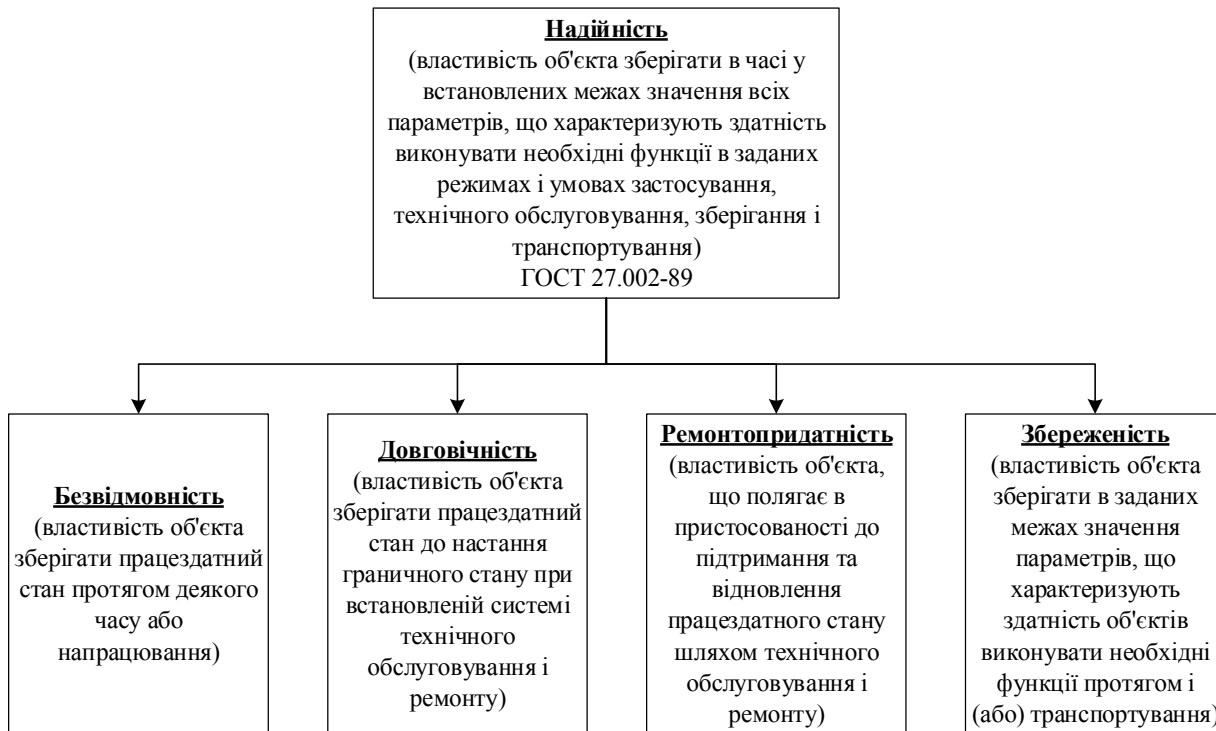


Рис. 1. Надійність згідно ГОСТу 27.002 – 89

Головна мета аналізу надійності ПЗ полягає в тому, щоб визначити форму функції інтенсивності прояву помилок і оцінити її параметри по спостережуваним даним. Як тільки функція інтенсивності прояву помилок визначена, можуть бути знайдені такі показники надійності як: загальна кількість помилок; кількість помилок, що залишаються; час до прояву наступної помилки; ймовірність безпомилкової роботи; інтенсивність прояву помилок; залишковий час випробувань (до прийняття рішення); максимальна кількість помилок (щодо терміну служби).

При цьому слід розрізняти поняття помилка і відмова. Відносно до надійності ПЗ помилка – це помилка або спотворення коду програми, ненавмисно внесені в неї під час розробки, які в ході функціонування цієї програми можуть викликати відмову або зниження ефективності функціонування. Під відмовою в загальному випадку розуміють подію, що полягає в порушенні працездатності об'єкта [10].

Стан об'єкта, при якому значення всіх параметрів, що характеризують здатність виконувати задані функції, відповідають вимогам нормативно-технічної і проектної документації – називається працездатним. При цьому критерії відмов, як ознаки або сукупність ознак порушення працездатного стану ПЗ, повинні визначатися виходячи з його призначення в нормативно-технічній і проектній документації. У загальному випадку відмову ПЗ можна визначити як: припинення функціонування програми нормального ходу її виконання, зациклення на час, що перевищує заданий поріг; припинення функціонування програми, тобто спотворення нормального ходу її виконання, зациклення на час що не перевищує заданий поріг, але з втратою всіх або частини оброблюваних даних; припинення функціонування програми, що привело до перезавантаження ПК, на якому функціонує ПЗ. При цьому, виходячи з [10], всі відмови в ПЗ слід трактуємо як відмови, що самоусуваються або одноразові відмови, які усуваються незначним втручанням оператора, оскільки відновлення працездатного стану ПЗ може статися без втручання оператора, перезавантаження ПК

не потрібно, або при участі оператора або експлуатуючого персоналу. Наведені вище критерії відмов призводять до необхідності аналізу тимчасових характеристик функціонування програми і динамічних характеристик споживачів даних, отриманих в ході функціонування ПЗ. Часовий проміжок перерви нормальної видачі інформації та втрати працездатності, яку слід розглядати як зону відмови, систем ширше, ніж більш інертний об'єкт знаходиться під впливом даних, отриманих в ході роботи програми. Граничний час відновлення працездатного стану системи, при перевищенні якого слід фіксувати відмову, близько до періоду вирішення завдань для підготовки інформації відповідному споживачу.

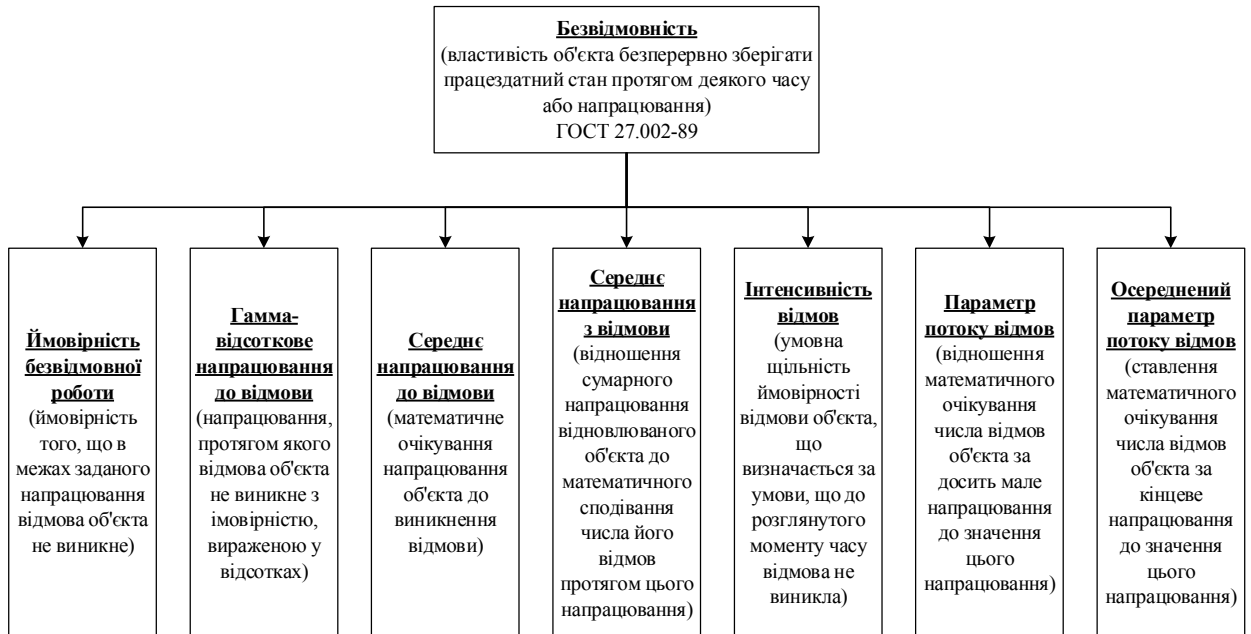


Рис. 2. Показники безвідмовності

Для будь-якого споживача даних існує допустимий час відсутності даних від програми, при якому його характеристики знаходяться в допустимих межах. Виходячи з цього часу, можна встановити межі тимчасової зони, яка розділяє працездатний та непрацездатний стан ПЗ і дозволяє використовувати дані критерії відмов.

З наведеного вище визначення програмної помилки з точки зору надійності, можна зробити висновок про те, що помилки, при їх прояві, не завжди викликають відмову ПЗ і кожен помилку можна характеризувати умовною ймовірністю виникнення відмови при прояві цієї помилки. Слід також зазначити, що сама по собі наявність помилки у вихідному коді не визначає надійність програми до тих пір, поки не відбудеться прояв цієї помилки, тому користуватися для оцінки надійності ПЗ тільки показниками, що характеризують загальну кількість помилок в програмі, кількість помилок, що залишилась і максимальну кількість помилок, не можна [9].

В [13] стабільність пропонується оцінювати ймовірністю безвідмовної роботи, яка оцінюється виходячи з моделі відносної частоти, при цьому застосування її обмежено періодом експлуатації ПЗ, що не завжди прийнятно, оскільки надійність об'єкта, як правило, необхідно оцінювати не тільки в процесі його експлуатації, але і до початку експлуатації цього об'єкта. Обмеження моделі відносної частоти викликано тим, що в цій моделі не враховуються процеси тестування і налагодження, а саме те, що при виникненні відмови ПЗ, помилка, що викликала цю відмову, виправляється.

Найбільш прийнятними показниками, що характеризують стабільність ПЗ, видаються показники подібні до показників безвідмовності технічних систем: ймовірність безвідмовної роботи, інтенсивність відмов і середній час напрацювання на відмову. Ці показники взаємопов'язані і, знаючи один з них, можна визначити інші [10]. При визначенні цих показників в більшості випадків можна виходити з моделі надійності, яка передбачає, що інтенсивність прояву помилок зменшується в міру виправлення цих помилок, час між проявами помилок розподілено експоненціально, а інтенсивність прояву помилок постійна між двома сусідніми проявами помилок. Застосування такої моделі надійності ПЗ дозволить оцінити надійність ПЗ під час тестування і налагодження.

Стійкість, як властивість або сукупність властивостей ПЗ, що характеризують його можливість підтримувати прийнятний рівень функціонування при проявах помилок в ньому, можна оцінювати умовною ймовірністю безвідмовної роботи при прояві помилки.

Показники надійності ПЗ в значній мірі адекватні аналогічним характеристикам, прийнятих для інших технічних систем. Найбільш широко використовується показник напрацювання на відмову. Напрацювання на відмову – це відношення сумарного напрацювання об'єкта до математичного сподівання числа його відмов протягом цього напрацювання. Для ПЗ використання даного показника утруднено, в силу особливостей тестування і налагодження ПЗ. Тому доцільно використовувати показник середнього

напрацювання до відмови – математичного очікування часу функціонування ПЗ до відмови. При використанні моделі надійності ПЗ передбачає експоненціальний розподіл часу між відмовами, середній час напрацювання до відмови дорівнює величині зворотної інтенсивності відмов. Інтенсивність відмов можна оцінити виходячи з оцінок стабільності і стійкості ПЗ.

Розглянемо математичну модель оцінки надійності ІУС. Для цього ми використовуємо динамічні моделі оцінювання надійності ПЗ, що ґрунтуються на випадковому процесі Пуассона, який можна описати так: нехай $\{G(t), t \geq 0\}$ – неоднорідний пуассонівський процес з функцією інтенсивності $\lambda(t)$, який являє собою загальну кількість відмов ПЗ за час його використання t . Тоді кількість відмов ПЗ в інтервалі $(0, t)$ має розподіл Пуассона, тобто:

$$P\{G(t) = k\} = \frac{(\mu(t))^k}{k!} \exp(-\mu(t)), \quad (1)$$

$$k = 0, 1, \dots, N \text{ визначається емпірично,}$$

де $\mu(t) = \int_0^t \lambda(u) du$ – середня кількість відмов ПЗ за час t

Оскільки функція $\mu(t)$ характеризує ступінь підвищення надійності ПЗ, то її зазвичай називають функцією зростання надійності. Вона описує взаємозв'язок між кількістю виявлених відмов ПЗ і часом, витраченим на тестування та усунення дефектів.

Введемо такі позначення:

ν – кількість дефектів у ПС на початку тестування (дорівнює кількості відмов, які можуть виникнути при $t \rightarrow \infty$), β – коефіцієнт пропорційності, що дорівнює швидкості виявлення одного дефекту.

До основних моделей надійності цього типу належать:

1. експоненційна, функція зростання надійності якої задається формулою

$$\mu(t) = \nu(1 - \exp(-\beta t)), \quad \nu, \beta > 0, \quad (2)$$

2. логарифмічна, функція зростання надійності для якої

$$\mu(t) = \frac{1}{\theta} \ln(\lambda_0 \theta t + 1), \quad \lambda_0, \theta > 0, \quad (3)$$

де λ_0 – початкова інтенсивність відмов, θ – коефіцієнт, який характеризує зниження інтенсивності відмов з кожним усуненням дефектом.

3. сповільненого S-подібного зростання надійності, для якої

$$\mu(t) = \nu(1 - (1 + \beta t) \exp(-\beta t)), \quad \nu, \beta > 0, \quad (4)$$

4. S-подібного зростання надійності з перегинами, яка має вигляд

$$\mu(t) = \nu(1 - \exp(-\beta t)) / (1 + \psi \exp(-\beta t)), \quad (5)$$

де ψ – параметр перегину, який визначається як: $\psi(r) = (1 - r) / r$, $0 \leq r \leq 1$, де r – частота перегинів, що являє собою відношення виявлених дефектів до загальної їх кількості.

Отже, розглянемо випадковий процес із дискретними станами і неперервним часом, що буде протікати у такій системі, можна назвати марковським, тобто для будь-якого моменту часу t умовні імовірності всіх станів системи S у майбутньому (при $t > t_0$) залежать тільки від того, у якому стані S_j знаходиться система S у теперішній час (при $t = t_0$), але не залежать від того, коли і яким чином вона перейшла у цей стан (тобто, якими були стани системи у минулому (при $t < t_0$)) [16]. Застосуємо до системи марковську модель розрахунку надійності.

Визначимо показники надійності елементів.

Середнє напрацювання елемента на відмову – середньостатистична тривалість безвідмовної роботи елемента:

$$M_{відм} = \sum_{i=1}^N \frac{t_{imp}}{N}, \quad (6)$$

де t_{imp} – тривалість i -го інтервалу безперервної роботи системи, N – кількість відмов. Інтенсивність відмов:

$$\lambda(t) = \lim_{\Delta t \rightarrow 0} \frac{G[X(t, \Delta t)]}{\Delta t}, \quad (7)$$

де $G[X(t, \Delta t)]$ – математичне очікування випадкової кількості відмов, які припадають на елементарний проміжок часу $(t, t + \Delta t)$.

Інтенсивність відмов обернено пропорційна середньому напрацюванню на відмову:

$$\lambda = \frac{1}{M_{\text{відм}}} \quad (8)$$

Середня тривалість відновлення елемента:

$$M_{\text{відн}} = \sum_{i=1}^N \frac{t_{\text{inp}}}{N}, \quad (9)$$

де t_{inp} – тривалість простою системи, викликаного i -ю відмовою, N – кількість відмов.

Інтенсивність відновлень – обернена середній тривалості відновлення величина:

$$\mu = \frac{1}{M_{\text{відн}}} \quad (10)$$

Коефіцієнт готовності – імовірність знаходження елемента в робочому стані у довільний момент часу:

$$K_g = \frac{M_{\text{відм}}}{M_{\text{відм}} + M_{\text{відн}}} = \frac{\mu}{\mu + \lambda} \quad (11)$$

За умови, що елементи у системі небезвідмовні та відновлювані, можна отримати ергодичну множину станів системи. Застосувавши до системи марковську модель, отримаємо граф із 2^n станів системи із n елементів.

Складемо за графом систему диференціальних рівнянь Колмогорова. Для кожного стану S_i системи S запишемо рівняння вигляду

$$\frac{dp_i(t)}{dt} = \sum_{j=1}^{2^n} p_j(t) \lambda_{ji}(t) - p_i(t) \sum_{j=1}^{2^n} \lambda_{ij}(t), \quad (12)$$

де $p_i(t)$ – імовірність знаходження системи S у стані S_i у часі, $\lambda_{ji}(\lambda_{ij})$ – інтенсивність потоку подій, що ведуть до переходу зі стану $S_j(S_i)$ у стан $S_i(S_j)$, перша сума для всіх j таких, що $\lambda_{ji} \neq 0$, тобто можливий безпосередній перехід із S_j у S_i , друга сума для всіх j таких, що $\lambda_{ij} \neq 0$, тобто можливий безпосередній перехід із S_i у S_j . Щоб провести розрахунки необхідне ієрархічне розбиття системи на підсистеми. Метою даного розбиття є виділення логічних з точки зору стійкості до відмов фрагментів системи разом із зменшенням обсягу обчислень.

Система може бути представлена диз'юнктивним сімейством підсистем, кожна із яких сама по собі буде вершиною чи вузлом графа і характеризуватиметься власними показниками надійності.

На даному етапі слід виділяти фрагменти таким чином, щоб можна було однозначно визначити справність кожного із них.

Множина станів системи розбивається на дві підмножини справних станів і тих, у яких система несправна у залежності від критерію працездатності. Для спрощення оцінки працездатності системи на множині усіх станів, а також із метою спрощення розрахунку коефіцієнта готовності системи можна виділити множини поглинаючих робочих станів та поглинаючих неробочих станів системи. Назвемо множиною поглинаючих неробочих станів таку підмножину множини неробочих станів системи, де відновлення кожного неробочого елемента веде до безпосереднього переходу у робочий стан, а відмова кожного робочого елемента веде до переходу у неробочий стан. Аналогічно множиною поглинаючих робочих станів будемо називати таку підмножину множини робочих станів системи, де відмова кожного робочого елемента веде до безпосереднього переходу у неробочий стан, а відновлення кожного неробочого елемента веде до переходу у робочий стан. Знаючи вказані вище множини станів, можна встановити працездатність будь-якого стану системи без безпосередньої перевірки критерію працездатності системи.

На основі розрахунків системи S_i (п. 3), вважаючи, що вона є підсистемою будь-якої системи S , можна отримати імовірність знаходження системи S , у будь-якому стані як добуток імовірностей знаходження у відповідних станах окремих елементів системи:

$$p_i(t) = \prod_{j=1}^n p_{ji}(t), \quad (13)$$

де $l \in \{1; 2\}$, p_{j1} – імовірність знаходження j -ого елемента системи у робочому стані (K_2), p_{j2} – імовірність знаходження j -ого елемента системи у неробочому стані ($1 - K_2$).

Індекс l вибирається у залежності від того, у якому стані входить j -й елемент у даний стан s системи S .

На даному етапі прискорити розрахунок можна, звернувшись за допомогою до визначених вище множин поглинаючих робочих та неробочих станів, знайдемо імовірність перебування системи у робочому стані як суму імовірностей перебування системи у відповідних станах S_i , тобто множина станів, у яких ми визначили систему справною:

$$K_2 = P(t) = \sum_{i=1}^{2^n} p_i(t) k, \quad (14)$$

де $k \in \{0; 1\}$ – коефіцієнт справності системи у стані S_i .

Таким чином може бути розрахована імовірність знаходження у справному стані окремого фрагмента системи.

Для розрахунку середньої тривалості напрацювання системи на відмову застосуємо поняття множини граничних робочих станів. Назвемо множиною граничних робочих станів таку підмножину множини робочих станів системи, де із кожного стану можливий безпосередній перехід у неробочий (робочий) стан системи.

Вважаючи систему справною, можна знайти інтенсивність відмов як добуток сум інтенсивностей переходу системи із робочого стану у неробочий на імовірність знаходження системи у кожному із множини граничних робочих станів:

$$\lambda = \frac{1}{K_2} \sum_{i=1}^{2^n} p_i(t) k_{pi} \sum_{j=1}^{2^n} \lambda_{ij} k_{ij} \quad \forall \lambda_{ij} \neq 0, \quad (15)$$

де k_{ij} – коефіцієнт належності стану S_i множині граничних робочих станів, k_{ij} – коефіцієнт належності стану S_j множині граничних неробочих станів.

Середню тривалість напрацювання на відмову знайдемо із формули (8).

Аналогічно для середньої інтенсивності відновлень системи:

$$\mu = \frac{1}{1 - K_2} \sum_{j=1}^{2^n} p_j(t) k_{ij} \sum_{i=1}^{2^n} \mu_{ji} k_{pi} \quad \forall \mu_{ji} \neq 0. \quad (16)$$

Середню тривалість відновлення знайдемо із формули (10).

Висновки

Проведено аналіз надійності функціонування програмного забезпечення інформаційно-управляючих систем, визначено функції інтенсивності прояву помилок і оцінено їх параметри по спостережуванім даним. Визначено підходи до оцінки надійності програмного забезпечення і аналізу видів, наслідків та критичності помилок програмного забезпечення інформаційно-управляючих систем. Запропонована методика розрахунку коефіцієнта готовності та середнього напрацювання на відмову програмного забезпечення інформаційно-управляючих систем, що дало змогу розрахувати коефіцієнт готовності та середнього напрацювання на відмову програмного забезпечення інформаційно-управляючих систем.

Література

1. Боэм Б., Браун Дж., Каспар Х. Характеристики качества программного обеспечения. М.: Мир, 1981.
2. Гецци К., Джазайери М., Мандриоли Д. Основы инженерии программного обеспечения. 2-е изд.: Пер. с англ. СПб.: БХВ - Петербург, 2005. 832 с.
3. Липаев В.В. Системное проектирование сложных программных средств для информационных систем. М: СИНТЕГ, 2002. 268 с.
4. Половко А.М., Гуров С.В. Основы теории надежности. 2-е изд. перераб. и доп. СПб.: БХВ Петербург, 2006.
5. Соммервилл И. Инженерия программного обеспечения. 6-е изд.: Пер. с англ. М: Издательский дом «Вильямс», 2002.
6. Черкесов Г.Н. Надежность аппаратно-программных комплексов. Учебное пособие. СПб.: Питер, 2005. 479 с.
7. Avizienis A. The N-Version approach to fault-tolerant software. IEEE Trans. on Software Engineering. 1985. Vol. SE-11, №12. P. 1491-1501.

8. Avizienis A. A methodology of N-version programming. In: Software Engineering Tolerance. Chichester: John Wiley and Sons, 1995. P. 23-69.
9. Липаев В. В. Программная инженерия. Методологические основы. М.: ТЕИС, 2006.
10. ГОСТ 27.002–89. Надежность в технике. Основные понятия. Термины и определения. М.: Издательство стандартов, 1990.
11. ГОСТ Р ИСО/МЭК 9126–93. Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению. М.: Издательство стандартов, 1994.
12. ГОСТ 51901.5–2005. Менеджмент риска. Руководство по применению методов анализа надежности. М.: Издательство стандартов, 2007.
13. ГОСТ 28195–89. Оценка качества программных средств. Общие положения. М.: Издательство стандартов, 1989.
14. ГОСТ 27.310–95. Надежность в технике. Анализ видов, последствий и критичности отказов. М.: Издательство стандартов, 1995.
15. ГОСТ 51901.12–2007. Менеджмент риска. Метод анализа видов и последствий отказов. М.: Издательство стандартов, 2007.
16. Вентцель Е.С., Овчаров Л.А. Теория случайных процессов и её инженерные приложения. М.: Наука, 1991. 384 с.

References

1. Boem B., Braun J., Kaspar X. Charakteristiki kachestva programmnoho obespecheniya. M.: Mir, 1981.
2. Geczy K., Dzhazajery M., Mandrioli D. Osnovy inzhenerii programmnoho obespecheniya. 2 izd.: Per. s angl. SPb.: BXV Peterburg, 2005. 832 p.
3. Lipaev V. V. Systemnoe proektyrovanie slozhnykh programmnykh sredstv dlya informacionnykh sistem. M: SINTEG, 2002. 268 p.
4. Polovko A.M., Gurov S.V. Osnovy teorii nadezhnosti. 2 izd. pererab. i dop. SPb.: BXV Peterburg, 2006.
5. Sommerville I. Inzheneriya programmnoho obespecheniya. 6 izd.: Per. s angl. M: Izdatelskiy dom «Viliyams», 2002.
6. Cherkosov G.N. Nadezhnost apparatno-programmnykh kompleksov. Uchebnoe posobiye. SPb.: Piter, 2005. 479 p.
7. Avizienis A. The N-Version approach to fault-tolerant software. IEEE Trans. on Software Engineering. 1985. Vol. SE-11, №12. P. 1491-1501.
8. Avizienis A. A methodology of N-version programming. In: Software Engineering Tolerance. Chichester: John Wiley and Sons, 1995. P. 23-69.
9. Lipaev V. V. Programmная инженерия. Metodologicheskiye osnovy. M.: TEIS, 2006.
10. GOST 27.002–89. Nadezhnost v tehnikе. Osnovnyye ponyatiya. Terminy i opredeleniya. M.: Izdatelstvo standartov, 1990.
11. GOST R ISO/MEK 9126–93. Informacionnaya tehnologiya. Ocenka programmnoy produkci. Charakteristiki kachestva i rukovodstva po ih primeneniyu. M.: Izdatelstvo standartov, 1994.
12. GOST 51901.5–2005. Menedzhment riska. Rukovodstvo po primeneniyu metodov analiza nadezhnosti. M.: Izdatelstvo standartov, 2007.
13. GOST 28195–89. Ocenka kachestva programmnykh sredstv. Obshie polozheniya. M.: Izdatelstvo standartov, 1989.
14. GOST 27.310–95. Nadezhnost v tehnikе. Analiz vidov, posledstvij i kritichnosti otkazov. M.: Izdatelstvo standartov, 1995.
15. GOST 51901.12–2007. Menedzhment riska. Metod analiza vidov i posledstvij otkazov. M.: Izdatelstvo standartov, 2007.
1. *Ventcel E.S., Ovcharov L.A. Teoriya sluchajnykh processov i inzhenernye prilozheniya. M.: Nauka, 1991. 384 p.*

Рецензія/Peer review : 21.11.2017 p.

Надрукована/Printed : 13.01.2018 p.

Стаття рецензована редакційною колегією