УДК 004.891

д.т.н. с.н.с. **Селюков О.В.** (ТОВ «Укрспецконсалтинг»)
к.т.н., доц. **Огнєвий О.В.** (ХмНУ**)**
к.військ., доц. **Осипа В.О.** (ВІКНУ)
Бабій І.М. (ХмНУ)

РОЗРОБКА МОДЕЛЕЙ НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ТЕХНОЛОГІЇ ТИПУ «КЛІЄНТ-СЕРВЕР»

У статті проведено аналіз моделей надійності програмного забезпечення (ПЗ). Найбільш відомих моделей надійності ПЗ сьогодні існує близько десятка, тому в цій роботі вони згруповані за ознаками.

Розглянуто комп'ютерні системи з централізованою архітектурою додатків файл-сервер та клієнт-сервер.

Особливу увагу приділено клієнт-серверній архітектурі. Існують різні класифікації, але однією з найпоширеніших є використання чотирьох моделей технології «Клієнт-сервер»: модель файлового серверу (File Server - FS), модель віддаленого доступу до даних (Remote Data Access - RDA), модель сервера БД (Data Base Server - DBS), модель сервера додатків (Application Server - AS) .

З метою оцінки надійності для систем типу "клієнт-сервер" на етапі тестування визначено такі важливі характеристики функціонування програмного комплексу, як: розрахунок поточного часу напрацювання повністю, розрахунок середнього часу напрацювання повністю за весь час моделювання роботи системи, розрахунок імовірності відмови ПЗ за одиницю часу, розрахунок коефіцієнта готовності.

Ключові слова: надійність програмного забезпечення, моделі надійності, клієнт-серверна архітектура, програма-клієнт, область визначення вхідних даних.

Вступ. Прогнозування надійності програмного забезпечення (ПЗ) в процесі його експлуатації здійснюється на основі математичних моделей надійності програм.

В роботі [1] приведені імовірнісні моделі надійності. Теорія надійності для апаратного забезпечення розвинена досить добре, і є також можливість застосувати її і до надійності ПЗ. У цих моделях знаходиться число помилок, що залишилися в програмі. Це необхідно знати для завершення процесу тестування, і оцінки вартості супроводу, яка пропорційна кількості помилок, що залишилися в програмі. Також ці моделі дозволяють знаходити надійність програми, яка розуміється як вірогідність, що програма функціонуватиме без помилок впродовж заданого інтервалу часу, а також - середній час між відмовами програми.

Аналіз відомих досліджень і публікацій. На рисунку1 приведена класифікація моделей надійності ПЗ. Найбільш відомих моделей надійності ПЗ сьогодні існує близько десятка, тому в цій роботі вони згруповані за ознаками [1]. В якості класифікаційних ознак вибрані наступні:

– тимчасова структура процесів прояви помилок в ПЗ (час появи помилки, кількість помилок за заданий інтервал часу);

– складність програми (міра складності ПЗ - довжина, кількість функцій або модулів, даних і т.д.);

– розмітка помилок (штучне внесення в ПЗ відомих помилок);

– структура простору вхідних даних;

– структура тексту програми (розподіл помилок за текстом програми).

На практиці прості, елементарні помилки програм і даних можуть призводити до катастрофічних наслідків при функціонуванні ПЗ. В той же час, великі системні дефекти можуть тільки дещо погіршувати експлуатаційні характеристики ПЗ. Тому неможливо ранжирувати типи первинних помилок по мірі впливу на надійність та слід однаково ретельно відноситися до їх виявлення і усунення.

Статистика помилок в комплексних програмах і їх характеристики можуть служити орієнтиром для розробників при розподілі зусиль на відлагодження і оберігати їх від зайвого оптимізму при оцінці досягнутої якості і надійності ПЗ. Вони допомагають:

1. оцінювати реальний стан проекту і планувати необхідну трудомісткість і тривалість до його завершення;

2. вибирати методи і засоби проектування, програмування і тестування.

Реєстрація, збір і аналіз характеристик помилок в програмах є складним і трудомістким процесом. Тому є відносно невелике число робіт, в яких опубліковані реальні характеристики помилок.

При автономній роботі, і на початку комплексного відлагодження доля системних помилок невелика (~ 10%%), але вона істотно зростає (до 35-40%%) на завершальних етапах комплексного відлагодження. В процесі супроводу системні помилки є переважаючими (~ 80%% від усіх помилок).

Різними авторами були зроблені ряд уточнень вищевикладених моделей (до теперішнього часу запропоновані близько 15 математичних моделей для опису кількості помилок в ПЗ різної міри складності). Проте жодна з цих моделей не має явних переваг по точності апроксимації розподілів і прогнозування числа помилок в програмах в порівнянні з простою експоненціальною моделлю.

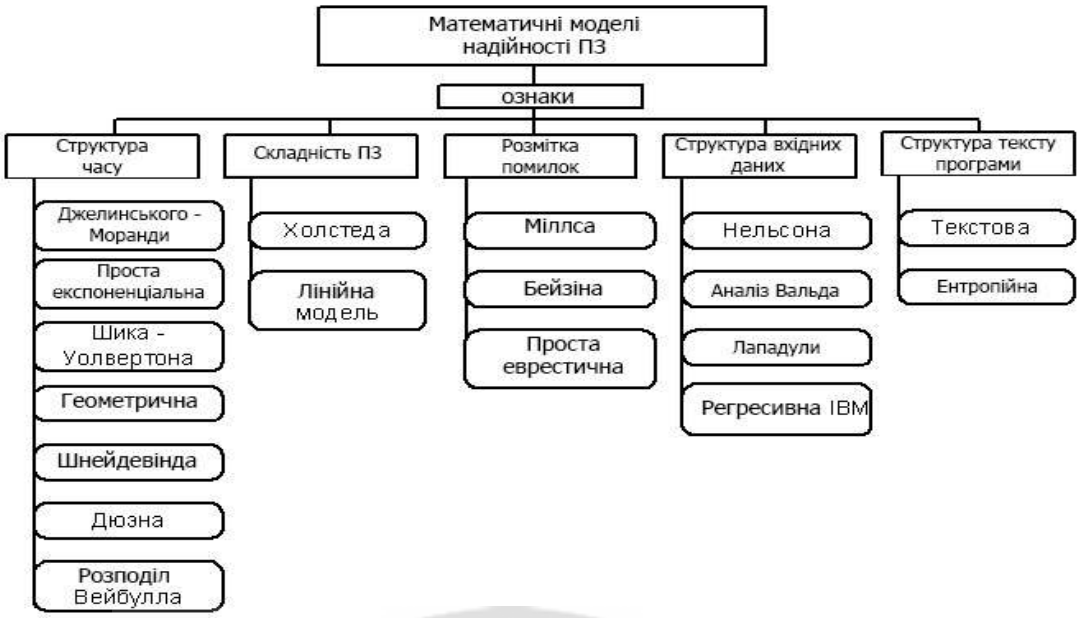

Рис. 1. Класифікація моделей надійності ПЗ

У роботі [2] приводиться порівняння моделей. Моделі Джелинського-Моранди і Шика-Уолвертона доцільні при моделюванні надійності ПЗ невеликого об'єму, а модифікована модель Шика-Уолвертона - для ПЗ великих проектів. Якщо при моделюванні необхідно отримати значення надійності (наприклад, середнє напрацювання повністю), то краще використати геометричні моделі. Деякі моделі не мають рішень (тобто розходяться за певних вхідних умов). Якщо є дані про інтервали часу між помилками, то краще скористатися геометричною моделлю, а якщо є дані про число помилок, що поступають одиницю часу, то краще застосовувати модель Шнейдевінда. Експоненціальна і дискретна моделі були перевірені при тестуванні реальних програм і добре відповідають дійсності [3].

В [2] робиться висновок, що на сьогодні неможливо вибрати найкращу модель серед десятка запропонованих.

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

Ці моделі у ряді випадків дозволяють не лише оцінювати показники надійності і готовності технічних систем і їх компонентів, але і дають можливість передбачати значення цих показників на основі накопиченого досвіду. Крім того, ряд моделей дозволяє на основі накопичених даних висловлювати припущення відносно режимів роботи, при яких найчастіше проявляються відхилення від нормального функціонування, а також про підхід до відновлення (ремонту) 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 заданого завдання в процесі застосування за призначенням. Під це визначення системи повністю підходить програмне забезпечення.

Із-за значних невизначеностей в усіх вищеописаних моделях в [1] рекомендується використати декілька моделей одночасно і об'єднувати їх результати.

Таким чином, запропоновані моделі дають задовільний результат при відносно високих рівнях інтенсивності прояву помилок, тобто при невисокій надійності ПЗ. У цих умовах математичні моделі призначені для наближеної оцінки:

1) потенційно можливій надійності функціонування програм в процесі випробувань і експлуатації;

2) числа пропущених помилок;

3) часу тестування, потрібного для виявлення наступної помилки;

4) часу, необхідного для виявлення із заданою вірогідністю більшості наявних помилок.

Можна припустити, що оцінки, приведені для декількох конкретних систем, дозволять прогнозувати ці характеристики для інших проектів. Імовірнісний підхід до надійності ПЗ повинен дати відповідь на одне з найскладніших питань при тестуванні ПЗ - коли треба зупинитися і завершити тестування, тобто, на протязі якого часу треба тестувати програму, щоб вона задовольняла вимогам по надійності? За допомогою запропонованої в цій роботі моделі надійності можна отримати відповідь на це питання.

Виклад основного матеріалу. У міру збільшення числа ПЕОМ та підвищення кваліфікації користувачів, на підприємстві все сильніше усвідомлювалася потреба в швидкому, надійному і зручному способі оперативного обміну великими обсягами даних між комп'ютерами, а головне, між користувачами. Це дозволило б, як мінімум, передавати інформацію між окремими робочими місцями, а як максимум, вирішувати завдання більш високого рівня, об'єднувати окремі автоматизовані місця в інтегровані комплекси. І, природно, попит породив пропозицію - були створені локальні обчислювальні мережі та архітектури файл-сервер та клієнт-сервер.

Як і для файл-серверної архітектури, складовими компонентами клієнт-серверної архітектури є сервер, клієнтські місця і мережева інфраструктура. Однак, на відміну від попереднього випадку, сервер тут є вже не сервером файлів, а сервером баз даних або навіть сервером додатків. Таким чином, на сервер лягає не просто зберігання файлів, а підтримання бази даних в цілісному стані або, в разі сервера додатків, навіть виконання тієї чи іншої частини прикладної задачі. Природно, що вимоги до сервера при цьому можуть зростати в рази [2].

У своєму розвитку технології «клієнт-сервер» пройшли кілька етапів, тому є різні моделі технології. Їх реалізація заснована на поділі структури СКБД на три компоненти:

– введення і відображення даних (інтерфейс з користувачем);

– прикладний компонент (запити, події, правила, процедури та функції, які характерні для даної предметної області);

– функції керування ресурсами (файловою системою, базою даних тощо).

Існують різні класифікації, але однією з найпоширеніших є використання чотирьох моделей технології «Клієнт-сервер»:

1. Модель файлового серверу (File Server - FS) .
2. Модель віддаленого доступу до даних (Remote Data Access - RDA) .
3. Модель сервера БД (Data Base Server - DBS) .
4. Модель сервера додатків (Application Server - AS) .

Для забезпечення надійності програмного забезпечення усі ці моделі мають свої переваги та недоліки.

З метою оцінки надійності системи на етапі тестування необхідно визначити такі важливі характеристики функціонування програмного комплексу, як:

1) розрахунок поточного часу напрацювання повністю;

2) розрахунок середнього часу напрацювання повністю за весь час моделювання роботи системи;

3) розрахунок імовірності відмови ПЗ за одиницю часу;

4) розрахунок коефіцієнта готовності.

Це завдання має сенс і для систем типу "клієнт-сервер", рисунок 2.

Є ПЗ типу "клієнт-сервер". Сервер обслуговує запити від N програм-клієнтів (далі просто клієнти). У ПЗ рівномірно по області визначення вхідних даних (ОВД) (A, B) розташовані Er помилок.

Помилками (відмовами) ПЗ є:

Відмови в програмі. Якщо ПЗ не модифікується, то інтенсивність його відмов залишається постійною.

Внутрішні відмови в програмі. Такі відмови обумовлені фундаментальними обмеженнями алгоритму, що використовується в ПЗ (наприклад, використання евристичних алгоритмів може привести до випадкових відмов).

Відмови, обумовлені обмеженнями при функціонуванні в реальному часі. У даних системах середовище може змінюватися динамічно. Тому якщо планування або розрахунок відгуку занадто великий, то до моменту надходження відгуку середовище може бути вже змінене настільки, що розрахований або спланований відгук не матиме необхідного ефекту.

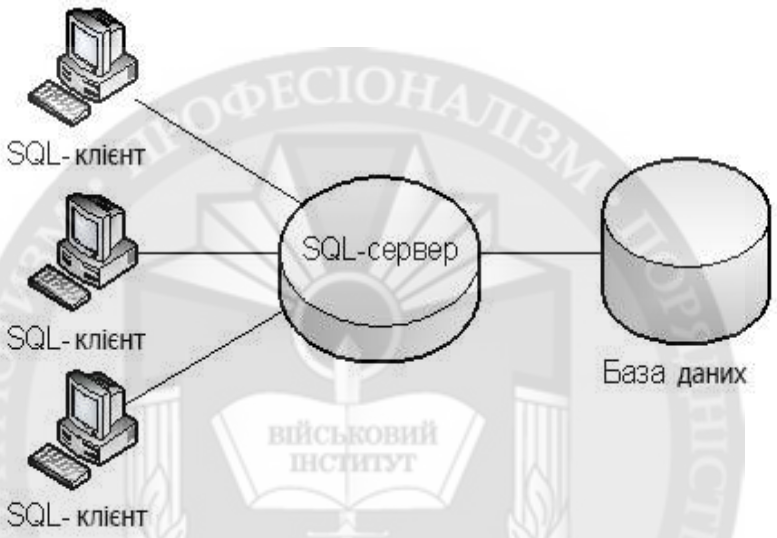

Рис. 2. Типова клієнт-серверна структура

Сервер складніший за програми-клієнти з точки зору розробки ПЗ в S разів. S - коефіцієнт складності сервера по відношенню до клієнтів. Кожен k -ий (k = 1, 2, ..., N) клієнт породжує пуасонівський потік даних до сервера інтенсивністю $\lambda зв$. Дані від клієнта розподілені по ОВД за нормальним законом з характеристиками m_k і σ_k, де m_k розподілене між клієнтами рівномірно по усій області вхідних даних, σ_k - розподілене рівномірно на меншій ділянці m_k, що відсікаються, на осі області даних. Це потрібно для імітації нерівномірності використання ОВД при малій кількості клієнтів [3].

На запит клієнта сервер відповідає даними, які розподілені рівномірно по усій області визначення даних (A, B).

На рис. 3 зображений розподіл запитів одного клієнта по області усіх можливих запитів до сервера, а також показаний рівномірний розподіл помилок по ОВД. При попаданні запиту клієнта або відповіді сервера в область ОВД, що містить помилку, вважається, що помилка виявлена і відповідний модуль виводиться з експлуатації для її виправлення:

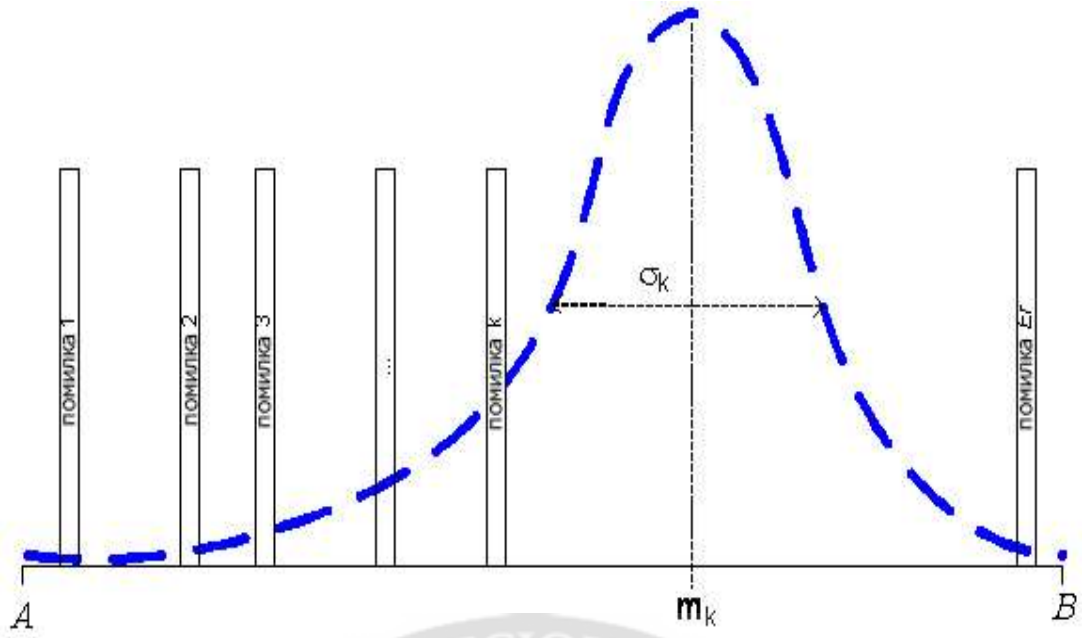

Рис. 3. Розподіл запитів k –го клієнта по області даних

Вхідними даними для моделі є:

P - кількість програмістів, обслуговуючих систему;

K - кількість програм-клієнтів;

α - ширина одного запиту клієнта як частина від ОВД (від 0 до 1, де 1 - це уся ОВД);

Δt - крок ітерації (доба);

s - коефіцієнт складності сервера в порівнянні з програмою-клієнтом;

$\lambda_{зв}$ - інтенсивність потоку звернень одного клієнта до сервера (1/доба);

$\lambda_{випр}$ - інтенсивність потоку виправлення помилки одним програмістом (1/доба);

$\lambda_{внес}$ - інтенсивність внесення помилки при виправленні одним програмістом (1/доба) або

$\rho_{внес}$ - імовірність внести помилку при виправленні одним програмістом;

M - кількість ітерацій (кількість спроб звернень програм-клієнтів до сервера в одному експерименті);

R - кількість експериментів для усереднення;

Er - початкова кількість помилок.

Для моделювання потоків запитів в ПЗ застосовується метод Монте-Карло.

Також є можливість оцінити первинну кількість помилок за наступним алгоритмом : Приймаємо ОВД за одиницю. Кожен клієнт в запиті генерує долю α; від ОВД. За час Δt клієнт звернеться до сервера ($\Delta t \cdot \lambda_{зв}$) разів. За час Δt усі клієнти звернуться до сервера ($\Delta t \cdot \lambda_{зв} \cdot K$) разів. І об'єм даних, який буде в ОВД при цьому рівний ($\Delta t \cdot \lambda_{зв} \cdot K \cdot \alpha$). Оскільки в нашій моделі помилки розподілені рівномірно по ОВД, то за час Δt буде виявлене ($\Delta t \cdot \lambda_{пом}$), де $\lambda_{пом}$ - первинна інтенсивність помилок в системі. Якби за час Δt клієнти звернулися в усю ОВД, то було б виявлено усі Er помилок. Тому, можна записати наступну пропорцію:

$$\frac{\Delta t \cdot \lambda_{зв} \cdot K \cdot \alpha}{\Delta t \cdot \lambda_{пом}} = \frac{1}{Er}.$$

Звідси знаходимо Er :

$$Er = \frac{\lambda_{пом}}{\lambda_{зв} \cdot K \cdot \alpha}.$$

При цьому вважається, що кожен з K клієнтів звернувся до сервера із запитом з даними, що не перетинаються в ОВД. Проте в реальності клієнти найчастіше звертаються до сервера з однотипними запитами, тому вважаємо $K = 1$ Тоді первинну кількість помилок можна оцінити як:

$$Er = \frac{\lambda_{ош}}{\lambda_{обр} \cdot \alpha}.$$

Висновки. Таким чином, вже на етапі тестування можливе передбачення характеристик ПЗ і оцінки часу, що необхідно витратити для досягнення заданого рівня надійності ПЗ при наявних ресурсах, та вироблення рекомендацій по підвищенню надійності ПЗ.

ЛІТЕРАТУРА :

1. Полонников Р.И. Методы оценки надежности программного обеспечения / Р.И. Полонников, А.В. Никандров. – СПб: Политехника, 1992. – 80 с.

2. Мясіщев О.А. Показники надійності захисту інформації у випадку системи із кінцевим числом діалогових каналів / А.В. Джулій, О.А. Мясіщев // Вимірювальна та обчислювальна техніка в технологічних процесах. – 2010. – № 2. – С. 118-124.

3. Лєнков С.В.Динамічні показники оцінки рівня функціональної безпеки інформаційної системи / С.В Лєнков, В.М Джулій, І.В Муляр // Сучасна спеціалізована техніка. Науково-практичний журнал. – 2016. – № 2(45). – С. 59-68.

REFERENCES:

1. Polonnykov R.Y. Metodш otsenky nadezhnosty prohrammnoho obespechenyya / R.Y. Polonnykov, A.V. Nykandrov // – SPb: Polytekhnyka, 1992. – 80 s.

2. Myasishchev O.A. Pokaznyky nadiynosti zakhystu informatsiyi u vypadku systemy iz kintsevym chyslom dialohovykh kanaliv / A.V. Dzhuliy, O.A. Myasishchev // Vymiryuval'na ta obchyslyuval'na tekhnika v tekhnolohichnykh protsesakh. – 2010. – # 2. – C. 118-124.

3. Lyenkov S.V. Dynamichni pokaznyky otsinky rivnya funktsional'noyi bezpeky informatsiynoyi systemy / S.V. Lyenkov, V.M. Dzhuliy, I.V. Mulyar // Suchasna spetsializovana tekhnika. Nakovo-praktychnyy zhurnal. – 2016. – # 2(45). – C. 59-68.

Рецензент: д.т.н., проф. Лєнков С.В., начальник науково-дослідного центру Військового інституту Київського національного університету імені Тараса Шевченка

д.т.н. с.н.с. Селюков А.В., к.т.н. Огневой А.В., к.військ., доц. Осыпа В.А., Бабий И.М.

РАЗРАБОТКА МОДЕЛЕЙ НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ТЕХНОЛОГИИ ТИПА «КЛИЄНТ-СЕРВЕР»

В статье проведен анализ моделей надежности программного обеспечения (ПО). Наиболее известных моделей надежности ПО сегодня существует около десятка, поэтому в этой работе они сгруппированы по признакам.

Рассмотрены компьютерные системы с централизованной архитектурой приложений файл-сервер и клиент-сервер.

Особое внимание уделено клиент-серверной архитектуре. Существуют различные классификации, но одной из самых распространенных является использование четырех моделей технологии «клиент-сервер»: модель файлового сервера (File Server - FS), модель удаленного доступа к данным (Remote Data Access - RDA), модель сервера БД (Data Base Server - DBS), модель сервера приложений (Application Server - AS).

С целью оценки надежности для систем типа "клиент-сервер" на этапе тестирования определены такие важные характеристики функционирования программного комплекса, как:

расчет текущего времени наработки до отказа, расчет среднего времени наработки до отказа за все время моделирования работы системы, расчет вероятности отказа ПО за единицу времени расчет коэффициента готовности.

Ключевые слова: надежность программного обеспечения, модели надежности, клиент-серверная архитектура, программа-клиент, область определения входных данных.

prof. Selyukov A.V., Ph.D. Ognievyi O.V., Ph.D. Osupa V.A., Babiy I.M.

DEVELOPMENT OF RELIABILITY MODELS FOR SOFTWARE TECHNOLOGY TYPE "CLIENT-SERVER"

In the article the analysis of models of reliability of software (SOFTWARE). The most famous reliability models FOR today there are about a dozen, so in this work they are grouped by characteristics.

Considered a computer system with a centralized application architecture file-server and client-server.

Special attention is paid to the client-server architecture. There are various classifications, but one of the most common is the use of four models of the technology "Client-server": a model of file server (File Server FS), model RDA (Remote Data Access - RDA) model of the DB server (Data Base Server DBS), a model of the application server (Application Server - AS) .

To assess reliability for systems of type "client-server" at the stage of testing identified the following important features of the operation of software, such as: calculation of the current operating time fully, the calculation of the average operating time fully during the simulation of the system, the calculation of the probability of failure per unit time, the calculation of the coefficient of readiness.

Key words: software reliability, reliability models, client-server architecture, the client program, the scope of the input data.