

## ПОРІВНЯЛЬНИЙ АНАЛІЗ МОДЕЛЕЙ НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАСОБІВ СПЕЦІАЛЬНОГО ЗВ'ЯЗКУ

### **Анотація:**

*Розглянуто питання моделювання надійності програмно-керованих радіозасобів, а також їх програмного забезпечення. Наведено порівняльну характеристику термінів і окремих показників надійності складових частин комп'ютерних систем. Запропоновано структуру моделі надійності програмного забезпечення. Узагальнено класифікацію відомих моделей надійності програмного забезпечення.*

### **Аннотация:**

*Рассмотрены вопросы моделирования надежности программно-управляемых радиосредств, а также их программного обеспечения. Приведена сравнительная характеристика сроков и отдельных показателей надежности составных частей компьютерных систем. Предложена структура модели надежности программного обеспечения. Обобщенно классификацию известных моделей надежности программного обеспечения.*

### **Abstract:**

*The question of reliability modeling and software-controlled radio equipment and their software. The comparative characteristic of the timing and the reliability of individual components of the computer system. The structure model of software reliability. Generalized classification of known reliability models software.*

Надійність сучасних програмно-керованих радіозасобів (ПРЗ) спеціального зв'язку залежить як від апаратних засобів (АЗ), так і від програмного забезпечення (ПЗ), оскільки відмова будь-якої складової веде до відмови систем в цілому. Тому для оцінки надійності системи в цілому необхідно враховувати вплив на її показники обох складових. Мета статті – обґрунтування вибору моделі надійності ПЗ, що дозволяє в умовах реальної експлуатації ПРЗ кількісно оцінити значення їх показників надійності.

Вирішення поставленого завдання ускладнюється тим, що період експлуатації складних технічних об'єктів і систем залежить від технічних, економічних і моральних чинників, причому останній для комп'ютерних систем являється визначальним. Внаслідок скорочення часу їх використання за призначенням, до декількох років, ускладнює отриманням об'єктивних статистичних даних про надійність як АЗ, так і ПЗ.

Програмно-керована радіостанція *SDR (software defined radio)* використовує стандартні апаратні засоби для виконання функцій під управлінням програмного забезпечення. Хоча програмувальні зв'язні радіостанції й розробляються з 70-х років, а деякі моделі навіть є на озброєнні, програмні й апаратні засоби усе ще тісно пов'язані один з одним. Тому великі надії покладають на відкриту архітектуру *SDR* радіостанцій наступного покоління, причому самі ПРЗ можуть випускати одні виробники, а функції й режими роботи буде визначати програмне забезпечення інших розробників [1]. Технологічні успіхи в області сумісних програмних засобів, систем цифрової обробки сигналів і мікропроцесорів роблять подібні проекти реальними.

Архітектуру радіостанцій розробляє консорціум компаній на чолі з *Raytheon Command, Control, Communications and Information Systems*, що включає також *ITT Industries, Rockwell Collins* й *Marconi CNI Division*. З грудня 2000 року й по 2006 рік ця група представляла практично той самий варіант архітектури зв'язних програмних засобів (АЗПЗ) *SCA (Software Communications Architecture)* [2-5]. Структура архітектури *SCA* показана на рис. 1 [5-8].

Архітектура системи була розроблена із застосуванням об'єктно-орієнтованого підходу (ООП).

Мова *UML (Unified Modeling Language)* – уніфікована мова моделювання) використовується для графічного подання інтерфейсів, у той час як самі інтерфейси визначаються мовою *IDL (Interface Definition Language)* – мова опису інтерфейсів). Обидва описи (*UML* й *IDL*) можуть бути реалізовані за допомогою стандартних засобів розробки, що дозволяє почати безпосередню розробку з визначення базових концепцій архітектури.

Як базовий стандарт покладено в основу внутрішнього функціонування *SDR*, обраний стандарт (технологія) *CORBA (Common Object Request Broker Architecture)* – загальна архітектура брокера об'єктних запитів – визначає загальні принципи побудови розподілених додатків). Він базується на об'єктній моделі рекомендації *X.733 ITU-T*, спеціалізованій мові *IDL* й архітектурі управління об'єктами *OMA (Object Management Architecture)*.

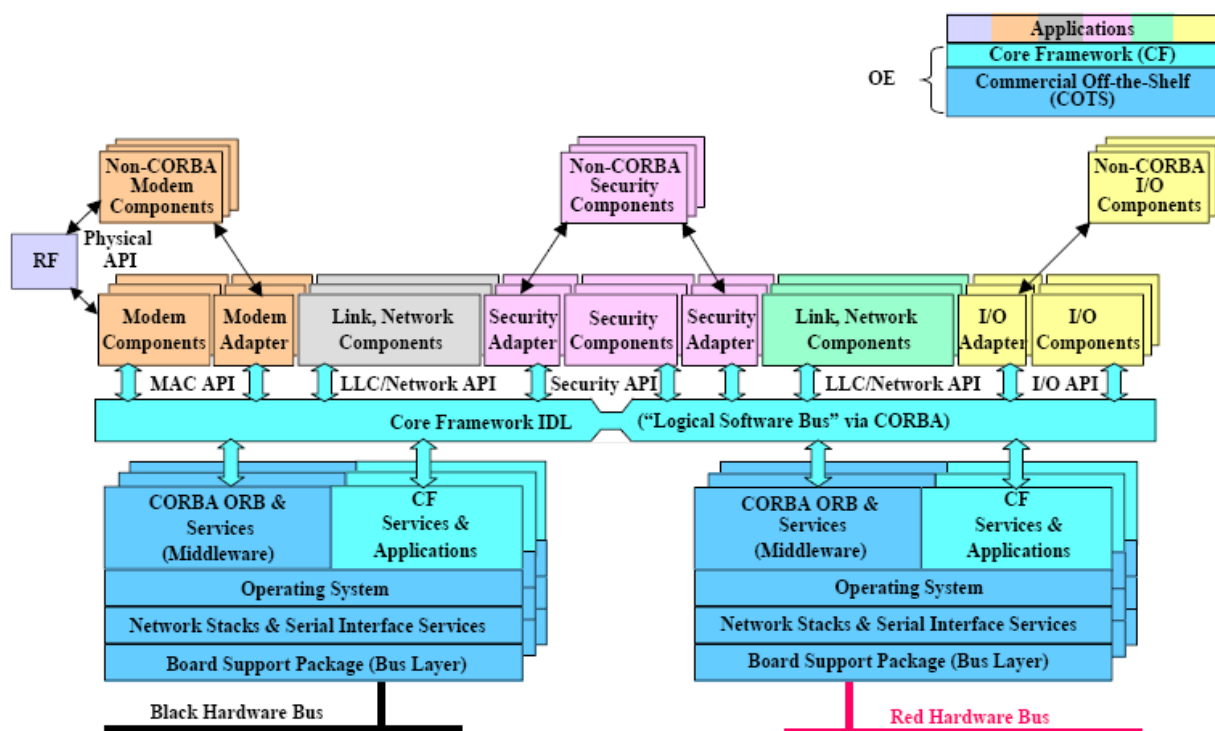


Рис. 1. Архітектура програмно-керованих засобів

Використання переваг *CORBA* дозволяє досягти максимальної універсальності модуля *SDR* за рахунок універсальності самого стандарту незалежно від протоколу передачі, структури даних й апаратної бази.

Відповідно до керівного принципу стандарту *TEMPEST NACSAM 5203* все інформаційне обладнання ділиться на «Red» (червоне) й «Black» (чорне). Червоне

обладнання – це таке обладнання, по ланцюгах якого розповсюджується секретна інформація у відкритому виді. Обладнання, що обробляє сигнали, які є невизначеними або через зміст тексту, або тому що відомості вже зашифровані, позначається чорним.

Ключовими перевагами подібної архітектури є [5-8]:

1. Максимальне використання комерційних протоколів і продуктів.
2. Ізолювання додатків рівня ядра й додатків нижнього рівня з використанням множинних рівнів відкритої комерційної програмної інфраструктури.
3. Використання середовища розподіленої обробки, що має в основі технологію *CORBA*, що забезпечує програмну переносимість, повторне використання коду й масштабованість.

Програмна архітектура визначає Операційне Середовище (*OE – Operating Environment*) як комбінацію набору сервісів ядра (*Core Framework Services*) і інфраструктурного програмного забезпечення (ПЗ), включаючи проміжне програмне забезпечення (ППЗ). Операційне середовище є інтегрованою в *SCA*.

На рис. 1 операційне середовище представлене у вигляді нижніх блоків, які складаються з [5-8]:

1. *CORBA Middleware* – ППЗ *CORBA*. Насамперед сам *ORB (Object Request Broker –* брокер об'єктних запитів – є ядром *CORBA* і будь-якого додатка написаного з використанням цієї технології, забезпечує пересилання запитів між *CORBA-об'єктами*), а також будь-які корисні сервіси *CORBA*, як сервіс іменування, сервіс подій і т.д. забезпечує роботу додатків, написаних за даною технологією. Всі інтерфейси взаємодії також визначаються за допомогою мови *IDL*.

2. *CF Services & Application* – сервіси й додатки, що відносяться до інфраструктури ядра. Забезпечують високорівневий підхід при програмуванні й проектуванні прикладних систем.

3. Обидва вищевказаних пункти функціонують під управлінням операційної системи (*Operation System*). Як зразок операційного середовища пропонують *POSIX*, як сумісний зі стандартом *CORBA OMG*. Конкретна версія *POSIX* – це *POSIX 1003.13*. Підвид *POSIX* спеціально розроблений для систем управління реального часу (*PSE52*). Надалі це називається *SCA POSIX Profile*. Всі інтерфейси взаємодії також визначаються за допомогою мови *IDL*.

4. *Network Stacks & Serial Interface Services* – стеки мережних протоколів і послідовні інтерфейси. Забезпечують взаємодію між різними засобами в мережі. Як підтримувані протоколи використовуються *PPP, SLIP, LAPx*, а також традиційний *IP*. Крім мережних протоколів цей блок відповідає за підтримку всіляких послідовних інтерфейсів передачі даних (*RS-232, RS-422, RS-423, RS-485, Ethernet* й *802.x*).

5. Передбачається, що *SCA* повинне функціонувати не тільки в розподіленому виді, але й на різних апаратних платформах, тобто повинен бути деякий засіб, за допомогою якого можна буде забезпечити кросплатформеність. Таким засобом є *Board Support Packages*, що забезпечує апаратний інтерфейс із конкретним середовищем передачі (*VME, PCI, CompactPCI, Firewire (IEEE-1394), Ethernet*).

Операційне середовище є базою, на основі якої функціонує корисний додаток. ОС, ППЗ, та *CF* забезпечують потрібну функціональність для корисної програми.

Рівень додатка (на рис. 1 зображений над шиною “*Logical Software Bus*”) складається з [5-8]:

1. *Modem-level digital signal processing* – рівня модему по обробці цифрового сигналу.
2. *Link-level protocol processing* – рівня обробки протоколу з'єднання.
3. *Network-level protocol processing* – рівня обробки протоколу мережі.
4. *Internetwork routing* – міжмережної маршрутизації (на схемі не зазначений).
5. *External input/output (I/O) access* – зовнішнього введення/виведення.
6. *Security* – безпеки.
7. *Embedded utilities* – впроваджених програм.

Всі рівні прикладної програми при роботі використовують сервісі *CF*. Взаємодія рівнів додатків безпосередньо з *OS* обмежено специфікацією *SCA POSIX Profile*. Взаємодія рівнів додатка з мережею не обмежується *OS*.

Додатки (на рис. 1 – *components* й *adapter*) складаються з одного або більше ресурсів [5-9]. Інтерфейс ресурсу надає загальний інтерфейс прикладного програмування – *API (Application Programming Interface* – набір функцій, необхідний для використання в прикладних програмах для управління й конфігурації програмного компонента. Як приклади ресурсів використовується: *LinkResource*, *NetworkResource* і *UtilityResource*.

Пристрої (*devices*) є різновидом ресурсу. Вони використовуються як представники реальних пристроїв. Прикладами пристроїв можуть бути *Modem Device*, *I/O Device* й *Security Device*. Розробка внутрішньої функціональності пристроїв не декларується *SCA*, залишаючи рішення за розроблювачем.

Адаптери є ресурсами або пристроями, призначеними для підтримки використання не *CORBA* елементів.

Апаратною реалізацією ППЗ є набори цифрових сигнальних процесорів (*digital signal processors – DSP*) і програмувальних логічних інтегральних схем (*field programmable gate arrays – FPGA*).

Ефективне функціонування ППЗ ґрунтується на надійності технічних засобів, а також надійності їх програмного забезпечення. Тому необхідно провести дослідження надійності функціонування ППЗ та їх компонентів.

Згідно [10] аналіз надійності, це систематизоване дослідження з метою визначення досягнутого рівня надійності при виконанні запланованих заходів щодо забезпечення і підвищення надійності та оцінки ефективності цих заходів.

В табл. 1 наведені визначення надійності та її властивостей як АЗ, так і ПЗ [10-12], а в табл. 2 – основні відмінності деяких показників їх надійності [11, 12].

Таблиця 1

Визначення надійності та її властивостей

Терміни	Апаратні засоби	Програмне забезпечення
Надійність	Властивість об'єкта зберігати у часі в установлених межах значення всіх параметрів, які характеризують здатність виконувати потрібні функції в заданих режимах та умовах застосування, технічного обслуговування, зберігання та транспортування	Властивість програми виконувати задані функції в заданих умовах роботи і на заданій ЕОМ

Безвідмовність	Властивість об'єкта виконувати потрібні функції в певних умовах протягом заданого інтервалу часу чи наробітку	Властивість програми виконувати задані функції без відмов протягом заданого часу в певних умовах зовнішнього середовища
Ремонто-придатність	Властивість об'єкта бути пристосованим до підтримання та відновлення стану, в якому він здатний виконувати потрібні функції за допомогою технічного обслуговування та ремонту	Пристосованість програми до усунення відмов через помилки програмування та їхні наслідки
Відмова	Подія, яка полягає у втраті об'єктом здатності виконувати потрібну функцію, тобто у порушенні працездатного стану об'єкта	Подія, яка полягає у втраті здатності програми виконувати потрібну функцію

Таблиця 2

## Порівняння окремих показників надійності складових комп'ютерних систем

Показник	Апаратні засоби	Програмне забезпечення
Безвідмовність	Визначають випадковими відмовами, що залежать від зміни параметрів на етапі експлуатації	Визначає правильність (коректність) ПЗ і цілком залежить від наявності помилок в ньому, допущених під час розробки
Середній наробіток на відмову	З часом скорочується, особливо на завершальному етапі експлуатації	З часом збільшується по мірі виявлення та усунення помилок
Відновлюваність	Відмова елемента в одному виробі не означає відмови в інших	Виявлення помилки в ПЗ одного вибору дозволяє виключити її в усіх інших

Розраховувати точні значення показників надійності складної системи, як і підтвердити їх випробуваннями, практично неможливо зважаючи на неадекватності математичних моделей фізичним, а так само складності розрахунків через велику розмірність рівнянь.

Відомо, що розрахунки показників надійності при експоненційному законі розподілу часу напрацювання на відмову дають занижені значення, але якщо вони задовільняють вимогам споживача, то можна твердити, що показники надійності системи в процесі експлуатації будуть не нижче розрахункових. Якщо неможливо встановити реальні закони розподілу відмов, то має сенс обчислити показники надійності системи оціночними методами в процесі моделювання [11].

Для прогнозування значень показників надійності складних технічних об'єктів і систем використовують математичні моделі. Структура математичної моделі надійності ПЗ показана на рис.2.



Рис. 2. Математична модель надійності програмного забезпечення

На сьогодні відомі моделі надійності ПЗ, перелік яких наведено в табл. 3, а огляд та аналіз в [11, 13].

Наявність безлічі різноманітних моделей, що описують один і той же процес свідчить, про актуальність задачі розрахунку надійності ПЗ і здавалося б, про всебічне теоретичне та практичне опрацювання цього питання. Проте це не так, оскільки кожна з моделей орієнтована на певний клас ПЗ або конкретні умови його тестування.

Слід відмітити, що на сьогодні відсутня єдина теорія, адекватно відображаючи надійність ПЗ, так як всі відомі моделі одностороннє розглядають процес появи програмних

помилки. Встановлено, що процес виявлення помилки в ПЗ практично є перехідним процесом в лінійній системі першого порядку, що відкриває нові можливості в дослідженні та уточненні моделей надійності ПЗ [11-18].

На рис. 3 виділено моделі, які кількісно оцінюють показники надійності на етапі експлуатації комп'ютерних систем.

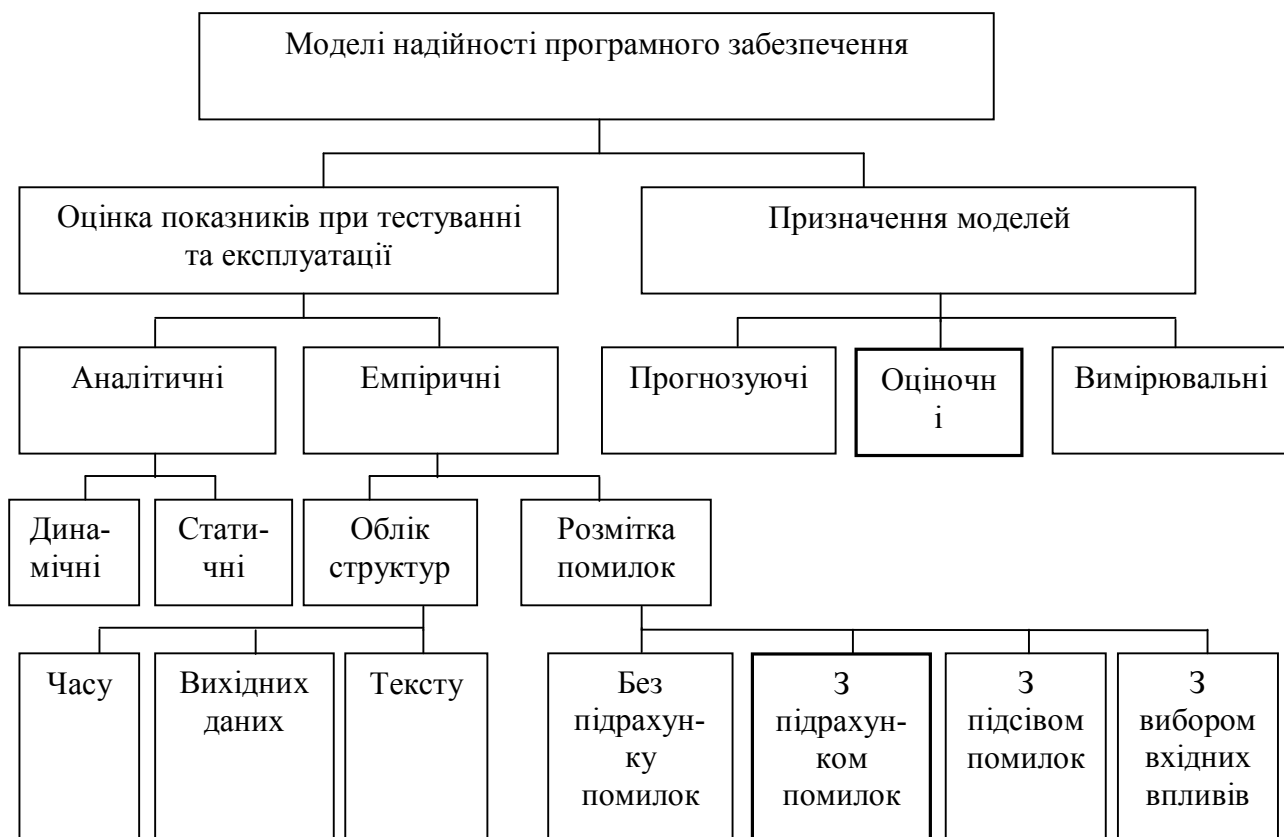


Рис. 3. Види моделей надійності програмного забезпечення

На основі дослідження реальних статистичних даних по надійності ПЗ [13] в табл. 3 показані результати розрахунку значень дисперсії ( $D$ ) і середнього квадратичного відхилення ( $\sigma$ ) прогнозованого ( $m_i$ ) від експериментального ( $m_i^*$ ) числа відмов ПЗ за  $i$ -місяців експлуатації комп'ютерної системи ( $i = \overline{1,12}$ )

$$D = \frac{1}{T} \sum_{i=1}^T (m_i^* - m_i)^2; \quad \sigma = \sqrt{D}; \quad T = 12.$$

Аналіз отриманих результатів показує, що з урахуванням доступності вхідних даних, обмежень і допущень, що відповідають реальним умовам експлуатації комп'ютерних систем, достатню для практики точність забезпечує модель Джелінські-Моранді ( $m_i'$  на рис. 4), при цьому в кращу сторону виділяється модель перехідних процесів ( $m_i''$  на рис. 4), а в гіршу – геометрична Моранді ( $m_i'''$  на рис. 4).

Таблиця 3

**Значення імовірнісних характеристик моделей надійності ПЗ**

Ранг	Моделі надійності програмного забезпечення	$\sum_i^T (m_i^* - m_i)^2$	$D$	$\sigma$
1	Перехідного процесу	13.37	1.114	1.056
2	Експоненціальна модель Мусси	13.91	1.159	1.077
3	Проста експоненціальна модель	13.93	1.161	1.077
4	Модель Джелінські-Моранді	17.15	1.429	1.195

Ранг	Моделі надійності програмного забезпечення	$\sum_i^T (m_i^* - m_i)^2$	$D$	$\sigma$
5	Модель Дюєна	22.40	1.867	1.366
6	Логарифмічна модель Мусси-Окумото	27.80	2.317	1.522
7	Модель S-образного зростання надійності з перегинами	43.97	3.664	1.914
8	Модель S-образного зростання надійності	57.02	4.752	3.180
9	Модель Вейбулла	64.86	5.405	2.325
10	Модель Шика-Вольвертона	69.00	5.750	2.398
11	Геометрична модель Моранді	71.20	5.933	2.436

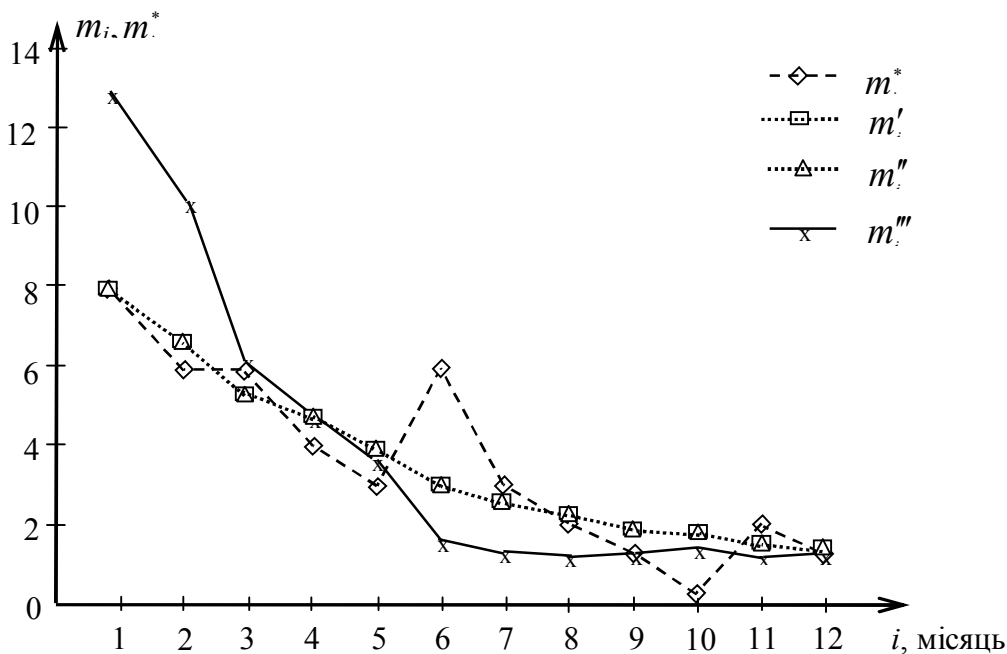


Рис. 4. Залежності істинного і прогнозованого числа відмов програмного забезпечення за рік експлуатації комп'ютерної системи

Залежності експериментальних даних ( $m_i^*$ ) і результатів розрахунків з використанням моделі Джелінскі-Моранді ( $m_i$ ) за рік експлуатації комп'ютерної системи показані на рис. 4.

Прояви помилок ПЗ в моделі Джелінскі-Моранді розглядаються як марковський процес, при цьому розподіл інтенсивності відмов експоненціальний, що добре погодиться з експериментальними даними (рис. 4).

Допущення при використанні цієї моделі [12-15]:

- інтенсивність виявлення помилок пропорційна поточному числу помилок в ПЗ;
- усі помилки однаково ймовірні й їх поява незалежна одна від іншої;



поява кожної помилки приводить к порушенню правильності функціонування ПЗ;  
час до слідуючого відказу ПЗ розподілено експоненціальне;  
ПЗ функціонує у середовищі близькому до реальних умов експлуатації комп'ютерних систем;  
помилки ПЗ після виявлення усуваються без внесення нових;  
інтенсивність виявлення помилок постійна в інтервалі між двома суміжними моментами появи помилок;  
після виявлення і усунення усіх помилок ПЗ надійність комп'ютерної системи визначається показниками АЗ.

На основі використання моделі надійності ПЗ Джелінські-Моранді в [19] запропоновано методу, що дозволяє в реальних умовах експлуатації програмно-керованих засобів зв'язку оцінити показники надійності окремих складових (АЗ та ПЗ) і систем в цілому. По мірі накопичення статистичних даних про відмови ПЗ точність прогнозування результатів підвищується.

### Висновки

1. Проведений аналіз архітектури програмно-керованих радіозасобів показав необхідність розробки методичного апарату аналізу надійності пристроїв, що ґрунтуються на базі цифрових процесорів та програмних засобів.
2. Наведено порівняльну характеристику термінів і окремих показників надійності складових частин комп'ютерних систем. (табл. 1, 2).
3. Запропоновано структуру моделі надійності програмного забезпечення (рис. 2).
4. Узагальнено класифікацію відомих моделей надійності програмного забезпечення (рис. 3).
5. Вперше проведено розрахунок дисперсії і середнього квадратичного відхилення значення числа відмов програмного забезпечення для відомих моделей (табл. 3) і показано доцільність використання моделі Джелінські-Моранді для кількісної оцінки показників надійності програмного забезпечення в методиці [10].

### Література:

1. Щербак Н. Программируемые радиостанции – будущее тактической связи // ЭЛЕКТРОНИКА: Наука, Технология, Бизнес. – 2001. С. 16-19.
2. Maier K. Mapping waveforms to systems: What would a wideband networking waveform system require? // Military EMBEDDED SYSTEMS. – October, 2005. P. 38-41.
3. Uhm M. Adaptivity in Action for SDR and Cognitive Radio // COTS Journal. – February, 2006. (<http://www.cotsjournalonline.com>).
4. Joint Tactical Radio System (Encyclopedia) // (<http://www.absoluteastronomy.com/>)
5. Software Communications Architecture Specification // MSRC-5000SCA, V2.2. Contract No. DAAB15-00-3-0001. – November 17, 2001. (<http://jtrs.spawar.navy.mil/>).
6. Gonzalez A., Hess R. Joint Tactical Radio System. SCA Developer's Guide // Raytheon Company. Contract No. DAAB15-00-3-0001, Document Number: Rev 1.1. – June 18, 2002. (<http://jtrs.spawar.navy.mil/>).
7. Software Communications Architecture Specification // JPEO JTRS-5000, SCA V3.0. – August 27, 2004. (<http://jtrs.spawar.navy.mil/>).
8. Software Communications Architecture Specification // JPEO JTRS, Version 2.2.2. – April 21, 2006. (<http://jtrs.spawar.navy.mil/>).

9. Specialized Hardware Supplement to the Software Communication Architecture Specification // JTRS-5000 SP, V3.0. – August 27, 2004. P.4-1 – 4-7. (<http://jtrs.spawar.navy.mil/>).
10. ДСТУ 2860-94 Надійність техніки. Терміни та визначення. –К.:Держстандарт України, 1995. - 92 с.
11. Половко А.М. , Гуров С.В. Основы теории надежности. –СПб.: БХВ – Петербург , 2006. - 704 с.
12. Локазюк В.М., Савченко Ю.Г. Надійність, контроль, діагностика і модернізація ПК. –К.: Академія, 2004. - 376 с.
13. Маевский Д.А., Яремчук С.А. Анализ моделей надежности программного обеспечения гарантоспособных информационных систем. - 2010, С. 68-79. (<http://www.nbuiv.gov.ua/portal/>)
14. Антощук С.Г. Прогнозирование количества ошибок на этапе эксплуатации адаптируемых учетных информационных систем / С.Г. Антощук, Д.А. Маевский, С.А. Яремчук // Радиоэлектронные и компьютерные системы, НТЖ, Харьков: ХАИ. -2010. -№6 (47) – С. 204-209.
15. Моделирование гарантоспособных систем и сетей / Харченко В.С., Боярчук А.В., Куланов В.Н., Локазюк В.Н., Одарущенко О.Н., Поморова О.В., Фурманов А.А.// Харьков, ХАИ : -2008. – С. 5-6.
16. Полонников Р.И. Методы оценки показателей надежности программного обеспечения / Полонников Р.И., Никандров А.В.// СПб.: Политехника. - 1992. - 78 с.
17. Оценка качества и экспертиза программного обеспечения / Харченко В.С., Андрашов А.А., Гордеев А.А., Лобачева Е.И. // Харьков: ХАИ, 2008. - 99 с.
18. Маєвський Д.А. Електричне моделювання процесу виявлення помилок в програмному забезпеченні інформаційних систем // Сб. Електромашиностроение и электрооборудование. 2010, - Вып.75, – С. 113-115.
19. Сакович Л.Н. Подходы к оценке надежности программно-управляемых средств связи // Зв'язок. – 2010. - №4, – С. 36-39.