

Міністерство освіти і науки України  
Чернівецький національний університет  
імені Юрія Федьковича

Добровольський Ю.Г.

**ІНЖЕНЕРІЯ НАДІЙНОСТІ  
ПРОГРАМНОГО  
ЗАБЕЗПЕЧЕННЯ**

*Матеріали лекцій*

Навчальний посібник

Чернівці  
КПЗКС ІФТКН  
2022

ББК 28.9  
Е62  
УДК 004.312.26:001.891

*Рекомендовано Вченою радою*

*Навчально-наукового інституту фізико-технічних та комп'ютерних наук Чернівецького національного університету імені Юрія Федьковича  
протокол No 8 від 22.09.2022 р.*

*Інженерія надійності програмного забезпечення. Матеріали лекцій: навч. посіб. / Укл.: Добровольський Ю.Г. Чернівці: ЧНУ ім. Ю.Федьковича, 2022. 126с.*

Навчальний посібник з дисципліни “Інженерія надійності програмного забезпечення” призначений для формування у студентів знань, вмінь та навичок з сучасних методів та засобів дослідження програмного забезпечення на надійність, які будуть корисними при проведенні власних розробок при підготовці магістерської роботи і подальших фахових практичних та наукових роботах для створення сучасних інформаційних систем. А також формування навиків аналізу та вирішення задач з надійності інформаційних систем, які доцільно вирішувати засобами інтелектуальних систем, вміти використовувати сучасні інформаційні технології для створення та дослідження надійних інтелектуальних систем для широкого застосування.

Навчальний посібник призначений для підготовки фахівців другого (магістерського) рівня за спеціальностями 121 - Програмна інженерія, 122 - Комп'ютерні науки, 123 - Комп'ютерна інженерія та інших напрямків інформаційних технологій.

© Добровольський Ю.Г. 2022

## ВСТУП

В останні роки вартість розроблення програмного забезпечення (ПЗ) та витрати на відмови програмних систем стали одними із основних затрат на розроблення складних технічних систем.

Відмова ПЗ може призвести до неочікуваного стану системи або потоку непередбачених дій. Внаслідок відмови ПЗ можливі пошкодження чи руйнування майна, поранення або загибель людей, втрата грошових коштів. Загальновизнаним в індустрії програмного забезпечення є той факт, що висококваліфіковані програмісти допускають в середньому 6 помилок в програмному забезпеченні при написанні 1000 рядків коду. При такому показнику звичайне комерційне ПЗ, яке складається з 350000 рядків коду, містить близько 2000 помилок, зокрема помилки, пов'язані з витоком пам'яті, пов'язані з текстом програми, помилки використання сторонніх бібліотек, стандартні бібліотечні помилки та інші. Тому питання оцінювання та забезпечення надійності ПЗ відіграють все більшу роль у розробленні складних технічних систем, а побудова все адекватніших моделей надійності ПЗ є важливою науково-технічною проблемою.

## ЗМІСТ

Вступ	3
Зміст	4
1. Введення у інженерію надійності програмного забезпечення	7
1.1. Мета навчальної дисципліни	7
1.2. Завдання вивчення дисципліни	8
1.3. Організаційні засади курсу «Інженерія надійності програмного забезпечення»	9
1.4. Актуальність питання надійності ПЗ	9
1.5. Сучасні уявлення про надійність ПЗ	10
2. Фундаментальні поняття і визначення теорії надійності	14
2.1. Визначення поняття "надійність"	14
2.2. Класифікація і характеристики відмов	19
2.3. Термінологія теорії надійності	21
3. Класифікація технічних систем	24
3.1 Проблема побудови природної систематики техніки	24
3.2. Класифікація технічних систем за функцією та принципом дії	29
3.3. Класифікація технічних систем за рівнем складності	30
3.4. Класифікація технічних систем за способом виготовлення	32
3.5. Класифікація технічних систем за ступенем оригінальності	33
3.6. Класифікація технічних систем за типом виробництва	34
3.7. Класифікація ПЗ за його місцем у технічному процесі	35

4. Програмне забезпечення та його класифікація	36
4.1. Класифікація програмного забезпечення	36
4.2. Системне програмне забезпечення	36
4.3. Прикладне програмне забезпечення	39
4.4. Системи програмування	41
5. Критерії надійності	42
5.1. Критерії надійності невідновних систем	42
5.2. Критерії надійності відновлюваних об'єктів	46
6. Закони розподілу часу до відмови	49
6.1. Життєвий цикл технічного засобу	49
6.2 Розподіл Вейбулла	50
6.3 Експоненційний розподіл	52
6.4 Розподіл Релея	54
6.5 Гамма-розподіл	55
6.6 Нормальний розподіл	57
6.7 Трикутний розподіл	59
6.8 Закони розподілу дискретних випадкових величин	61
6.8.1 Біноміальний розподіл	61
6.8.2 Розподіл Пуассона	62
6.8.3 Геометричний розподіл	62
7. Надійність програмного забезпечення. Визначальні фактори	63
7.1. Основні поняття	63
7.2. Модель аналізу надійності програмних засобів	63
7.3. Фактори, що впливають на надійність ПЗ	65
7.4. Помилки ПЗ	65
7.5. Засоби та способи підвищення надійності ПЗ	70
7.6. Проблеми дослідження надійності ПЗ	71
7.7. Тестування ПЗ	71
7.8. Класифікація показників якості ПЗ	73
7.9. Основні показники надійності ПЗ	74

7.10. Загальна характеристика та класифікація моделей надійності програмного забезпечення	78
7.11. Моделі надійності програмного забезпечення з урахуванням недосконалого відлагодження	86
7.12. Моделі надійності програмного забезпечення з урахуванням розподілу зусиль тестування	90
8. Моделі надійності ПЗ	97
8.1. Динамічні моделі надійності	98
8.2. Статичні моделі надійності	102
Висновки	107
Список літератури	109
Додаток 1. Контрольні запитання	113
Додаток 2. Приклади тестових завдань	116

# 1. Введення у інженерію надійності програмного забезпечення

## 1.1. Мета навчальної дисципліни

Формування теоретичних знань та практичних умінь виявляти, ставити і вирішувати проблеми в галузі інформаційних технологій, аналізувати предметну область, ідентифікувати, класифікувати та описувати проблеми, знаходити методи й підходи до їх розв'язання, формулювати вимоги.

Під час вивчення дисципліни студенти познайомляться з підходами до:

- критичного аналізу, оцінки і синтезу нових та складних ідей в інженерії програмного забезпечення;

- розвитку й реалізації нових конкурентоздатних ідей в галузі інформаційних технологій;

- критичного переосмислення наявних інформаційних технологій та відстеження тенденцій їх розвитку;

- використання теоретичних та практичних принципів і інструментальних засобів в професійній галузі та підходами до їх застосування;

- пошуку, обробки та аналізу інформації з різних джерел;

- розробки наукових проектів та управління ними;

- використання новітніх інформаційних і комунікаційних технологій.

Студенти мають засвоїти поняття та визначення:

- теорії надійності, критеріїв надійності, законів розподілу часу до відмови;

- методів аналізу надійності технічних систем; основних математичних моделей надійності програмних систем; основних методів забезпечення і підвищення надійності техніки;

У студентів має сформуватися цілісна система теоретичних знань з курсу «Інженерія надійності програмного забезпечення».

## **1.2. Завдання вивчення дисципліни**

Основними завданнями вивчення дисципліни «Інженерія надійності програмного забезпечення» є вивчення теоретичних основ математичних методів теорії надійності, засвоєння студентами понять про методи моделювання, оцінки та оптимізації надійності технічних систем, оптимізація досвіду з аналізу показників надійності функціональних систем.

Мета проведення лекцій полягає у:

- засвоєнні студентами понять та визначень теорії надійності, критеріїв надійності, законів розподілу часу до відмови; методів аналізу надійності технічних систем; основних математичних моделей надійності програмних систем; основних методів забезпечення і підвищення надійності техніки;

- формуванні у студентів цілісної системи теоретичних знань з курсу «Інженерія надійності програмного забезпечення».

Мета проведення практичних занять полягає у тому, щоб навчити студентів:

- визначати показники надійності програмної системи за експериментальними даними;

- досліджувати надійність і ризик відновлюваних та не відновлюваних даними;

- досліджувати надійність і ризик відновлюваних та не відновлюваних систем;

- аналізувати надійність систем складної структури.



### **1.3. Організаційні засади курсу «Інженерія надійності програмного забезпечення»**

Дисципліна «Інженерія надійності програмного забезпечення» вивчається на першому році другого семестру підготовки магістрів зі спеціальності 121 «Програмна інженерія». Навчальний курс передбачає 30 лекційних годин (15 лекцій) та 15 годин лабораторних робіт (8 занять). На самостійну роботу студентам відводиться 105 годин. Усього 150 годин.

Курс поділений на два модулі – 8 лекцій та 4 лабораторні роботи у першому модулі та 7 лекцій і 8 лабораторні роботи у другому модулі.

Підсумковий контроль має відбуватись у вигляді іспиту. Розподіл балів між лабораторними роботами і іспитом має бути таким:

70 балів максимум за лабораторні роботи

30 балів максимум за іспит.

### **1.4. Актуальність питання надійності ПЗ**

В останні роки вартість розроблення програмного забезпечення (ПЗ) та витрати на відмови програмних систем стали одними із основних затрат на розроблення складних технічних систем.

Відмова ПЗ може призвести до неочікуваного стану системи або потоку непередбачених дій. Внаслідок відмови ПЗ можливі пошкодження чи руйнування майна, поранення або загибель людей, втрата грошових коштів. Загальновизнаним в індустрії програмного забезпечення є той факт, що висококваліфіковані . При такому показнику звичайне комерційне ПЗ, яке складається з 350000 рядків коду, містить близько 2000 помилок, зокрема помилки, пов'язані з витоком пам'яті, пов'язані з текстом

програми, помилки використання сторонніх бібліотек, стандартні бібліотечні помилки та інші. Тому питання оцінювання та забезпечення надійності ПЗ відіграють все більшу роль у розробленні складних технічних систем, а побудова все адекватніших моделей надійності ПЗ є важливою науково-технічною проблемою.

### **1.5. Сучасні уявлення про надійність ПЗ**

Важливою проблемою сучасності є питання забезпечення надійності технологій. Розвиток сучасних інформаційних технологій та використання програмно-апаратних комплексів і систем у всіх сферах людського життя призводить до значного збільшення ризику виникнення збоїв у роботі такого обладнання. У цьому випадку через специфіку характеру програмного забезпечення та особливості його поведінки з точки зору надійності, питання аналізу та забезпечення його надійності все ще залишається широким полем для досліджень. Разом із ускладненням технічного завдання пропорційно зростає складність ПЗ і, як наслідок – процес точного і правдивого оцінювання показників надійності пропорційно ускладнюється.

Для кількісного оцінювання та прогнозування надійності ПЗ використовуються відповідні моделі надійності, призначені для передбачення кількості програмних дефектів у ПЗ. Існує багато моделей надійності ПЗ, які переважно ділять на два класи: статичні та динамічні [1, 9]. Поширеніші з них динамічні моделі припускають, що в програмі кількість помилок є величиною дискретною, а враховуючи, що кумулятивне значення величини помилок – це ціле число, динамічні моделі розраховують величину початкових дефектів на етапі відлагодження ПЗ і кількість дефектів, які ми отримаємо на кінець етапу

відлагодження. Ці моделі використовують показник інтенсивності відмов як основну характеристику появи відмови.

Звичайно, що динамічні моделі припускають, що інтенсивність відмов є сталим числом або функцією часу відлагодження або ж випадковим числом із заданим законом розподілу [6].

Багато моделей припускають ідеальне відлагодження – імовірність усунення помилки становить 100 %, під час усунення помилки не вводиться нових, у такий спосіб кількість помилок, що залишилась в програмі, є незростаючою функцією тривалості відлагодження. У моделях, що припускають не ідеальне відлагодження (помилки не усуваються, вводяться, чи не змінюються під час кожного відлагодження), кількість помилок, що залишилися, може зростати чи зменшуватись [9]. Але на сьогодні не існує універсальної моделі надійності ПЗ, яка б повністю адекватно описувала процес прогнозування надійності та була б застосовною для всіх видів і типів ПЗ, а також враховувала всі метрики, характеристик тики та особливості програмних продуктів.

Вартість виправлення дефекту на етапі експлуатації ПЗ є в рази вищою, аніж вартість виправлення дефекту на етапі розроблення чи тестування. Звичайно, що такі етапи життєвого циклу програмного забезпечення, як тестування чи відлагодження, можуть зменшити кількість дефектів у програмі, але це водночас зумовлює збільшення вартості розроблення цього ПЗ [3, 4, 7, 8]. Отже, постає питання у правильному балансуванні витрат на тестування розроблення та проєктування програмного забезпечення. Під час стадії тестування показники надійності є важливими чинниками, які впливають на рішення чи програма буде випущена в експлуатацію. Тому виникає потреба у

використанні різних моделей надійності ПЗ, які давали б змогу розраховувати кількісні показники надійності ПЗ.

Моделі, які використовуються для визначення надійності ПЗ, поділяють на два класи: ймовірнісні та детерміністичні [3, 4, 7, 8]. Звичайно, що класифікація тих чи інших моделей може відрізнятися від праці до праці.

Можна виділити такі класи моделей: моделі, які засновані на висіванні помилок, апроксимації залежностей, інтенсивності відмов, на підставі неоднорідного пуассонового процесу, на підставі марковської структури та на підставі зростання надійності. Також варто зазначити, що у певних джерелах загальний розподіл моделей зберігається, натомість додаються моделі, які враховують архітектурний підхід, уніфіковані моделі, також моделі з урахуванням джерела походження вхідних даних і т.ін. [8]. До того ж моделі з урахуванням архітектури програмного забезпечення нерідко виділяють в окрему групу, які водночас поділяють на моделі, що засновані на використанні певних архітектурних підходів. Найчастіше згадувані моделі надійності ПЗ проілюстровано на рис. 1 [8].

Моделі на підставі неоднорідного пуассонового процесу розглядають стадію розроблення та від лагодження як процес підрахунку кількості знайдених дефектів, що визначається функцією середнього значення. У разі використання однієї з моделей, які засновані на неоднорідному пуассоновому процесі, визначення показників надійності ПЗ можливе після обчислення функції математичного сподівання відмов. Обчислення параметрів, які використовуються у вибраних моделях, зазвичай проводиться за використання методу Ньютонa, методу максимальної правдоподібності або регресійним методом [8].

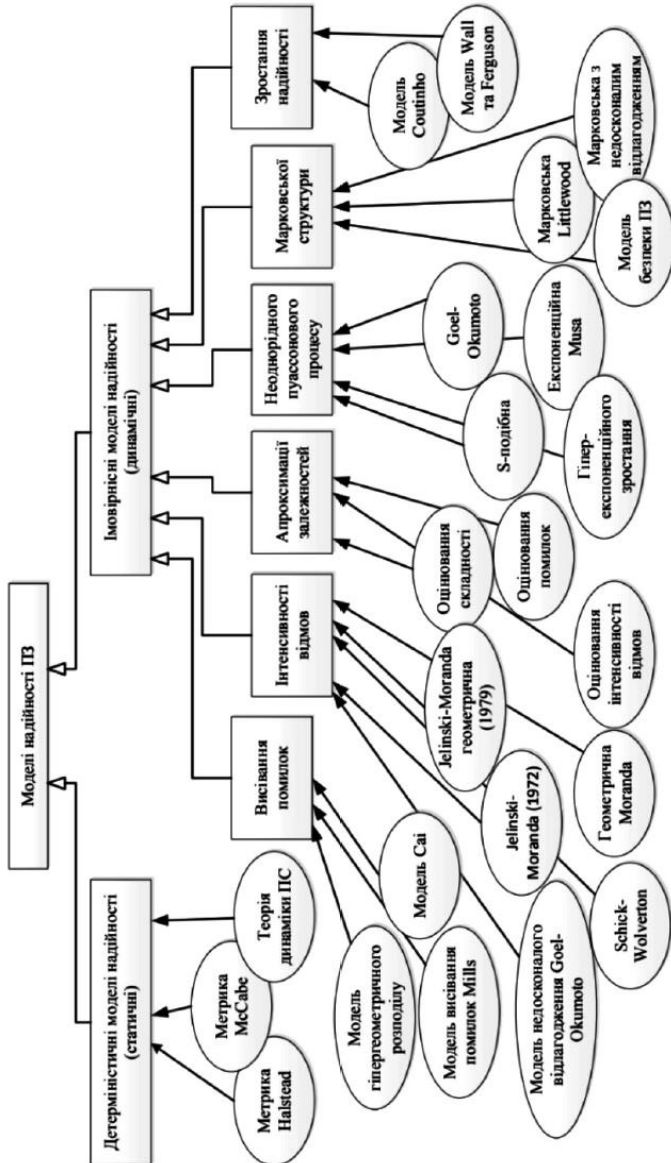


Рис. 1. Схема класифікації моделей надійності ПЗ / Scheme of classification of software reliability models [8]

## 2. Фундаментальні поняття і визначення теорії надійності.

### 2.1. Визначення поняття "надійність".

Обчислювальні пристрої, ПК і комп'ютерні системи є важливою складовою сучасного виробництва. Нині ця галузь розвивається особливо інтенсивно. Однак розвиток засобів комп'ютерної техніки неможливий без забезпечення їх надійності. Її, в свою чергу, характеризують багато показників і параметрів. Однією з особливостей комп'ютерних систем і мереж порівняно з іншими, наприклад, механічними, є те, що їх надійність визначає надійність елементів та надійність програмного забезпечення.

Згідно ДСТУ ISO 9000:2007 Системи управління якістю. Основні положення та словник термінів (ISO 9000:2005, IDT) у п. 3.5.3 надійність (dependability), визначається як «збірний термін, який вживають для описування характеристики готовності та чинників, що її зумовлюють: характеристик безвідмовності, ремонтпридатності та забезпеченості технічного обслуговування й ремонту». До речі, згідно цього ДСТУ (п. 3.7.1) інформація (information) – це «значущі дані», а документ (document) – це «інформація (3.7.1) та її носій» (п. 3.7.2).

Для нормативного забезпечення методів, заходів та засобів вимірювання, спрямованих на досягнення необхідного рівня надійності, використовується ДСТУ 2861-94. Надійність техніки. Аналіз надійності. Основні положення та ДСТУ 2860-94. Надійність техніки. Терміни та визначення. Ці стандарти забезпечують ефективність організаційно-технічних, конструкторсько-технологічних і експлуатаційних заходів, спрямованих на досягнення необхідного рівня надійності технічних засобів (ТЗ).

Аналізуванням і дослідженням цих питань займається наука, що називається теорією надійності (theory of reliability). Основною її задачею є вивчення закономірностей виникнення відмов ТЗ. Ця наука базується на теорії ймовірності і математичної статистики, тому всі розрахунки надійності носять ймовірнісний та статистичний характер.

При проектуванні технічних засіб (hardware) має відповідати всім технічним вимогам. Ці вимоги можна розділити на:

- головні, що забезпечують виконання заданих функцій;
- допоміжні, що пов'язані зі зручністю використання, загальним виглядом та ін.

З точки зору теорії надійності будь-який ТЗ можна охарактеризувати його властивостями, технічним станом та можливістю відновлення після втрати працездатності. При цьому найважливішою комплексною властивістю ТЗ є його надійність (reliability).

Обчислювальні пристрої, ПК і комп'ютерні системи є важливою складовою сучасного виробництва. Нині ця галузь розвивається особливо інтенсивно. Однак розвиток засобів комп'ютерної техніки неможливий без забезпечення їх надійності. Її, в свою чергу, характеризують багато показників і параметрів. Однією з особливостей комп'ютерних систем і мереж порівняно з іншими, наприклад, механічними, є те, що їх надійність визначає надійність елементів та надійність програмного забезпечення.

**Надійність** – властивість об'єкта зберігати в часі у встановлених межах значення всіх параметрів, які характеризують здатність виконувати потрібні функції в заданих режимах та умовах застосування, технічного обслуговування, зберігання і транспортування.

Надійність включає в себе такі властивості: безвідмовність (faultlessness), довговічність (longevity), збережність (safety) та ремонтпридатність.

**Безвідмовність (faultlessness)** – властивість об'єкта виконувати потрібні функції в певних умовах протягом заданого інтервалу часу чи наробітку.

**Напрацювання (operating time) – наробіток; напрацювання.** Тривалість чи обсяг роботи об'єкта.

Примітка. Нарробіток може бути як неперервною величиною (тривалість роботи в годинах, кілометрах пробігу тощо), так і цілочисельною величиною (кількість робочих циклів, запусків тощо).

**Довговічність (durability)** – властивість об'єкта виконувати потрібні функції до переходу у граничний стан при встановленій системі технічного обслуговування та ремонту.

**Ремонтпридатність (maintainability)** – властивість об'єкта бути пристосованим до підтримання та відновлення стану, в якому він здатний виконувати потрібні функції за допомогою технічного обслуговування та ремонту.

**Збережуваність (storability)** – властивість об'єкта зберігати в заданих межах значення параметрів, що характеризують здатність об'єкта виконувати потрібні функції, під час і після зберігання та (чи) транспортування.

### Показники безвідмовності

**Імовірність безвідмовної роботи  $P(t)$**  (reliability function; survival function) - імовірність того, що протягом заданого наробітку відмова об'єкта не виникне.

Примітка. За звичай вважають, що на початку інтервалу часу чи наробітку об'єкт у змозі виконувати потрібні функції:



$$P(t) = \frac{N_p}{N},$$

де  $N_p$  – кількість працездатних примірників ПЗ ( $N_p = N - n$ );  
 $n$  – кількість відмов.

**Ймовірність безвідмовної роботи  $P(t)$**  також визначається як функція розподілу ймовірностей безвідмовної роботи об'єкта, яка характеризує ймовірність того, що об'єкт за час  $t$  не втратить працездатності. Ймовірність безвідмовної роботи є одним із основних показників надійності технічних виробів. Значення  $P(t_1, t_2)$  за відносно короткі проміжки часу (година, доба) для сучасних електронних пристроїв близьке до 1. Наприклад, ймовірність безвідмовної роботи типової інтегральної схеми (ІС) за кілька годин роботи становить 0,999999. Проводити розрахунки з такими числами незручно тому доцільно перейти до відповідних ймовірностей відмови  $- 10^{-7} \dots 10^{-6}$ .

**Наробіток до відмови ( $t_{\text{сеп}}$ )** (operating time to failure) - наробіток об'єкта від початку експлуатації до виникнення першої відмови.

$$t_{\text{сеп}} = \frac{\sum_{i=1}^N t_i}{N}$$

де  $t_i$  – час роботи до відмови  $i$ -го примірника ПЗ;

$N$  – загальна кількість примірників ПЗ.

**Інтенсивність відмов;  $\lambda(t)$**  (instantaneous failure rate; failure rate) - умовна густина імовірності виникнення відмови об'єкта, яка визначається за умови, що до цього моменту відмова не виникла.

Примітка. Інтенсивність відмов є показником безвідмовності не ремонтпридатних і не відновлюваних об'єктів.

$$\lambda(t) = \frac{n(t)}{N_p \cdot \Delta t}$$

де  $N_p$  – кількість працездатних примірників ПЗ;

$n(t)$  – кількість відмов ПЗ за час  $\Delta t$ .

Практично для всіх систем інтенсивність відмов залежить від часу

має характеристику у вигляді “ванни” (рис. 1).

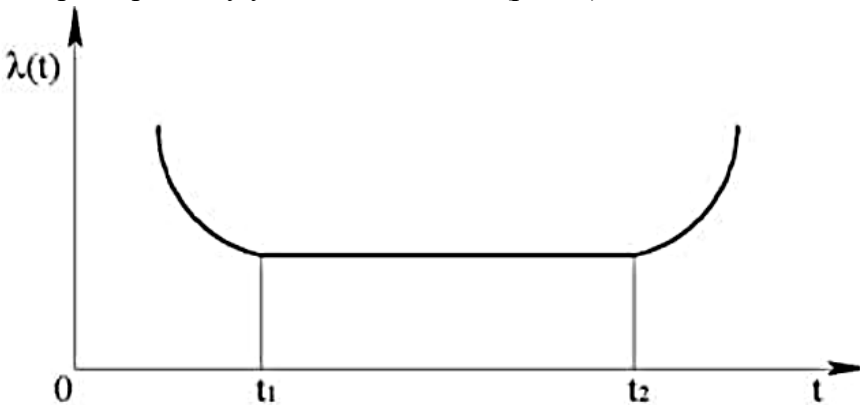


Рис. 2.1. Зміна інтенсивності відмов ПЗ:

0 –  $t_1$  - період припрацювання;

$t_1$ – $t_2$  - період нормальної експлуатації;

$t > t_2$  - період зношування та старіння.

### Показники довговічності

**Ресурс технічний ресурс** (useful life) - сумарний наробіток об'єкта від початку його експлуатації чи поновлення після ремонту до переходу в граничний стан.

$$T_{p.сер} = R_{сер} = \frac{1}{N} \cdot \sum_{i=1}^N R_i$$

де  $R_i$  – ресурс  $i$ -го примірника ПЗ;

$N$  – кількість примірників ПЗ.

**Гамма-відсотковий ресурс** – напрацювання, упродовж якого ПЗ не досягне граничного стану. Виражають його у відсотках.

**Середній термін служби** – середнє календарне напрацювання машини до граничного стану в роках:

$$T_{сл.сер} = \frac{1}{N} \cdot \sum_{i=1}^N T_{сл.i}$$

де  $T_{сл.i}$  – термін служби  $i$ -го примірника ПЗ.

Таким чином, термін служби від ресурсу відрізняється лише розмірністю.

**Технічний ресурс виробу** - прогнозована тривалість його експлуатації від початку до моменту, коли інтенсивність відмов збільшується до рівня, який робить подальшу експлуатацію неможливою або недоцільною з економічних міркувань.

## 2.2. Класифікація і характеристики відмов

**Відмова** – подія, яка полягає в утраті функційним модулем (системою) здатності виконувати потрібну функцію (згідно ДСТУ 2860-94).

Основна відмінність класифікації відмов ПЗ від відмов технічних засобів полягає в тому, що програмні відмови можуть бути тільки раптовими, оскільки працездатність ПЗ - це працездатність обчислювального процесу. В цьому випадку не застосовні показники старіння (ніяких фізико-хімічних процесів, що призводять до поступової відмови в програмних засобах не спостерігається).

Таким чином, при класифікації відмов ПЗ можна виділити п'ять основних ознак класифікації:

- залежність від відмов інших об'єктів (залежні і незалежні відмови);

- можливість подальшого використання об'єкта (повні та часткові відмови);
- наявність зовнішніх проявів відмови (явні і неявні відмови);
- можливість усунення відмови (самоусувні, переборні і непереборні відмови);
- причина виникнення відмови по етапах життєвого циклу (конструкційні, виробничі і експлуатаційні відмови).

**Незалежна відмова** - це відмова ПЗ, не обумовлений відмовою іншого об'єкту. **залежна відмова** - Відмова ПЗ, обумовлений відмовою іншого об'єкту. В якості такого об'єкта можуть виступати інше ПЗ, помилки у вхідних даних і обслуговуючий персонал. Наприклад, відмова операційної системи при відмові роботи деякої прикладної програми - залежний відмова. Відмова, обумовлений програмної помилкою в логіці виконання алгоритму програми - незалежний відмову.

При **повній відмові** ПЗ припиняє виконання всіх покладених на нього функцій, а при **частковій** - деякі функції ПЗ ще виконуються. Найбільш наочним прикладом є функціонування операційної системи: частина відмов не завжди призводить до скоєного зависання системи. Операційна система хоч і зі збоями продовжує виконувати частину своїх функцій.

**Причинами відмов** ПЗ є процеси, події чи стану, що зумовили виникнення відмови ПЗ. Залежно від причини виникнення відмови їх класифікують на

- **конструкційні** - відмови, що з'явилися в результаті недосконалості і порушення встановлених правил і (або) норм проектування і розробки ПЗ;

- **виробничі** - відмови, що виникли в результаті недосконалості або порушення встановленого процесу виготовлення (тиражування), установки (інсталяції) або супроводу ПЗ;

- *експлуатаційні* - відмови, що виникли в результаті порушення встановлених правил і (або) умов експлуатації ПЗ.

**Самоусувна відмова** - це відмова ПЗ, яка може бути усунута без рестарту (перезапуску) обчислювального процесу і проводиться самим ПЗ. *усувна* відмова дозволяє ліквідувати наслідки відмови при перезапуску обчислювального процесу, але вимагає для свого усунення втручання іншого ПЗ або користувача (обслуговуючого персоналу).

**Неперебірна відмова** зачіпає не тільки обчислювальний процес ПЗ, але і допоміжні дані (наприклад, файли конфігурації і т.д.), що призводить до неможливості подальшого використання даного ПЗ без його перевстановлення.

**Явна відмова** - відмова ПЗ, наслідки якого мають явні прояви.

При *неявній відмові* таких явних проявів не спостерігається.

**Перемежована відмова** - це відмова ПЗ одного і того ж характеру, яка виникає багаторазово.

### 2.3. Термінологія теорії надійності.

Окрім вище згаданих термінів, розглянемо ще деякі основні терміни і означення, що використовуються в теорії надійності згідно ДСТУ 2860-94. Надійність техніки. Терміни та визначення.

**Ремонтопридатність (властивість) (Instandhaltbarkeit)** - властивість об'єкта бути пристосованим до підтримання та відновлення стану, в якому він здатний виконувати потрібні функції за допомогою технічного обслуговування та ремонту

**Відмова (refuse)** – подія, яка полягає у втраті об'єктом здатності виконувати потрібну функцію, тобто у порушенні працездатного стану об'єкта.

**Критерій відмови (criterion of refuse)** – ознака чи сукупність ознак порушення працездатного стану об'єкта,

встановлені у нормативній та (або) конструкторській (проектній) документації.

**Граничний стан (maximum state)** – стан ПЗ, при якому його подальше застосування за призначенням стає неприпустимим чи недоцільним.

**Термін служби (unseful lifetime; lifetime)** - календарна тривалість експлуатації об'єкта від початку чи її поновлення після ремонту до переходу в граничний стан.

$$T_{сл.сер} = \frac{1}{N} \cdot \sum_{i=1}^N T_{сл.i}$$

де  $T_{сл.i}$  – термін служби  $i$ -го примірника ПЗ.

Таким чином, термін служби від ресурсу відрізняється лише розмірністю.

**Тривалість відновлення (time to recovery; lime io restoration)** - інтервал часу, протягом якого об'єкт перебуває в непрацездатному стані через відмову.

Конструктивно всі ПЗ можна розділити на невідновлювані та відновлювані.

**Невідновлюваними (unrefurbishable)** називають такі ПЗ, що у процесі виконання своїх функцій не можуть ремонтуватися, а **відновлювані** – ремонтуються. З огляду на цю властивість окремо розраховують і нормують показники надійності для відновлюваних та невідновлюваних ПЗ.

**Показник надійності (dependability measure)** - кількісна характеристика однієї чи декількох із тих властивостей, які в сукупності складають надійність об'єкта.

**Метрологічна справність (metrology good condition) технічного засобу (ТЗ)** – це стан ТЗ, що визначає відповідність його нормованих метрологічних характеристик встановленим вимогам.

**Метрологічна надійність (metrology reliability) ТЗ** – це надійність ТЗ в частині збереження його метрологічної справності.

**Метрологічна відмова (metrology refuse) ТЗ** – це відмова ТЗ, що полягає у втраті його метрологічної справності.

**Нестабільність (instability) метрологічної характеристики ТЗ** – це зміна метрологічної характеристики ТЗ за встановлений інтервал часу.

**Довірчі межі (confiding limits) нестабільності метрологічної характеристики ТЗ** – це верхня і нижня межі інтервалу, що охоплює нестабільність метрологічної характеристики ТЗ з деякою довірчою вірогідністю.

**Вірогідність (authenticity) метрологічної справності ТЗ** – це вірогідність того, що в заданий момент часу ТЗ виявиться метрологічно справним.

**Середній час (середнє напрацювання) до метрологічної відмови ТЗ** – це математичне сподівання календарного часу експлуатації (напрацювання) ТЗ до першої метрологічної відмови.

**Напрацювання на метрологічну відмову ТЗ** – це відношення сумарного напрацювання ТЗ в стані метрологічної справності на заданий період експлуатації до математичного сподівання числа його метрологічних відмов за цей період.

### 3. Класифікація технічних систем

#### 3.1. Проблема побудови природної систематики техніки

Необхідність систематики виникає у всіх науках. Найдосконалішою систематикою є природна система об'єктів, заснована на об'єктивних законах. Систематика повинна спиратися на виділення в об'єктах деяких стійких характеристик: ознак, властивостей, функцій, зв'язків. В ідеалі систематика повинна ґрунтуватися на найголовніших фундаментальних властивостях і ознаках об'єктів, законах їх будови і розвитку. Прикладом і еталоном природної систематики може слугувати, наприклад, Періодична система елементів у хімії. Ще одна важлива властивість природної систематики об'єктів полягає в її єдності, тобто звичайно існує одна науково-обґрунтована і загально-визнана систематика.

Слід зазначити, що разом із природною систематикою об'єктів існують ще штучні класифікації об'єктів, які створюють із міркувань корисності і зручності в рішенні певних практичних завдань. Частіше всього це завдання пошукового характеру, а класифікація об'єктів може їх упорядкувати, наприклад, за абеткою. Разом із цим є також класифікації об'єктів із солідним науковим обґрунтуванням.

Такі класифікації займають проміжне положення між довільними штучними класифікаціями і систематикою об'єктів. Приклад природної систематизації технічних систем приведений у табл. 3.1. Тут дано тільки два рівні ієрархії. Побудова повної класифікації є нелегким завданням, а побудова природної систематики техніки – ще більш важка проблема.

Проблема побудови систематики техніки є надзвичайно важливою самостійною проблемою, актуальність якої безперервно зростає.



Практичне значення цієї проблеми зв'язано, по-перше, із створенням єдиної науково-обгрунтованої класифікації патентів і стандартів. Ще більшу актуальність має проблема систематики техніки останнім часом, у зв'язку із створенням національних і міжнародних активних інформаційних ресурсів автоматизованих банків даних і банків знань на електронних носіях [13].

Таблиця 3.1.

Класифікація технічних систем відповідно до загальної ієрархії техніки

Група <i>ТС</i>	Підгрупа <i>ТС</i>	Приклади <i>ТС</i>
1. Техніка для видобування природних речовин	1.1 Техніка для видобутку корисних копалин 1.2 Техніка для лісозаготівлі і первинної обробки деревини 1.3 Сільськогосподарська техніка	Вугільний комбайн, навантажувач, землесосний снаряд, зернозбиральний комбайн, трактор, сівалка
2. Техніка для отримання, передачі, перетворення і використання енергії	2.1 Техніка для виробництва первинної енергії 2.2 Техніка для отримання вторинної (електричної) енергії 2.3 Техніка для передачі трансформації енергії 2.4 Техніка, що споживає енергію	Піч. паровий котел, парова турбіна, гідротурбіна, генератор, трансформатор, трубопровід, радіатор, електричний двигун. електрична лампа
3. Техніка для переробки сировини, здобутої в природі	3.1 Техніка для металургійного переділу 3.2 Техніка хімічного виробництва 3.3 Техніка для виробництва будівельних матеріалів 3.4 Техніка легкої промисловості	Дробарка, доменна піч. мартенівська піч. автоклав, реактор. змішувач, швейна машина, млин. центрифуга, прес

## Продовження таблиці 3.1.

Група <i>ТС</i>	Підгрупа <i>ТС</i>	Приклади <i>ТС</i>
	3.5 Техніка для переробки продуктів харчування 3.6 Техніка для деревообробки 3.7 Техніка фармацевтичного виробництва	
4. Техніка для виробництва технічних систем	4.1 Техніка для виробництва засобів виробництва 4.2 Техніка для виробництва предметів споживання 4.3 Техніка для будівництва 4.4 Техніка для виробництва озброєнь	Токарний верстат, фрезерний верстат, прес, формувальна машина, конвеєр, підйомний кран, бетономішалка, рентгенівський апарат
5. Техніка для транспортування	5.1 Залізничний транспорт 5.2 Автомобільний транспорт 5.3 Водний транспорт 5.4 Повітряний транспорт 5.5 Спеціальні види транспортування	Судно, літак, ракета, нафтопровід, стрічковий конвеєр, підвісна канатна дорога
6. Техніка для захисту і (або) ураження	6.1 Засоби індивідуального захисту і враження 6.2 Військова техніка для дії на суші 6.3 Військова техніка для дії на морі 6.4 Військова техніка для дії в повітрі 6.5 Спеціальні види озброєнь	Бронежилет, автомат, танк, авіаносець, винищувач - бомбардувальник. ядерна зброя, крилата ракета
7. Техніка для отримання,	7.1 Вимірювальна техніка 7.2 Техніка для виробництва	Нівелір, далекомір, персональний

## Продовження таблиці 3.1.

обробки і передачі інформації	інформації 7.3 Техніка для обробки інформації 7.4 Техніка для зберігання інформації 7. Техніка для передачі інформації	комп'ютер, книга, компакт-диск, радіостанція, телевізор, телескоп
8. Техніка для задоволення духовних потреб	8.1 Техніка для навчання 8.2 Техніка для фізичного розвитку 8.3 Техніка для розваг та відпочинку	Модель екскаватора, діапроектор, чучело тварини, тренажер

Класифікація технічних систем з різних визначальних ознак вносить достатньо рівний порядок в їх загальну множину і дозволяє краще орієнтуватися. Як наслідок цього з'являється можливість вивчення передового досвіду, що дозволяє часом знайти між досить далекими технічними системами цікаві, досі приховані відносини.

Перерахуємо спершу ті аспекти, які беруться до уваги при проведенні класифікації (систематизації) технічних систем, і приведемо характерні приклади. Технічні системи можуть бути класифіковані за такими ознаками [5]:

- за функцією (робочою дією), наприклад, технічні системи для транспортування, підтримки, фіксації, буріння, підйому;
- за типом операцій, наприклад, технічні системи для перетворення матерії, енергії, інформації, біологічних об'єктів;
- за принципом здійснення робочої дії, наприклад, технічні системи, засновані на механічному, гідравлічному,

пневматичному, електронному, хімічному, оптичному, акустичному принципі;

- за характером функціонування, наприклад, потужні, швидкісні, імпульсні технічні системи, системи для різних умов навколишнього середовища (наприклад, для тропічного клімату) і т. п.;

- за рівнем складності, наприклад, конструктивні елементи, вузли, машини, підприємства в цілому;

- за способом виготовлення, наприклад, технічні системи, виготовлені шляхом виливання, кування, штампування, обточування;

- за формою, наприклад, технічні системи (конструктивні елементи) у вигляді тіла обертання, плоскі, складної форми;

- за способом упорядкування більш низьких рівнів технічної системи, наприклад, установки з упорядкуванням підсистем за способами їх дії або технології;

- за матеріалом, наприклад, технічні системи з металів, пластмаси, деревини, комбіновані;

- за ступенем оригінальності конструкції, наприклад, запозичені технічні системи, допрацьовані, модифіковані, оригінальні;

- за типом виробництва, наприклад, технічні системи, виготовлені в умовах одиничного, серійного або масового виробництва;

- за місцем у технічному процесі, за експлуатаційними властивостями, за зовнішнім виглядом, за техніко-економічними характеристиками і т.п.

Ясно, що одна і та ж технічна система може належати одночасно до декількох класів. Нижче більш детально будуть розглянуті ті принципи класифікації технічних систем, які, із погляду проектувальника і конструктора, є особливо важливими.

### **3.2. Класифікація технічних систем за функцією та принципом дії**

Назви технічних систем часто вибираються відповідно до їх функції. Складання номенклатури виробів стосовно вимог збуту, планування, контролю, порівняльної оцінки то що також здійснюється, як правило, відповідно до функції технічних систем. Межею у взаєморозумінні фахівців є збіг класифікації і термінології у всіх галузях науки і техніки [5]. Вироби класифікуються за функцією також у тих випадках коли необхідно допомогти потенційному споживачу знайти той або інший технічний засіб для виконання необхідної функції. Цьому слугують торгові і промислові каталоги і т.п.

Вузли і деталі машин часто також можна розглядати як технічні системи, класифікацію доцільно проводити за функцією, оскільки конструктор, виробничник і експлуатаційник застосовують різні деталі відповідно до їх функціональної придатності.

Таку класифікацію називають конструктивно-функціональною; разом з класифікацією за способом виготовлення вона є основною при запозиченні існуючих технічних систем, уніфікації, типізації і стандартизації елементів і груп. Класифікація за цими принципами дозволяє економити робочий час конструктора.

Для розробника, у тому числі ПЗ, важливо, щоб технічні системи, що виконують виробничі функції, були далі згруповані за ще якою-небудь важливою ознакою. Такою ознакою можна вважати принцип дії технічної системи. Такого роду ознаки технічних систем відносяться переважно до групи функціонально обумовлених властивостей, вельми характерних для технічних

систем і мають велике значення для методичної роботи конструктора.

Щодо розробки ПЗ то тут класифікацію проводити досить складно, виходячи з того факту, що мови програмування, методики створення програм, методи їх перевірки надзвичайно динамічно змінюються. Однак, разом з цим, існують міжнародні стандарти, наприклад для створення технічного опису програмного продукту Software Design Descriptions, IEEE Standard 1016-2009 та IEEE Recommended Practice for Architectural Description of Software-Intensive Systems, IEEE Standard 1471-2000, затверджені у 2000 та 2009 роках відповідно.

Про що це говорить?

Основні, базові визначення, порядок дій при створенні ПЗ, залишаються не змінними. Що найменше з 2009 року. Змінюються лише способи виконання завдань.

### **3.3. Класифікація технічних систем за рівнем складності**

Розподіл технічних систем на класи за їх структурою – звичайна справа в роботі розробника ПЗ. Основною ознакою, за якою утворюються класи, повинна служити функція системи. Проте, враховуючи потреби виробництва, наприклад з міркувань монтажу, деколи виникає необхідність у проведенні іншої класифікації. Іноді через організаційні причини технічні системи доцільно розділяти на підсистеми.

Таблиця 3.2 дає загальне уявлення про класифікацію технічних систем за рівнем складності, визначуваних рівнями складності її частин. На більш високих рівнях складності можна розрізняти ще і проміжні рівні.

Проте слід пам'ятати, що йдеться про відносну ієрархію. Одна і та ж система більш низького рівня, наприклад редуктор

або ґрунтовий насос, в одній системі може розглядатися як підгрупа, а в іншій системі - як група або машина (підсистема).

На практиці загальноновизнано, що нижні рівні технічних систем знаходять більш універсальне вживання. Наприклад, елемент «гайка» застосовується в машинобудуванні повсюдно, «редуктор» досить часто, а «технологічна лінія» використовується лише в певних, спеціальних процесах.

Таблиця 3.2

## Класифікація технічних систем за рівнем складності

Рівень скл.	Технічна система	Характеристика	Приклади
I	Конструктивний елемент Деталь машини	Елементарна система, виготовлена без монтажних операцій	Гайка, кришка, шпонка
II	Підгрупа: Група; Вузол: Механізм	Проста система, що виконує нескладну функцію	Редуктор, насос, пневматичний привід
III	Машина, Прилад, Апарат	Система, що складається з груп і елементів і виконує певну функцію	Амперметр, вольтметр.
IV	Установка Підприємство Промисловий комплекс	Складна система, що складається з машин, груп і елементів, яка виконує ряд функцій	Технологічна лінія. Приладобудівне підприємство

Класифікація технічних систем за рівнем складності має важливе значення для розробника ПЗ як конструктора, оскільки рівень складності технічної системи:

а) знаходиться в певному співвідношенні із ступенем складності рішення поставленого перед розробником завдання;

б) припускає встановлення відомих меж для спеціалізації розробника (наприклад, інженер-проектувальник має справу з підприємством, інженер-конструктор – із машиною, конструктор деталей - з елементами машини);

в) допомагає розробнику орієнтуватися в процесі роботи, бо, якщо він вирішує завдання на якомусь певному рівні складності, йому важливо знати лише те, як його завдання злагоджено із більш високим рівнем (відносно більш низького рівня розробник ухвалює частіше за все тільки принципові рішення).

На підставі складального креслення окремі рівні складності можна розглядати так само, як сукупності процесів виготовлення і монтажу. Утворення відповідних сукупностей, перш за все з деталей, підгруп і груп, є необхідною умовою створення модульних конструкцій, а також доцільної організації виробничого процесу.

### **3.4. Класифікація технічних систем за способом виготовлення**

Для виготовлення певних груп технічних систем потрібне однотипне технологічне обладнання. Наприклад, на одному і тому ж обладнанні (комп'ютері) можна змодельовати відцентрові насоси і компресори, на іншому - електродвигуни. При проектуванні деталей машин у 3D форматі їх можна також звести в технологічні групи за принципом схожості технологічних операцій виготовлення, де головною відмітною ознакою служить форма.



Така класифікація дозволяє раціонально провести підготовку виробництва і підвищити ефективність виробничого процесу оскільки дає можливість об'єднати робочі місця для виготовлення однакових за способом виготовлення деталей. Це у свою чергу полегшує здійснення самих різних заходів раціоналізації, наприклад спеціалізацію робочих цехів, підприємств. Значення такої класифікації особливо велике при моделюванні, розробці і здійсненні планів підготовки виробництва, методів управління і планування. Вона є складовою частиною так званої групової технології обробки.

### **3.5. Класифікація технічних систем за ступенем оригінальності**

При розробці нового ПЗ розробник завжди прагне використовувати в технології та допоміжне ПЗ, що виправдали себе на практиці. За ступенем оригінальності ПЗ технічні системи можна розділити на категорії, що подані нижче.

#### ***Запозичене ПЗ***

Для виконання необхідної функції вже існує яка-небудь програмна система або навіть декілька систем, із яких можуть бути вибрані відповідні. До них відносяться в першу чергу уніфіковані елементи і групи, а також не уніфіковані елементи і групи, які можуть запозичати з іншого ПЗ.

#### ***Допрацьоване ПЗ***

У наявності є яка-небудь технічна система (ПЗ), що виконує необхідну функцію, але не відповідає деяким вимогам. Виникає потреба, наприклад, змінити частоту запитів, ресурсомісткість, швидкодію, тощо. Структури системи і найважливіші властивості елементів в цьому випадку залишаються без зміни. Таким чином,

добробка ПЗ проводиться виключно з метою пристосування її до особливих умов і вимог нового завдання.

### ***Модифіковане ПЗ***

Існуючі системи не відповідають вимогам, що пред'являються до деяких властивостей груп і елементів ПЗ. У модифікованій версії звичайно не змінюються лише функція, деякі параметри і по можливості принцип дії. В елементах можуть бути змінені форма, об'єм або технологія програмування, в складних системах змінюються органоструктура і алгоритмічна схема, тобто деякі елементи і групи, їх з'єднання і розміщення в дереві програми.

Звичайно модифікація здійснюється шляхом переробки ПЗ.

### ***Нові технічні системи***

Для виконання бажаної функції відсутнє ПЗ або ж існуюче має недоліки принципового характеру. Необхідно ПЗ з новим принципом дії і іншими технічними властивостями.

## **3.6. Класифікація технічних систем за типом виробництва**

Тип виробництва, який визначається кількістю одиниць продукції, що виготовляються, додає кожному виробу ряд характерних технічних і економічних властивостей.

### ***ПЗ одиничного виробництва.***

У цьому випадку роботи по створенню ПЗ необхідно пристосувати до потреб поштучного виробництва ПЗ, в умовах якого вартість кожної одиниці ПЗ збільшується. Не виключено, що в умовах одиничного виробництва необхідна функція ПЗ взагалі не буде досягнута оскільки при виготовленні крупних програмних проєктів доводиться працювати без прототипу. От

чому ця категорія систем пред'являє високі вимоги до розробника ПЗ.

### ***ПЗ серійного або масового виробництва***

Таке ПЗ в цілому краще пропрацьовано з погляду виробництва (тиражування). Через великий об'єм партій виробів частка розробницьких витрат по відношенню до загальних витрат невелика. Проте оскільки контролю піддається, як правило, лише невелика частина копій ПЗ, то не виключені різні похибки і дефекти. Тільки при здійсненні безперервного контролю за всіма операціями тиражування ПЗ, у цілому можна добитися стабільної якості при серійному і масовому виробництвах.

### **3.7. Класифікація ПЗ за його місцем у технічному процесі**

Ця класифікація має мало спільного з іншими класифікаційними принципами, особливо з принципом складності, і підрозділяє ПЗ (технічні системи) лише на крайові і внутрішні (проміжні).

Крайові системи, виступаючі в технічному процесі як оператори, – це переважно ПЗ, що виконують дію. Вони утворюють з оператором пару взаємодії. Внутрішнє (проміжне) ПЗ, тільки при спільній роботі з крайовими системами може здійснювати необхідну дію на оператора.

## 4. Програмне забезпечення та його класифікація

### 4.1. Класифікація програмного забезпечення

Сучасне програмне забезпечення різнопланове. Його можна класифікувати на такі види (рис. 4.1):

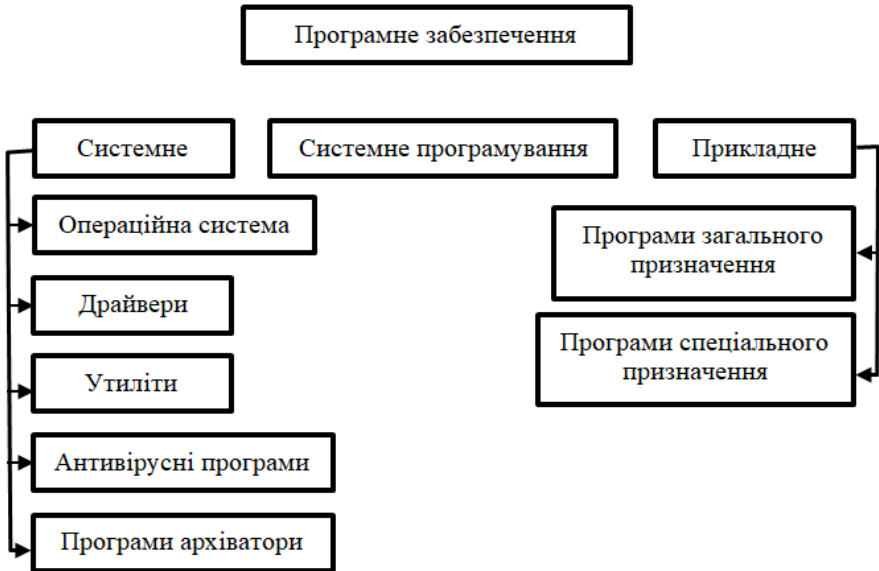


Рис. 4.1. Види програмного забезпечення

### 4.2. Системне програмне забезпечення

**Системне програмне забезпечення** призначене для керування роботою складових комп'ютера та обміном даними між ними, діагностування та усунення недоліків у роботі комп'ютера, автоматизації процесу обробки даних, організації обміну даними між користувачем і комп'ютером.

Системне (базове) програмне забезпечення слугує для організації й ефективної роботи комп'ютера та охоплює:

- операційні системи (ОС), основною функцією яких є

керування ресурсами (фізичними та логічними) і процесами обчислювальних систем. Існують, як відомо такі ОС, як Microsoft Windows 95 (98, 2000, XP, 7, 10), Windows NT, Unix, OS/2;

- мережеве програмне забезпечення, призначене для керування спільними ресурсами в розподілених обчислювальних системах (Novell NetWare 6.5 SP8, Windows Server, Windows Web Server 2010 R2, SQL Server Workgroup Edition, Sun Solaris);

- сервісні програми, до яких належать файлові менеджери (наприклад, Total Commander 7.55); утиліти, тобто системні програми для виконання службових функцій (антивіруси, архіватори, програми для обслуговування дисків тощо). Ці програми створюють додаткові зручності під час роботи користувача на комп'ютері.

## Операційна система

Класифікація ОС наведена на рисунку 4.2.



Рис. 4.2. Класифікація ОС

Операційна система – це комплекс програм, що забезпечують:

- керування роботою пристроїв комп'ютера та обмін даними між ними;
- зберігання даних в оперативній пам'яті та на зовнішніх носіях;
- виконання інших програм;
- розподіл ресурсів комп'ютера між окремими програмами, які працюють одночасно;
- організацію обміну даними між користувачем і комп'ютером.

У сучасних комп'ютерах використовуються операційні системи Windows, Linux, Unix, MacOS, Netware, Palm OS та ін.

### **Класифікація сервісних програмних засобів**

1. **Диспетчери файлів** (файлові менеджери). За їх допомогою виконується більшість операцій по обслуговуванню файлової структури копіювання, переміщення, перейменування файлів, створення каталогів (папок), знищення об'єктів, пошук файлів та навігація у файловій структурі. Базові програмні засоби містяться у складі програм системного рівня і встановлюються разом з операційною системою

2. **Засоби стиснення даних** (архіватори). Призначені для створення архівів. Архівні файли мають підвищену щільність запису інформації і відповідно, ефективніше використовуються носії інформації.

3. **Засоби діагностики**. Призначені для автоматизації процесів діагностування програмного та апаратного забезпечення. Їх використовують для виправлення помилок і для оптимізації роботи комп'ютерної системи.

4. **Програми інсталяції** (встановлення). Призначені для контролю за додаванням у поточну програмну конфігурацію нового програмного забезпечення. Вони слідкують за станом і зміною оточуючого програмного середовища, відслідковують та

протоколюють утворення нових зв'язків, загублені під час знищення певних програм. Прості засоби управління встановленням та знищенням програм містяться у складі операційної системи, але можуть використовуватись і додаткові службові програми.

**5. Засоби комунікації.** Дозволяють встановлювати з'єднання з віддаленими комп'ютерами, передають повідомлення електронної пошти, пересилають факсимільні повідомлення тощо.

**6. Засоби перегляду та відтворення.** Переважно для роботи з файлами, їх необхідно завантажити у "рідну" прикладну систему і внести необхідні виправлення. Але, якщо редагування не потрібно, існують універсальні засоби для перегляду (у випадку тексту) або відтворення (у випадку звука або відео) даних.

**7. Засоби комп'ютерної безпеки.** До них відносяться засоби пасивного та активного захисту даних від пошкодження, несанкціонованого доступу, перегляду та зміни даних. Засоби пасивного захисту - це службові програми, призначені для резервного копіювання. Засоби активного захисту застосовують антивірусне програмне забезпечення. Для захисту даних від несанкціонованого доступу, їх перегляду та зміни використовують спеціальні системи, базовані на криптографії.

### **4.3. Прикладне програмне забезпечення**

**Прикладне програмне забезпечення** — це програми, що призначені для реалізації конкретних задач опрацювання даних, які користувач розв'язує в ході своєї діяльності. Їх поділяють на прикладні програми загального і спеціального призначення.

До прикладних програм загального призначення відносять програми, які можуть застосовуватися в різних галузях людської

діяльності для опрацювання текстів, малюнків, баз даних, електронних таблиць, створення презентацій тощо.

Прикладні програми спеціального призначення використовуються для реалізації завдань опрацювання даних у певній галузі діяльності, на конкретному підприємстві, в організації, фірмі або їх підрозділі. До такого типу програм відносять програми для створення відеоефектів при виробництві кінофільмів, креслень машин і механізмів у конструкторських і проектних бюро, діагностування захворювань у медичних закладах, створення шкільного розкладу уроків тощо.

**До прикладних програм загального призначення належать:**

- **текстові процесори** (Microsoft Word, WordPad, Блокнот, OpenOffice.org Writer та ін.);
- **табличні процесори** (Microsoft Excel, Open Office Calc, IBM Lotus Symphony тощо);
- **системи ілюстративної і ділової графіки та видавничі системи** (Corel Draw, PageMaker, Adobe Photoshop, Adobe Acrobat, Macromedia Flash, QuarkXPress);
- **системи управління базами даних** (MySQL, Microsoft SQL Server 2008, Paradox, Microsoft Access, Oracle та ін.);
- **програми створення презентацій** (Microsoft Power Point, Quick Slide Show, MySlideShow та ін.);
- **програми розпізнавання символів** (Scanitto Pro, Fine Reader);

До прикладних програм спеціального призначення належать:

- **системи ведення бухгалтерського обліку** (1С-бухгалтерія та ін.);
- **правові БД** (Лига, Право);
- **експертні системи** (OpenCyc, WolframAlpha та ін.);



• **системи автоматизованого проектування** (AutoCAD, Mathcad тощо) та інші.

#### 4.4. Системи програмування

Інструментальні мови та **системи програмування** служать для розробки нових програм. Комп'ютер "розуміє" і може виконувати програми у машинному кодї. Кожна команда при цьому має вигляд послїдовностї нулїв й одиниць. Писати програми машинною мовою дуже незручно, а їх надїйность низка. Тому програми розробляють мовою, зрозумїлою людинї (їнструментальна мова або алгоритмїчна мова програмування), пїсля чого спецїальною програмою, яка називається **транслятором**, текст програми перекладається (транслюється) на машинний код.

Транслятори бувають двох типїв:

- інтерпретатори;
- компїлятори.

**Інтерпретатор** читає один оператор програми, аналізує його і вїдразу виконує, пїсля чого переходить до оброблення наступного оператора.

**Компїлятор** спочатку читає, аналізує та перекладає на машинний код усю програму і тїльки пїсля завершення всїєї трансляцїї ця програма виконується. Інструментальні мови подїляються на мови низького рївня (близькї до машинної мови) та мови високого рївня (близькї до мови людини).

До мов низького рївня належать асемблери, а високого - Pascal, Basic, C/C++, мови баз даних і т.д. Систему програмування, крїм транслятора, складають текстовий редактор, компонувальник, бїбліотека стандартних програм, налагоджувач, вїзуальнї засоби автоматизацїї програмування. Прикладами таких систем є Delphi, Visual Basic, Visual C++, Visual FoxPro та їн.

## 5. Критерії надійності

Критерієм надійності називається ознака, за якою можна кількісно оцінити надійність різних пристроїв. В тому числі і надійність програмного забезпечення.

На початку розглянемо загальні елементи теорії та критеріїв надійності довільних технічних систем.

До числа найбільш широко застосовуваних критеріїв надійності, яке було наведено раніше, відносяться:

- ймовірність безвідмовної роботи протягом певного часу  $P(t)$ ;
- середнє напрацювання до першої відмови  $T_{cp}$ ;
- напрацювання на відмову  $t_{cp}$ ;
- частота відмов  $p(t)$ ;
- інтенсивність відмов  $\lambda(t)$ ;
- параметр потоку відмов  $f(t)$ ;
- функція готовності  $Kz(t)$ ;
- коефіцієнт готовності  $Kz$ .

Характеристикою надійності слід називати кількісне значення критерію надійності конкретного пристрою.

Вибір кількісних характеристик надійності залежить від виду об'єкта.

### 5.1. Критерії надійності невідновних систем

Розглянемо наступну модель роботи пристрою.

Нехай у роботі (на випробуванні) знаходиться  $N_0$  елементів і робота вважається закінченою, якщо всі вони відмовили. Причому замість зламаних елементів відремонтовані не ставляться. Тоді критеріями надійності даних виробів є:

- ймовірність безвідмовної роботи  $P(t)$ ;
- частота відмов  $f(t)$  або  $p(t)$ ;
- інтенсивність відмов  $\lambda(t)$ ;
- середнє напрацювання до першої відмови  $T_{cp}$ .

В цьому випадку ймовірністю безвідмовної роботи називається ймовірність того, що за певних умов експлуатації в заданому інтервалі часу або в межах заданого напрацювання не відбудеться жодної відмови.

Згідно з визначенням:

$$P(t) = P(T > t), \quad (5.1)$$

де:  $T$  - час роботи елемента від його включення до першої відмови;

$t$  - час, протягом якого визначається ймовірність безвідмовної роботи.

Ймовірність безвідмовної роботи за статистичними даними про відмови оцінюється виразом:

$$P(t) = [N_0 - n(t)] / N_0, \quad (5.2)$$

де:  $N_0$  - число елементів на початку роботи (випробувань);

$n(t)$  - число елементів, що відмовили за час  $t$ ;

$P(t)$  - статистична оцінка ймовірності безвідмовної роботи.

При значному числі елементів (виробів)  $N_0$  статистична оцінка  $P(t)$  практично збігається з ймовірністю безвідмовної роботи  $P(t)$ . На практиці іноді більш зручною характеристикою є ймовірність відмови  $Q(t)$ .

Ймовірністю відмови називається ймовірність того, що за певних умов експлуатації в заданому інтервалі часу виникає хоча

б одна відмова. Відмова та безвідмовна робота є подіями несумісними і протилежними, тому:

$$\begin{aligned} Q(t) &= P(T \leq t), \\ Q(t) &= n(t) / N_0, \\ Q(t) &= 1 - P(t), \end{aligned} \quad (5.3)$$

Частотою відмов за статистичними даними називається відношенням числа зруйнованих елементів в одиницю часу до первинного числа працюючих (випробовуваних) елементів, за умови, що всі вироби, що вийшли з ладу, не відновлюються.

Згідно з визначенням:

$$f(t) = n(\Delta t) / N_0 \Delta t, \quad (5.4)$$

де:  $n(\Delta t)$  - число елементів, що відмовили, в інтервалі часу від  $(T - \Delta t) / 2$  до  $(t + N_t) / 2$ .

Частота відмов є щільність ймовірності (або закону розподілу) часу роботи виробу до першої відмови. Тому:

$$P(t) = 1 - Q(t), P(t)f(t)dt, \quad (5.5)$$

Інтенсивністю відмов за статистичними даними називається відношення числа зламаних виробів в одиницю часу до середнього числа виробів, справно працюючих в даний відрізок часу.

Згідно з визначенням

$$\lambda(t) = n(\Delta t) / (N_{cp} \Delta t), \quad (5.6)$$

де:  $N_{cp} = N_i + N_i + (I) / 2$  - середня кількість справно працюючих елементів в інтервалі часу  $t$ ;

$N_i$  - число виробів, справно працюючих на початку інтервалу часу  $t$ ;

$N_i + I$  - число елементів, справно працюючих в кінці інтервалу часу  $t$ .

Імовірнісна оцінка характеристики  $p(t)$  знаходиться з виразу:

$$\lambda(t) = p(t) / P(t), \quad (5.7)$$

Середнім напрацюванням до першої відмови називається математичне очікування часу роботи елемента до відмови.

Як математичне очікування,  $T_{cp}$  обчислюється через частоту відмов (щільність розподілу часу безвідмовної роботи).

Для визначення середнього напрацювання до першої відмови необхідно знати моменти виходу з ладу всіх випробовуваних елементів. Тому для обчислення середнього напрацювання на відмову користуватися зазначеної формулою незручно. Маючи дані про кількість відмов елементів ПЗ  $n_i$  в кожному  $i$ -му інтервалі часу, середнє напрацювання до першої відмови краще визначати з рівняння статичної ймовірності (5.2).

З виразів для оцінки кількісних характеристик надійності показано, що всі характеристики, крім середнього напрацювання до першої відмови, є функціями часу.

Розглянуті критерії надійності дозволяють досить повно оцінити надійність невідновлювальних продуктів. Вони також дозволяють оцінити надійність відновлюваних продуктів до першої відмови. Наявність декількох критеріїв зовсім не означає, що завжди потрібно оцінювати надійність елементів за всіма критеріями.

Найбільш повно надійність характеризується частотою відмов  $p(t)$ . Це пояснюється тим, що частота відмов є щільністю розподілу, а тому несе в собі всю інформацію про випадкове явище - час безвідмовної роботи. Середнє напрацювання до першої відмови є досить наочною характеристикою надійності. Однак застосування цього критерію для оцінки надійності складної системи (складного ПЗ) обмежена в тих випадках, коли:

- час роботи системи набагато менше середнього часу безвідмовної роботи;

- закон розподілу часу безвідмовної роботи не однопараметричний і для достатньо повної оцінки потрібні моменти вищих порядків;
- система резервована;
- інтенсивність відмов зовсім постійна;
- час роботи окремих частин складної системи різний.

Однопараметричний закон з постійною інтенсивністю відмов ( $\lambda_0 = \text{const}$ ) називають Експоненціальним законом розподілу. Наприклад, ймовірність безвідмовної роботи згідно цього розподілу має експоненціальну залежність:

$$P(t) = e^{-\lambda_0 t}$$

Інтенсивність відмов - найбільш зручна характеристика надійності найпростіших елементів, так як вона дозволяє більш просто обчислювати кількісні характеристики надійності складної системи.

Найбільш доцільним критерієм надійності складної системи є ймовірність безвідмовної роботи. Це пояснюється такими особливостями ймовірності безвідмовної роботи:

- вона входить в якості складової до інших, більш загальних характеристик системи, наприклад, в ефективність і вартість;
- характеризує зміну надійності в часі;
- може бути отримана порівняно просто розрахунковим шляхом в процесі проектування системи і оцінена в процесі її випробування.

## 5.2. Критерії надійності відновлюваних об'єктів

Розглянемо наступну модель роботи на прикладі механічної системи.

Нехай в роботі знаходиться  $N$  елементів і елементи, що відмовили, негайно замінюються справними (новими або

відремонтованими). Якщо не враховувати часу, потрібного на відновлення системи, то кількісними характеристиками надійності можуть бути параметр потоку відмов -  $f(t)$  і напрацювання на відмову  $t_{cp}$ .

Параметром потоку відмов називається співставлення числа виробів що відмовили в одиницю часу до числа випробовуваних за умови, що всі вироби, які вийшли з ладу, замінюються справними (новими або відремонтованими).

Статистичним визначенням служить вираз:

$$\omega(t) = n(\Delta t)/N\Delta t, \quad (5.8)$$

де:  $n(\Delta t)$  - число зразків, які відмовили в інтервалі часу від

$t - f(t)/2$  до  $t + \Delta t/2$ ;

$N$  - число випробовуваних елементів;

$\Delta t$  - інтервал часу.

Параметр потоку відмов володіє наступними важливими властивостями:

1) для будь-якого моменту часу, незалежно від закону розподілу часу безвідмовної роботи, параметр потоку відмов більше, ніж частота відмов, т.т.  $f(t) > p(t)$ ;

2) незалежно від виду функцій  $f(t)$  параметр потоку відмов  $f(t)$  при  $t \rightarrow \infty$  прагне до  $1/T_{cp}$ . Це важлива властивість параметра потоку відмов означає, що при тривалій експлуатації ремонтного виробу потік його відмов, незалежно від закону розподілу часу безвідмовної роботи, стає стаціонарним. Однак це зовсім не означає, що інтенсивність відмов є величина постійна;

3) якщо  $\Delta(t)$  - зростаюча функція часу, то  $\Delta(t) > p(t) > f(t)$ , якщо  $p(t)$  - спадна функція, то  $p(t) > \Delta(t) > f(t)$ ;

4) при  $\lambda(t) \neq const$  параметр потоку відмов системи не дорівнює сумі параметрів потоку відмов елементів. Ця властивість параметра потоку відмов дозволяє стверджувати, що при обчисленні кількісних характеристик надійності складної

системи не можна підсумувати наявні в даний час значення інтенсивності відмов елементів, отриманих за статистичними даними про відмови виробів в умовах експлуатації, так як зазначені величини є фактично параметрами потоку відмов;

5) при  $\Delta(t) = p = const$  параметр потоку відмов дорівнює інтенсивності відмов.

З перегляду властивостей інтенсивності та параметра потоку відмов видно, що ці характеристики різні. В даний час широко використовуються статистичні дані про відмови, отримані в умовах експлуатації обладнання. При цьому вони часто обробляються таким чином, що приводяться характеристики надійності не є інтенсивністю відмов, а параметром потоку відмов.



## 6. Закони розподілу часу до відмови [9]

### 6.1. Життєвий цикл технічного засобу

Будь-який технічний засіб, у тому числі програмний продукт, з початку і до кінця експлуатації має три найбільш характерних періоди роботи:

- 1) припрацювання ( $0 < t < t_1$ );
- 2) нормальна експлуатація ( $t_1 < t < t_2$ );
- 3) старіння чи знос ( $t > t_2$ ) (рис. 6.1).

**Період припрацювання** характеризується високою інтенсивністю відмов, викликаних відхиленням від вимог документації, що розподіляються за законом розподілу Вейбулла й усуваються за рахунок введення технологічного припрацювання («технологічного прогону»). Як видно з рис. 6.1 інтенсивність відмов на першому періоді монотонно зменшується.

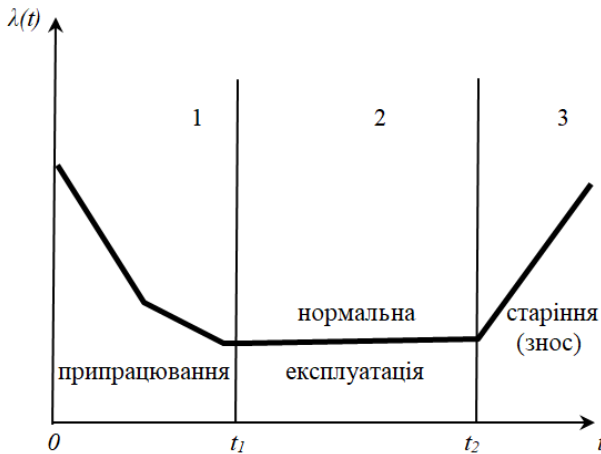


Рисунок 6.1 – Залежність інтенсивності відмов від часу роботи ТЗ

**Період нормальної експлуатації** характеризується мінімальною і постійною інтенсивностями відмов. Ці відмови називаються **раптовими**, носять випадковий характер і розподіляються як правило за експоненціальним законом розподілу. Тут інтенсивність відмов залишається приблизно однаковою (див. рис. 6.1).

**Період старіння або зносу** характеризується різким збільшенням інтенсивності **зносних** відмов, що розподіляються за нормальним законом розподілу (законом Гаусса). На третьому періоді, як видно з рис. 6.1, інтенсивність відмов постійно зростає.

Виходячи з вище викладеного розглянемо детальніше найчастіше використовувані для розрахунку надійності ТЗ закони розподілу, що характеризують безперервні випадкові величини.

## 6.2. Розподіл Вейбулла

Розподіл Вейбулла (distributing of Veybulla) – двопараметричний закон розподілу випадкового напрацювання до відмови з параметрами:  $\lambda_0$ , що визначає масштаб, і  $k$ , що визначає асиметрію.

Показники надійності при такому законі розподілу будуть визначатися так:

$$P(t) = e^{-\lambda_0 t^k}, \quad (6.1)$$

$$a(t) = -P'(t) = \lambda_0 k t^{k-1} e^{-\lambda_0 t^k}, \quad (6.2)$$

$$\lambda(t) = \frac{a(t)}{P(t)} = \lambda_0 k t^{k-1}, \quad (6.3)$$

$$T_{cp} = \int_0^{\infty} P(t) dt = \frac{\Gamma(\frac{1}{k}+1)}{\lambda_0^{1/k}}, \quad (6.4)$$

де  $\Gamma\left(\frac{1}{k} + 1\right)$  – гамма-функція.

Гамма-функція визначається за виразом

$$\Gamma(X) = \int_0^{\infty} t^{X-1} e^{-t} dt, \quad (6.5)$$

Характеристика зміни ймовірності безвідмовної роботи, що залежить від часу напрацювання при перерозподілі відмов за законом Вейбулла, подана на рис. 6.2.

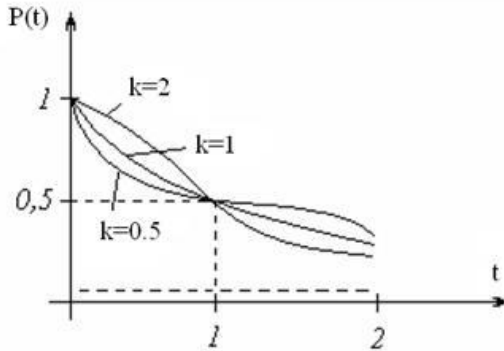


Рисунок 6.2 – характеристика зміни ймовірності безвідмовної роботи для розподілу Вейбулла при  $k=2$ ,  $k=1$  та  $k=0.5$

Дисперсія часу безвідмовної роботи для розподілу Вейбулла описується виразом

$$\sigma_T^2 = \lambda_0^{-2/k} \left[ \Gamma\left(\frac{2}{k} + 1\right) - \Gamma^2\left(\frac{1}{k} + 1\right) \right], \quad (6.6)$$

Отже, інтенсивність відмов  $\lambda(t)$ , що розподілені за законом Вейбулла, при  $k < 1$  – монотонно зменшується, при  $k > 1$  – монотонно збільшується і при  $k = 1$  –  $\lambda = const$  (рис. 6.3).

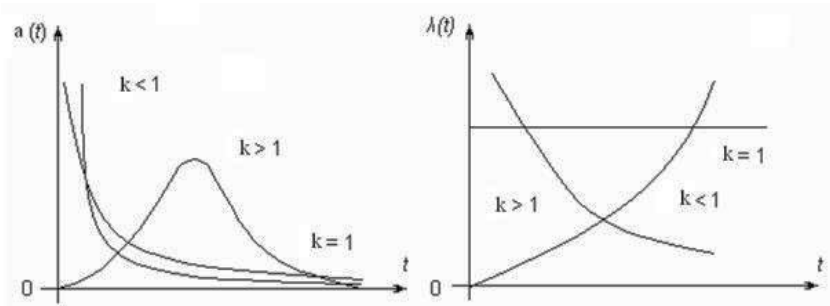


Рисунок 6.3 – характеристики зміни частоти та інтенсивності відмов, що розподілені за законом Вейбулла.

### 6.3 Експоненційний розподіл

Експоненційний закон розподілу – однопараметричний закон з постійною інтенсивністю відмов ( $\lambda_0 = const$ ). Він є частковим випадком розподілу Вейбулла при  $k = 1$ .

Ймовірність безвідмовної роботи, частота відмов і середнє напруцювання до відмови при експоненційному розподілі визначаються за формулами:

$$P(t) = e^{-\lambda_0 t}, \quad (6.7)$$

$$a(t) = \lambda_0 e^{-\lambda_0 t}, \quad (6.8)$$

$$T_{cp} = 1/\lambda_0, \quad (6.9)$$

З виразу (6.3) випливає, що при  $k=1$  інтенсивність відмов  $\lambda(t) \equiv \lambda_0$ . Саме тому експоненційний закон визначається тільки одним параметром  $\lambda_0$ , що являє собою постійну інтенсивність відмов.

Замінюючи у виразі (6.7)  $\lambda_0$  на  $1/T_{cp}$ , отримаємо

$$P(t) = e^{\frac{-t}{T_{cp}}}, \quad (6.10)$$

Ймовірність безвідмовної роботи на інтервалі часу  $t = T_{cp}$  при експоненційному розподілі дорівнює

$$P(T_{cp}) = e^{-1} = 0,368$$

Дисперсія часу безвідмовної роботи для експоненційного закону розподілу розраховується за формулою

$$\sigma_T^2 = 2 \int_0^{\infty} t e^{-\lambda_0 t} dt - \frac{1}{\lambda_0^2} = \frac{1}{\lambda_0^2} = T_{cp}^2, \quad (6.11)$$

Графік зміни ймовірності безвідмовної роботи від часу при експоненційному розподілі відмов зображено на рис. 6.4.

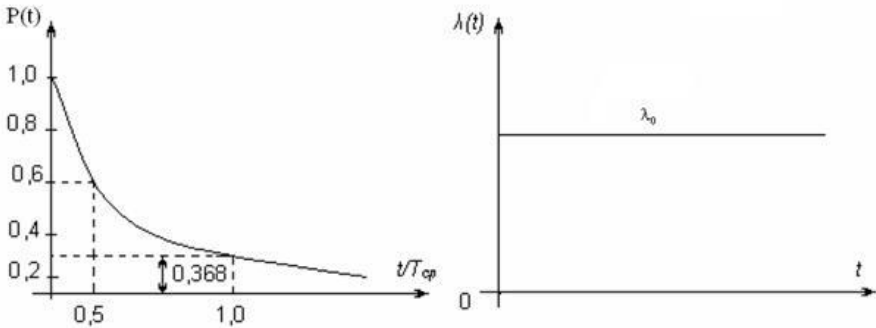


Рисунок 6.4 – Експоненційний розподіл ймовірності безвідмовної роботи та інтенсивності відмови

Знайдемо умовну ймовірність того, що для експоненційної моделі ТЗ пропрацює безвідмовно на інтервалі часу  $t$ , після того як він безвідмовно пропрацював на інтервалі  $T$ . В цьому випадку маємо

$$P\left(\frac{t}{T}\right) = \frac{P(t+T)}{P(T)} = \frac{e^{-\lambda_0(t+T)}}{e^{-\lambda_0 T}} = e^{-\lambda_0 t}, \quad (6.12)$$

Звідси випливає важливий висновок: для експоненційного закону розподілу ймовірності безвідмовної роботи розподіл часу

безвідмовної роботи не залежить від того, скільки часу ТЗ пропрацював до початку відліку від моменту першого ввімкнення. Інші закони розподілу такої властивості не мають, оскільки в інших розподілах  $\lambda_0 \neq const$ , а залежить від часу.

Модель експоненційного розподілу широко використовується для апіорного аналізу надійності. При апіорному аналізі надійності необхідно проводити перевірку відповідності експоненційної моделі результатам випробувань.

#### 6.4 Розподіл Релея

При розподілі Релея ймовірність безвідмовної роботи на інтервалі  $(0; t)$  дорівнює

$$P(t) = e^{\frac{-t}{2r^2}}, \quad (6.13)$$

де  $r$  – параметр розподілу Релея, який одночасно є модою цього розподілу (рис. 6.5, а).

Мода неперервного розподілу – це точка максимуму щільності розподілу ймовірності  $a(t)$ . Мода дискретного розподілу – це таке спектральне значення  $\eta_m$ , при якому попереднє і наступне спектральні значення мають ймовірність, меншу ніж  $P(\eta_m)$ .

Щільність розподілу напрацювання до відмови дорівнює (рис. 6.5, б)

$$a(t) = -P'(t) = \frac{1}{r^2} e^{\frac{-t}{2r^2}}, \quad (6.14)$$

Інтенсивність відмов дорівнює (рис. 6.5, в)

$$\lambda(t) = \frac{a(t)}{P(t)} = \frac{t}{r^2}, \quad (6.15)$$

Середній час безвідмовної роботи  $T_{cp}$  для розподілу Релея описується виразом

$$T_{cp} = \int_0^{\infty} t(a)dt = r\sqrt{\frac{\pi}{2}} = 1,253r, \quad (6.16)$$

Відповідно дисперсія часу безвідмовної роботи описується виразом

$$\sigma_T^2 = \left(2 - \frac{\pi}{2}\right)r^2 = 0,429r^2, \quad (6.17)$$

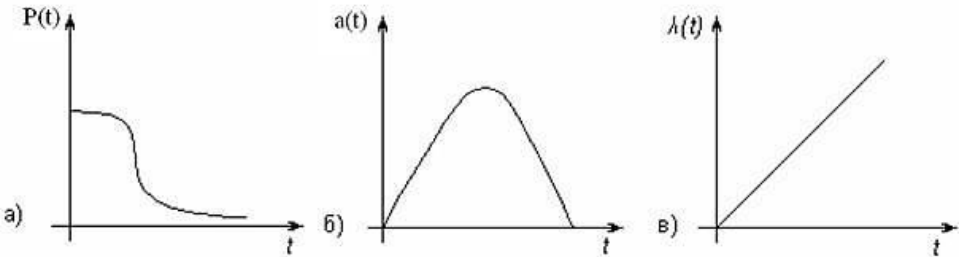


Рисунок 6.5 – Характеристики зміни ймовірності безвідмовної роботи, щільності розподілу напрацювання до відмови та інтенсивності відмови розподіленими за законом Релея.

## 6.5 Гамма-розподіл

При гамма-розподілі щільність розподілу напрацювання до відмови описується виразом

$$a(t) = \frac{\lambda_0^r}{\Gamma(r)} t^{r-1} e^{-\lambda_0 t}, \quad (6.18)$$

де  $\Gamma(r)$  – повна гамма-функція.

В теорії надійності гамма-розподіл, як правило, використовується при цілому значенні параметра  $r$ . Якщо  $r = 1$ , то гамма-розподіл перетворюється в експоненційний розподіл. Якщо  $r$  – ціле число більше 1, то гамма-розподіл є розподілом

суми незалежних випадкових величин, кожна з яких має експоненційний розподіл.

Гамма-розподіл при цілому значенні  $r$  інколи називають *розподілом Ерланга*. Для такого розподілу ймовірність безвідмовної роботи на інтервалі  $(0; t)$  описується виразом (рис. 6.6, а):

$$P(t) = e^{-\lambda_0 t} \sum_{i=0}^{r-1} \frac{(\lambda_0 t)^i}{i!}, \quad (6.19)$$

Щільність розподілу напрацювання до відмови в цьому випадку описується виразом (рис. 6.6, б)

$$a(t) = \lambda_0 \frac{(\lambda_0 t)^{r-1}}{(r-1)!} e^{-\lambda_0 t}, \quad (6.20)$$

Інтенсивність відмов визначається за формулою (6.6, в)

$$\lambda(t) = \lambda_0 \frac{(\lambda_0 t)^{r-1}}{(r-1)! \sum_{i=0}^{r-1} \frac{(\lambda_0 t)^i}{i!}}, \quad (3.21)$$

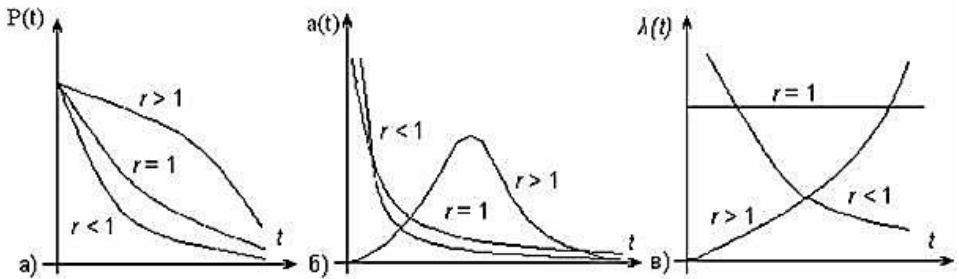


Рисунок 6.6 – Характеристики зміни гамма-розподілу: а) – функція надійності; б) – щільність розподілу напрацювання до відмови; в) – інтенсивність відмов

Середній час безвідмовної роботи і дисперсія часу безвідмовної роботи відповідно описуються виразами:



$$T_{cp} = \frac{r}{\lambda_0}, \quad (6.22)$$

$$\sigma_T^2 = \frac{r}{\lambda_0^2}, \quad (6.23)$$

При великих значеннях параметра  $r$  – гамма-розподіл наближається до нормального закону розподілу з параметрами:

$$\mu_{t_0} = rT_{cp}, \quad \sigma_{T_0}^2 = r\sigma_T^2$$

Прикладом використання гамма-розподілу є резервна система, що складається з  $r$  однакових елементів. При цьому під навантаженням знаходиться один елемент. Інші елементи почергово автоматично вмикаються в роботу після відмови працюючого елемента. При експоненційному напрацюванні до відмови елементів їх сумарне напрацювання буде підпорядковуватися гамма-розподілу.

## 6.6 Нормальний розподіл

Нормальний закон розподілу (normal law of distributing) – це двопараметричний закон з параметрами розподілу:

$T$  – математичне сподівання і

$\sigma_T$  – СКВ (час безвідмовної роботи).

Ймовірність події в інтервалі часу від  $t_1$  до  $t_2$  визначається за формулою

$$P(t_1 < t < t_2) = \frac{1}{2} [\Phi(x_2) - \Phi(x_1)] = \frac{1}{2} \left[ \Phi \left( \frac{t_2 - T_{cp}}{\sigma_T \sqrt{2}} \right) - \Phi \left( \frac{t_1 - T_{cp}}{\sigma_T \sqrt{2}} \right) \right], \quad (6.24)$$

де  $\Phi(x)$  – інтеграл ймовірності (інтеграл Лапласа) виду

$$\Phi(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt, \quad (6.25)$$

При використанні центрованої і нормованої функції Лапласа  $\Phi(z)$ , де  $z = (t - T_{cp})/\sigma_T$  ймовірність безвідмовної роботи визначається за формулою

$$P(t) = 0,5 - \Phi\left(\frac{t - T_{cp}}{\sigma_T}\right), \quad (6.26)$$

Щільність розподілу напрацювання до відмови при нормальному розподілі має вигляд (рис. 6.7)

$$a(t) = \frac{1}{\sigma_T \sqrt{2\pi}} e^{-\frac{(t - T_{cp})^2}{2\sigma_T^2}}, \quad (6.27)$$

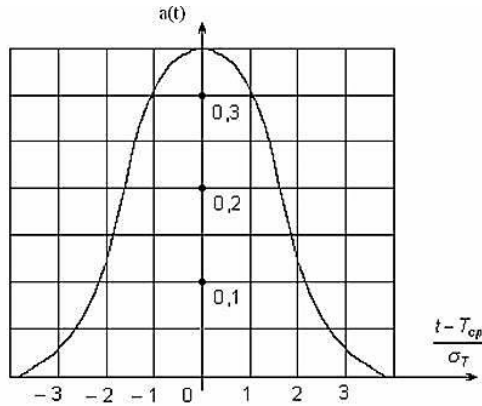


Рисунок 6.7 – Щільність розподілу напрацювання до відмови для нормального закону розподілу

Функцію надійності з врахуванням виразу (6.27) можна також записати у вигляді

$$P(t) = \int_t^{\infty} a(t) dt = \frac{1}{\sqrt{2\pi}} \int_{\frac{t - T_{cp}}{\sigma_T}}^{\infty} e^{-\frac{(t - T_{cp})^2}{2\sigma_T^2}} d\left(\frac{t - T_{cp}}{\sigma_T}\right) =$$

$$= \frac{1}{\sqrt{2\pi}} \int_{\frac{t-T_{cp}}{\sigma_T}}^{\infty} e^{-\frac{x^2}{2}} dx, \quad (6.28)$$

де  $X = \frac{t-T_{cp}}{\sigma_T}$ .

Ймовірність відмови на інтервалі  $(0; t)$  (функція ненадійності) визначається за формулою

$$Q(t) = 1 - P(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{t-T_{cp}}{\sigma_T}} e^{-\frac{x^2}{2}} dx, \quad (6.29)$$

### 6.7 Трикутний розподіл

Цей розподіл характеризує обмежену область значень випадкових величин  $(t_H; t_B)$ , де  $t_H$  і  $t_B$  – межі області можливих значень випадкових величин.

Розглянемо показники надійності трикутного розподілу:

- характеристики зміни щільності  $a(t)$  та інтенсивності відмов  $\lambda(t)$  (рис. 6.8, а);

- характеристики зміни функції надійності  $P(t)$  (рис. 6.8, б).

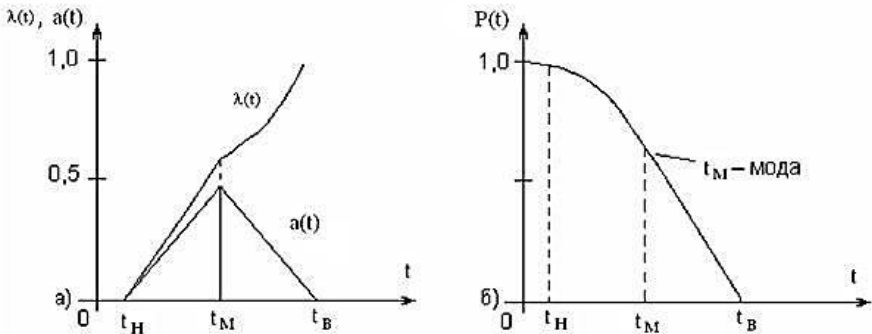


Рисунок 6.8 – Характеристики зміни показників надійності при трикутному розподілі відмов

Позначимо значення щільності розподілу в точці моди через  $a(t_M)=h$ , тоді  $h(t_B - t_M)/2=1$ .

В цьому випадку щільність розподілу можна записати у вигляді таких формул:

$$a(t) = \begin{cases} \frac{2(t-t_H)}{(t_B-t_H)(t_M-t_H)} & \text{при } t_H \leq t \leq t_M \\ \frac{2(t_B-t)}{(t_B-t_H)(t_B-t_M)} & \text{при } t_M \leq t \leq B \end{cases}, \quad (6.30)$$

Функція надійності  $P(t)$  буде описуватись такою системою рівнянь:

$$P(t) = \begin{cases} 1 - \frac{(t-t_H)^2}{(t_B-t_H)(t_M-t_H)} & \text{при } t_H \leq t \leq t_M \\ \frac{(t_B-t)^2}{(t_B-t_H)(t_B-t_M)} & \text{при } t_M \leq t \leq B \end{cases}, \quad (6.31)$$

Інтенсивність відмов  $\lambda(t)$  при цьому буде дорівнювати:

$$\lambda(t) = \begin{cases} \frac{2(t-t_H)}{(t_B-t_H)(t_M-t_H)-(t-t_H)^2} & \text{при } t_H \leq t \leq t_M \\ \frac{2}{t_B-1} & \text{при } t_M \leq t \leq B \end{cases}, \quad (6.32)$$

Середній час напрацювання до відмови  $T_{cp}$  визначається за формулою

$$T_{cp} = \int_{t_H}^{t_B} ta(t) dt = \frac{1}{3}(t_H + t_M + t_B), \quad (3.33)$$

## 6.8 Закони розподілу дискретних випадкових величин

Наведені вище розподіли характеризують неперервні випадкові величини, наприклад, час безвідмовної роботи або час відновлення.

Але в ряді випадків при розрахунку надійності ТЗ виникає необхідність оцінки дискретних випадкових величин, наприклад, кількості відмов протягом заданого інтервалу часу.

Тому розглянемо найбільш часто використовувані при розрахунках надійності розподіли дискретних випадкових величин.

### 6.8.1 Біноміальний розподіл

Для такого розподілу можливі значення випадкової величини  $0, 1, 2, 3, \dots, n$ .

Ймовірність появи  $m$  сприятливих подій із загальної кількості  $n$  подій дорівнює

$$P_n(m) = C_n^m P^m Q^{n-m}, \quad (6.34)$$

Математичне сподівання і дисперсія відповідно будуть дорівнювати:

$$\mu(m) = nP, \quad (6.35)$$

$$\sigma_T^2(m) = nPQ, \quad (6.36)$$

де  $P$  – ймовірність здійснення події при одноразовому випробуванні;

$$Q = 1 - P.$$

### 6.8.2 Розподіл Пуассона

Можливі значення випадкової величини для такого розподілу такі:  $0, 1, 2, \dots, n$ . Ймовірність появи  $m$  подій дорівнює

$$P_m = \frac{\lambda^m}{m!} e^{-\lambda}, \quad (6.37)$$

Математичне сподівання і дисперсія відповідно будуть дорівнювати:

$$\mu(m) = \lambda, \quad (6.38)$$

$$\sigma_T^2(m) = \lambda, \quad (6.39)$$

де  $\lambda$  – параметр розподілення.

### 6.8.3 Геометричний розподіл

Можливі значення випадкової величини такі:  $0, 1, 2, \dots, n$ . Ймовірність появи  $m$  подій дорівнює

$$P_m = PQ^{m-1}, \quad (6.40)$$

Математичне сподівання і дисперсія відповідно будуть дорівнювати:

$$\mu(m) = 1/P, \quad (6.41)$$

$$\sigma_T^2(m) = \frac{Q}{P^2}, \quad (6.42)$$

де  $P$  – ймовірність появи події при одноразовому випробуванні;

$$Q = 1 - P.$$

## 7. Надійність програмного забезпечення. Визначальні фактори.

### 7.1. Основні поняття

**Надійність програмного забезпечення** - здатність програмного продукту безвідмовно виконувати певні функції при заданих умовах протягом заданого періоду часу з досить великою ймовірністю.

Ступінь надійності характеризується ймовірністю роботи програмного продукту без відмови протягом певного періоду часу.

Існує 4 основні складові функціональної надійності програмних систем:

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

### 7.2. Модель аналізу надійності програмних засобів

Модель аналізу надійності програмних засобів наведена на рисунку 7.1.



Рисунок 7.1.



### 7.3. Фактори, що впливають на надійність ПЗ

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

- 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;
- взаємодія з людиною (розробником та користувачем) (див. наприклад метрику Холстед);
- організація ПЗ (проекування, постановка задачі та способи їх досягнення і реалізації) і якість його розробки. Цей фактор вносить найбільший вклад в надійність;
- тестування.

### 7.4. Помилки ПЗ

У боротьбі зі складністю ПЗ використовуються дві концепції:

**Ієрархічна структура.** Ієрархія дозволяє розбити систему за рівнями розуміння (абстракції, управління). Концепція рівнів дозволяє аналізувати систему, приховуючи несуттєві для даного рівня деталі реалізації інших рівнів. Ієрархія дозволяє розуміти, проектувати і описувати складні системи.

**Незалежність.** Відповідно до цієї концепції, для мінімізації складності, необхідно максимально посилити незалежність елементів системи. Це означає таку декомпозицію системи, щоб її високочастотна динаміка була укладена в окремих компонентах, а міжкомпонентні взаємодії (зв'язку) описували тільки низькочастотну динаміку системи.

**Методи виявлення помилок**, які базуються на введенні в ПЗ системи різних видів надмірності:

**Тимчасова надмірність.** Використання частини продуктивності ЕОМ для контролю виконання та відновлення працездатності ПЗ після збою.

**Інформаційна надмірність.** Дублювання частини даних інформаційної системи для забезпечення надійності та контролю достовірності даних.

**Програмна надмірність** включає в себе:

- взаємна недовіра - компоненти системи проєктуються, виходячи з припущення, що інші компоненти і вихідні дані містять помилки, і повинні намагатися їх виявити;
- негайне виявлення і реєстрацію помилок;
- виконання однакових функцій різними модулями системи і зіставлення результатів обробки;
- контроль і відновлення даних з використанням інших видів надмірності.

**Завдання забезпечення ПЗ стійкості до помилок** спрямовані на застосування методів мінімізації збитку, викликаного появою помилок, і включають в себе:

- обробку збоїв апаратури;
- повторне виконання операцій;
- динамічна зміна конфігурації;
- скорочене обслуговування у випадку відмови окремих функцій системи;
- копіювання і відновлення даних;
- ізоляцію помилок.

Існують чотири групи принципів забезпечення надійності:

- попередження помилок;
- виявлення помилок;

- виправлення помилок;
- забезпечення стійкості до помилок.

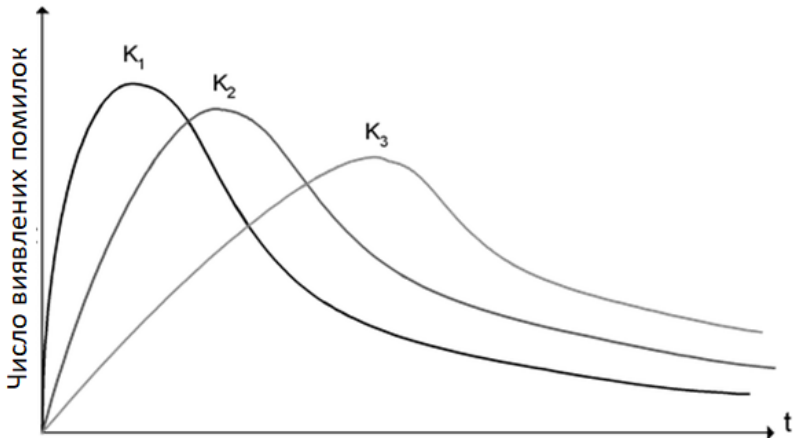
Дії, спрямовані на мінімізацію помилок і збоїв:

- 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.

Відповідно до ГОСТ 19.004-80 розрізняють наступні види робіт, спрямовані на усунення помилок в ПЗ: перевірка, налагодження і випробування програми.

Один з основних міжнародних стандартів програмної інженерії ISO/IEC 12207: 2008 «System and software engineering – Software life cycle processes» стандарт ISO, що описує процеси життєвого циклу програмного забезпечення. Будучи стандартом високого рівня, він не задає деталі того, як треба виконувати дії або завдання, складові процеси. Він також не ставить вимог до формату і змісту документів, що випускаються на різних процесах. Тому для опису виді робіт, спрямовані на усунення помилок в ПЗ, будемо користуватись ГОСТ 19.004-80.

Чим інтенсивніше використання ПЗ, тим швидше виявляються в ньому помилки. На рисунку 7.1 наведена залежність числа виявлених помилок від числа користувачів, які використовують ПЗ:



де  $K$  - число користувачів,  $K_1 > K_2 > K_3$ .

Рис. 7.1. Інтенсивність виявлення помилок від інтенсивності використання

Процентні частоти появи помилок в ПЗ за типами помилок представлені в таблиці 7.1.

Таблиця 7.1 - Процентні частоти 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

Тип помилки	Частота появи, %
Помилкові арифметичні операції	9
Брак часу для вирішення	4
Помилка обробки переривань	4
Помилка у вихідних даних	3
Неточний запис	8

Як видно з таблиці 7.1 основна кількість помилок робиться через невірну специфікації або ТЗ. Ці помилки, в свою чергу, можуть бути розділені на наступні категорії:

Таблиця 7.2 - 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
Неправильне опис або вимога до апаратури	2
Вихідні дані для розробки неповні, неточні або помилкові	52

Причина помилки	Частота появи, %
Двозначність вимог	13

З цих таблиць слідує, на що потрібно звертати особливу увагу при проведенні валідації та верифікації ПЗ (верифікація відповідає на питання, чи правильно і якісно створено програму, а валідація (або атестація) - на питання чи правильно працює програма).

### 7.5. Засоби та способи підвищення надійності ПЗ

На підставі методів виявлення помилок були розроблені наступні *засоби* підвищення надійності ПЗ.

#### **Засоби, що використовують тимчасову надмірність:**

- авторизація доступу користувачів до системи,
- аналіз доступних користувачеві ресурсів,
- виділення ресурсів згідно ролям і рівнями підготовки користувачів,
- розмежування прав доступу користувачів до окремих завдань, функцій управління, записам і полях баз даних.

#### **Засоби забезпечення надійності, використовують інформаційну надмірність:**

- посилення цілісності баз даних забезпечується за рахунок системи внутрішніх унікальних ключів для всіх інформаційних записів системи,
- відкрита система кодування, що дозволяє користувачеві в будь-який момент змінювати коди будь-яких об'єктів класифікації, забезпечує стикування системи класифікації АІС діловодство з ПЗ інших розробників, механізми перевірки

значень контрольних сум записів системи, забезпечують виявлення всіх несанкціонованих модифікацій (помилки, збоїв) інформації.

*Способи забезпечення і підвищення надійності ПЗ:*

- удосконалення технології програмування (наприклад, формальний опис етапів програмування за допомогою мови UML);
- вибір алгоритмів, не чутливих до різного роду порушень обчислювального процесу (використання алгоритмічної надмірності);
- резервування програм - *N-версійність* програмування;
- верифікація і валідація програм з подальшою корекцією.

## **7.6. Проблеми дослідження надійності ПЗ**

До основних проблем досліджень надійності ПЗ відносяться:

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

## **7.7. Тестування ПЗ**

Важливим етапом життєвого циклу ПЗ, що визначає якість і надійність системи, є **тестування**. *Тестування* - процес

виконання програм з наміром знайти помилки і включає в себе наступні етапи:

- автономне тестування;
- тестування сполучень;
- тестування функцій;
- комплексне тестування;
- тестування повноти і коректності документації;
- тестування конфігурацій.

Надійність ПЗ підвищується також за допомогою застосування різних *методів тестування*. Повне тестування ПЗ неможливо. Зазвичай застосовують такі види тестування:

- тестування гілок;
- математичний доказ правильності алгоритму розв'язання задачі (в деяких роботах саме в цьому сенсі вживається слово верифікація);
- символічне тестування (або з 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;
- динамічне тестування (за допомогою динамічно генерованих вхідних даних), що зручно при швидкому тестуванні у всьому широкому діапазоні вхідних параметрів;
- тестування шляхів виконання програми;
- функціональне тестування;
- перевірки за часом виконання програми;
- перевірка по використанню ресурсів та стресове тестування.



## 7.8. Класифікація показників якості ПЗ

Класифікація показників якості ПЗ наступна:

- За кількістю характеризуються властивостей розрізняють одиничні і комплексні показники. Одиничні показники якості характеризують одну із властивостей ПЗ, комплексний - декілька. Комплексні показники можуть бути груповими, узагальненими або інтегральними.

- Залежно від місця застосування у процедурі оцінки рівня якості ПЗ розрізняють базові та відносні показники. **Базовим значенням показника якості продукції** називають значення показника, прийняте за основу при порівняльній оцінці якості продукції. **Відносне значення показника якості продукції** являє собою відношення фактичного значення показника якості оцінюваної продукції до базового значення цього показника.

- За стадії визначення значень показників якості розрізняють *прогнозовані, проектні, виробничі і експлуатаційні показники*. **Прогнозовані показники** оперують на стадіях виконання науково-дослідних робіт та складання ТЗ на розробку ПЗ, тобто на тих стадіях, коли немає ще жодного детального проекту ПЗ, ні , тим більше, самого ПЗ. Значення прогнозованих показників в основному визначають на основі інтуїції і досвіду аналогічних розробок, тому ці показники носять суб'єктивний характер.

**Значення проектних показників** визначають на основі аналізу проектів ПЗ (ескізного, технічного, робочого), а також шляхом випробування дослідного зразка ПЗ. Ці показники носять більш об'єктивний характер. Ступінь їх достовірності залежить від ефективності використовуваних інструментальних засобів аналізу і випробування.

**Виробничі показники** мало відрізняються від проектних, особливо якщо виготовлення ПЗ зводиться до простого

копіювання. Якщо ж копіюванню передують операції складання або генерації ПЗ, то виробничі показники якості таких ПЗ можуть істотно відрізнятись від проектних.

**Значення експлуатаційних показників** визначають за результатами промислової експлуатації ПЗ. При дотриманні певних правил збору та обробки даних про якість ПЗ в процесі експлуатації експлуатаційні показники дають найбільш об'єктивну і достовірну оцінку. Тільки за цими показниками можна зробити дійсну оцінку науково-технічного рівня і якості ПЗ.

Близько 50% приватних показників можна визначити автоматично за допомогою ЕОМ, 25% - за допомогою компаратора. Таким чином, оцінка близько 75% показників може бути формалізована. Оцінка 20% показників може бути зроблена тільки кваліфікованим фахівцем. Більшість показників встановлюють шляхом статичного аналізу програм і лише близько 5% - у процесі динамічних випробувань.

## 7.9. Основні показники надійності ПЗ

1. *Ймовірність безвідмовної роботи  $P(t_3)$*  - це ймовірність того, що в межах заданого напрацювання відмова системи не виникає.

2. *Ймовірність відмови* - ймовірність того, що в межах заданого напрацювання відмова системи виникає.

Це показник, зворотний попередньому.

$$Q(t_3) = 1 - P(t_3), \quad (7.1)$$

де  $t_3$  - задане напрацювання, год;

$Q(t_3)$  - ймовірність відмови.

3. **Інтенсивність відмов системи** - це умовна щільність ймовірності виникнення відмови ПЗ в певний момент часу за умови, що до цього часу відмова не виникла.

$$Q(t) = \frac{f(t)}{P(t)}, \quad (7.2)$$

де  $f(t)$  - щільність імовірності відмови в момент часу  $t$ .

$$f(t) = \frac{d}{dt} Q(t) = \frac{d}{dt} (1 - P(t)) = -\frac{d}{dt} P(t), \quad (7.3)$$

Існує наступний зв'язок між інтенсивністю відмов системи і ймовірністю безвідмовної роботи

$$P(t) = e^{-\int_0^t Q(t)dt}, \quad (7.4)$$

В окремому випадку, при

$$P(t) = \exp(-Q(t)), \quad (7.5)$$

$$Q(t) = \text{const}$$

Якщо в процесі тестування фіксується число відмов за певний часовий інтервал, то інтенсивність відмов системи є число відмов в одиницю часу.

4. **Середнє напрацювання на відмову  $T_i$**  - математичне очікування часу роботи ПЗ до чергової відмови:

$$T_i = \int_0^t t \cdot f(t)dt, \quad (7.6)$$

Інакше середню напрацювання на відмову  $T_i$  можна представити:  $i = 1$ ;

$$T_i = \frac{t_1 + t_2 + t_3 \dots t_n}{n} = \left| \frac{i}{n} \right| \cdot \sum_{i=1}^n t_i, \quad (7.7)$$

де  $t$  - час роботи ПЗ між відмовами, с.

$n$  - кількість відмов.

5. **Середній час відновлення  $T$**  - математичне очікування часу відновлення -  $t$ ; часу, витраченого на виявлення і локалізацію відмови -  $t_1$ ; часу усунення відмови -  $t_2$ ; часу пропускнуї перевірки працездатності -  $t_3$ :

$$t = t_1 + t_2 + t_3,$$

де  $t_i$  - час відновлення після  $i$ -ї відмови.

$$T = i / nt,$$

$$i = 1$$

де  $n$  - кількість відмов.

Для цього показника термін "час" означає час, витрачений фахівцем з тестування на перелічені види робіт.

6. **Коефіцієнт готовності  $K$**  - ймовірність того, що ПЗ очікується в працездатному стані в довільний момент часу його використання за призначенням:

$$K = T / (T + T).$$

Необхідно прагнути підвищувати рівень надійності ПЗ, але досягнення 100 % надійності лежить за межами можливого. Кількісні показники надійності можуть використовуватися для оцінки досягнутого рівня технології програмування, для вибору методу проектування майбутнього програмного засобу.

Основним засобом визначення кількісних показників надійності є моделі надійності, під якими розуміють математичну модель, побудовану для оцінки залежності надійності від задалегідь відомих або оцінених в ході створення програмних засобів параметрів.

7. Всі наведені показники надійності ПЗ характеризують наявність помилок програми (виробничих дефектів), але жоден з них не характеризує характер цих помилок і можливі їх наслідки.

Тому запропоновано ввести новий показник надійності ПЗ - *середня тяжкість помилок (СТП)*:

$$B = \frac{1}{Q} \sum_i M(b_i n_i z_i), \quad (7.8)$$

де  $Q$  - ймовірність збою ПЗ;

$\Sigma$  - оператор підсумовування по змінній  $i$ ;

$b_i$  - функція залежності тяжкості наслідків помилки, що виникла при  $i$ -му наборі вхідних даних, до максимально важких наслідків;

$p_i$  - імовірність введення  $i$ -го набору вхідних даних при експлуатації ПЗ;

$z_i$  - дихотомическая змінна, що дорівнює  $1$ , якщо при  $i$ -му наборі вхідних даних був зафіксований збій, і  $0$  у протилежному випадку;

$M$  - загальне число наборів вхідних даних.

Значення показника надійності СТП лежить на інтервалі  $[0; 1]$ . Чим ближче значення СТП до одиниці, тим важче наслідки помилок ПЗ, і тим менш надійна програма. Близькість СТП до нуля показує незначність наслідків помилок програми.

Введення нового показника надійності ПЗ дозволило розрізнити по надійності програмні продукти, ймовірності збою яких мають один і той же порядок. До того ж, говорячи про надійність ПЗ, користувач бажає отримати не стільки безпомилкове ПЗ, скільки безпечне. А саме безпека ПЗ характеризує СТП. Значення цього показника суб'єктивно і може бути різним для одного і того ж програмного продукту в залежності від області його застосування. Це пояснюється тим, що при використанні конкретного ПЗ, наприклад, для виконання студентських розрахунків та для виконання конструкторських розрахунків в космічній промисловості наслідки помилок

програми - непорівнянні. У ряді випадків, якщо до ПЗ висуваються жорсткі вимоги, краще оцінювати максимальну тяжкість помилок ПЗ.

Таким чином, оцінюючи ймовірність збою ПЗ і СТП, отримуємо багатосторонню оцінку надійності ПЗ.

### **7.10. Загальна характеристика та класифікація моделей надійності програмного забезпечення**

Дослідження в галузі надійності ПЗ беруть початок з 1970-х років. Незважаючи на зусилля провідних світових розробників ПЗ, завдання зниження кількості помилок у програмних системах не отримало ефективного практичного вирішення [1]. Об'єктивно це зумовлено надзвичайно високою структурною складністю програмних систем, динамічністю версій і технологій. Одним із шляхів підвищення рівня надійності ПЗ є використання на етапах тестування і випробувань ПЗ моделей, що дають змогу отримати гарантовані оцінки показників безпеки ПЗ і ефективності технології його розроблення. Більшість таких моделей запозичене з теорії надійності технічних систем, тому в літературі їх часто називають моделями надійності ПЗ [2, 3].

Від 70-х років минулого століття було розроблено значну кількість моделей надійності ПЗ [4–7]. Як основні критерії класифікації таких моделей вибрано два доволі прості предмети дослідження:

1) дослідження кількості відмов за певний період часу (вимір часу “в режимі настінного годинника” або вимір часу відносно виконання процесів пристроями комп'ютера);

2) дослідження часових проміжків між помилками.

Вважали, що моделі, класифіковані цим способом, взаємно не перетинаються і можуть містити часткові випадки в кожному дослідженні [8]. Однією з перших складних моделей надійності

ПЗ вважають модель Муси і Окумото [9]. Для побудови цієї моделі використано набір атрибутів, зокрема:

- часовий проміжок;
- загальну кількість відмов, які можливо виявити за нескінченний або скінченний проміжок часу;
- розподіл кількості відмов, які відбулися за час  $t$  (розподіл Пуассона або біноміальний розподіл);
- клас відмови, або функціональна форма активності відмов за певний час (застосовується тільки для випробувань із скінченними часовими проміжками);
- тип відмови, або вигляд функції інтенсивності відмов протягом нескінченного проміжку часу (застосовується тільки для випробувань із нескінченними часовими проміжками) [9].

З розвитком технологій розроблення і практичного застосування ПЗ схема класифікації моделей надійності ПЗ значно розширилася. Не тільки розроблено нові критерії класифікації, але й ускладнено саму структуру класифікації внаслідок об'єднання і перетину декількох критеріїв у різних моделях [10].

До найбільш вживаних методик і факторів надійності ПЗ належать: стадія життєвого циклу розроблення ПЗ (класифікація моделей залежить від етапу, на якому розраховується надійність ПЗ), можливість раннього прогнозування помилок, орієнтованість на інформацію або архітектуру (класифікацію роблять на підставі перевірки правильності вхідних / вихідних даних, або перевірки функціонального наповнення ПЗ), зростання надійності ПЗ в процесі виявлення та виправлення помилок тощо [11]. Класифікація за такими факторами [12] є найповнішою та дає можливість представити не тільки самі моделі, але і взаємозв'язок між ними (рис. 7.2).

Кількість моделей надійності ПЗ сьогодні перевищує сотню і продовжує зростати [6]. Залежно від визначення поняття надійності ПЗ і глибини дослідження обирають різні критерії і характеристики моделей надійності. Тому наведена на рис. 7.2 схема класифікації моделей надійності ПЗ, як і розглянуті критерії, не є єдиною [13].

Схема, наведена на рис. 7.2, поділяє всі моделі надійності ПЗ на аналітичні та емпіричні.

Емпіричні моделі надійності, своєю чергою, поділяються на моделі складності і моделі, що визначають час, необхідний на “доведення” програми. Аналітичні моделі надійності ПЗ поділяють на динамічні (дискретні і неперервні) і статичні (за областю помилок, за областю даних) [6].

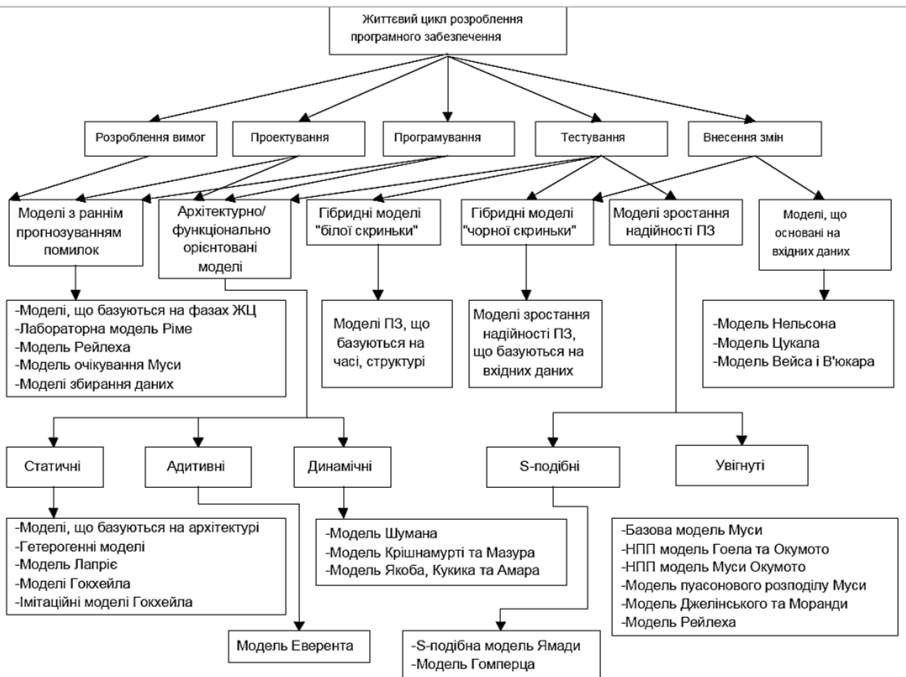


Рис. 7.2. Класифікація моделей надійності ПЗ та їх взаємозв'язок життєвим циклом ПЗ



Життєвий цикл ПЗ загалом зазвичай поділяють на такі етапи [14]: специфікація вимог, проектування, кодування, тестування, експлуатація і супровід. Етап проектування може включати попереднє проектування і деталізоване проектування. Етап тестування може містити модульне, інтеграційне та регресійне тестування, тестування в умовах експлуатації. Етап супроводу може містити один або два цикли, кожен з яких має всі етапи стадії розробки. Згідно з ДСТУ 3918-1999 (ISO/IEC 12207:1995) [14] основними процесами життєвого циклу ПЗ є процеси: замовлення, постачання, розроблення, експлуатації та супроводу. Своєю чергою, процес розроблення ПЗ містить такі дії: реалізацію процесу, аналіз системних вимог, проектування архітектури системи, аналіз вимог до ПЗ, проектування архітектури ПЗ, розроблення детального проекту ПЗ, кодування і тестування ПЗ, інтеграцію ПЗ, кваліфікаційне тестування ПЗ, системну інтеграцію, кваліфікаційне тестування системи, встановлення ПЗ та забезпечення приймання ПЗ [14].

На ранніх стадіях життєвого циклу ПЗ потрібна модель прогнозування надійності, оскільки даних про відмови немає. Моделі такого типу призначені для передбачення кількості помилок у програмі перед тестуванням, і в деяких літературних джерелах [15] належать до детерміністичних (статичних) моделей надійності ПЗ. На етапі тестування показники надійності ПЗ покращуються завдяки відлагодженню програми. Модель зростання надійності на цьому етапі потрібна для оцінювання поточного рівня надійності, часу і ресурсів, потрібних для досягнення заданого рівня надійності ПЗ. Впродовж цього етапу оцінка надійності ґрунтується на аналізі цих відмов. Моделі такого типу належать [15] до імовірнісних (динамічних) моделей надійності ПЗ. Після введення програми в експлуатацію при визначенні її надійності необхідно враховувати додавання нових

модулів, усунення старих модулів, усунення виявлених помилок, поєднання нового коду з попередньо написаним кодом, зміну середовища користувача, зміну апаратного забезпечення тощо.

На цьому етапі використовують еволюційні моделі надійності [16].

За моделями, в основу яких покладено підрахунок відмов (динамічні моделі), припускають, що концептуально в програмі наявна скінченна кількість помилок. Враховуючи, що кількість помилок є цілим числом, за динамічними моделями обчислюють кількість початкових помилок на етапі відлагодження програми і кількість помилок, що залишились під час чи в кінці етапу відлагодження. Моделі підрахунку відмов використовують інтенсивність відмов як основну характеристику появи відмови. Залежно від типу моделі припускають, що інтенсивність відмов кожної помилки є або сталою функцією часу відлагодження, або випадковою змінною із заданим законом розподілу [15–17]. Як тільки інтенсивність відмов, пов'язана з помилками певного типу, визначено, інтенсивність відмов програми загалом обчислюють як добуток кількості помилок, що залишились в програмі, на інтенсивність відмов, породжених помилкою кожного типу [16]. Під час етапу відлагодження кількість помилок, що залишились, змінюється. Одним із способів моделювання такого процесу відмов є представлення кількості помилок, що залишились, як стохастичного процесу [16, 17].

Оцінка надійності ПЗ є все важливішою під час розроблення та тестування нових програмних продуктів. Перед тим, як нове ПЗ випустити в користування, його ретельно перевіряють на наявність помилок, які могли з'явитися в процесі розроблення ПЗ. Незважаючи на те, що виявлені помилки негайно видаляються, нові помилки можуть бути введені під час процесу відлагодження.

Програмне забезпечення, яке містить помилки і випускається на ринок, несе високі ризики, пов'язані з відмовами. Відлагодження і тестування, з іншого боку, зменшує кількість помилок, але збільшує витрати на розроблення. Отже, існує необхідність визначити оптимальний час для зупинки тестування ПЗ. Під час тестування системи надійність є важливим критерієм при ухваленні рішення, коли саме потрібно випустити ПЗ. Декілька інших критеріїв, такі як кількість помилок, що залишилися, частота відмов, вимоги до надійності або загальна вартість системи можуть бути використані для визначення оптимального часу тестування ПЗ [18].

У дослідженнях в області інженерії надійності ПЗ широко використовують ряд моделей зростання надійності ПЗ на основі неоднорідного процесу Пуассона, які запропоновано для оцінювання надійності ПЗ. Насправді моделі надійності ПЗ на основі неоднорідного процесу Пуассона стали доволі успішними практичними інструментами аналізу надійності ПЗ. У цих моделях розглядають процес відлагодження ПЗ, який можна охарактеризувати цільовою функцією.

Надійність ПЗ можна оцінити як визначену цільову функцію. Параметри моделі, як правило, оцінюють з використанням методу максимальної правдоподібності або методу регресії. Різні моделі цього класу було побудовано на основі різних припущень.

Модель Гоеля–Окумото [18] також відома як експоненційна модель зростання надійності ПЗ, що ґрунтується на неоднорідному процесі Пуассона. Ця модель ґрунтується на таких припущеннях:

- 1) усі відмови в програмі є незалежними;
- 2) кількість відмов, виявлених у будь-який час, пропорційна поточній кількості помилок у програмі; це означає, що ймовірність виявлення кожної помилки є сталою величиною;

3) виявлені відмови видаляються до наступного тестування;

4) кожного разу, як відбувається відмова ПЗ, помилка, яка спричинила цю відмову, негайно виправляється, і жодних нових помилок до системи не вноситься [18].

Поведінка відмов ПЗ з часом у цій моделі описується таким диференціальним рівнянням:

$$\frac{\partial m(t)}{\partial t} = b[a - m(t)], \quad (7.9)$$

де  $m(t)$  – кількість відмов протягом часу  $t$ ;

$a$  – очікувана загальна кількість помилок, що існує в ПЗ перед тестуванням;

$b$  – коефіцієнт виявлення відмов.

Муса у 1985 році [18], розглядаючи взаємозв'язки між часом виконання та календарним часом, запропонував модель, подібну до моделі Гоеля–Окумото. Нехай  $m(t)$  – це кількість відмов, виявлених в результаті запуску тестів за певний час. Муса [18] запропонував таке диференціальне рівняння:

$$\frac{\partial m(t)}{\partial t} = \frac{c}{nT} [a - m(t)], \quad (7.10)$$

де  $a$  – кількість помилок в програмі;

$c$  – фактор компресії тестування;

$T$  – час до відмови на початку тестування;

$n$  – загальна кількість відмов, можливих за час існування програми;

$t$  – час виконання або процесорний час, який затрачено на виконання тестів за час дослідження [18].

Охба [18] запропонував гіперекспоненційну модель зростання надійності ПЗ. Ця модель базується на припущенні про те, що ПЗ містить певну кількість кластерів модулів, кожен з яких містить різну початкову кількість помилок, і кожен з них має свій

коефіцієнт відмов. Прикладами модулів можуть бути нові модулі та повторно використані модулі, прості та складні модулі, модулі, які взаємодіють з апаратним забезпеченням та ті, що не взаємодіють з ним тощо. Варто зауважити, що гіперекспоненційний розподіл є сумою експоненційних розподілів [18].

Подібне розширення експоненційної моделі зростання надійності ПЗ запропонували Ямада та Осакі [18], які поділяють ПЗ на  $k$  модулів. В цій моделі вважають, що інтенсивність відмов у межах різних модулів є різною, а інтенсивність відмов у межах однакових модулів є однаковою. Очікувану кількість відмов, виявлених для кожного модуля, вважають експоненційною. В такому випадку очікувану кількість відмов усього ПЗ можна отримати так:

$$m(t) = a \sum_{i=1}^k P_i [1 - e^{-b_i t}], \quad (7.11)$$

де  $k$  – кількість модулів у програмній системі;

$b_i$  – коефіцієнт виявлення однієї відмови у  $i$ -му модулі;

$p_i$  – ймовірність відмов для  $i$ -го модуля;

$a$  – очікувана кількість відмов ПЗ, виявлена у результаті, або загальна кількість помилок, наявна в ПЗ перед тестуванням.

Ще одним типом моделей надійності ПЗ на основі неоднорідного процесу Пуассона є *S-подібні моделі*. В цих моделях крива зростання надійності є  $S$ -подібною. Це означає, що крива перетинає експоненційну криву знизу, і перетинає лише один раз. Коефіцієнт виявлення відмов, який залежить від часу, є найбільшим у той час, коли тестування почалось, після чого він спадає експоненційно. Інакше кажучи, деякі помилки є екрановані іншими помилками на початку фази тестування, і

поки ці помилки не виправляться, приховані помилки так і залишаться невиявленими [18]. Ямада також зазначив, що процес тестування програмного забезпечення часто містить процес навчання, під час якого тестувальники ознайомлюються з ПЗ, середовищами та вимогами до програмної системи [18]. S-подібні моделі у вигляді неоднорідного процесу Пуассона ґрунтуються на таких припущеннях:

- 1) коефіцієнт виявлення відмов відрізняється між відмовами;
- 2) кожного разу, як виникає відмова, помилка в програмному забезпеченні, яка спричинила відмову, негайно видаляється, і нові помилки в систему не вносяться [18].

### **7.11. Моделі надійності програмного забезпечення з урахуванням недосконалого відлагодження.**

В основу більшості вищеописаних моделей надійності ПЗ покладено припущення про ідеальне відлагодження, коли під час процесу тестування виявляється відмова ПЗ, знаходиться відповідна помилка та виправляється, але при цьому нові помилки не вносяться. Однак, це припущення є спрощенням реального процесу розроблення програмних систем, оскільки в процесі розроблення не завжди вдається видалити помилку із системи, під час видалення однієї помилки можуть бути внесені інші, відмова може бути викликана кількома взаємопов'язаними помилками тощо.

Загалом моделі недосконалою відлагодження будуються на основі таких припущень:

- 1) коефіцієнт виявлення відмов відрізняється між відмовами;
- 2) кожного разу, як виникає відмова ПЗ, помилка в ПЗ, яка спричиняє цю відмову, виправляється, і нові помилки можуть бути внесені в систему.

Концепція недосконалого відлагодження була вперше представлена в моделях надійності ПЗ Гоеля. Він у своїй роботі [17] описав це як одне з найважливіших практичних обмежень існуючих моделей. Гоель та Окумото запропонували модель недосконалого відлагодження, яка є розширенням існуючої моделі Джелінські–Моранди. В цій моделі кількість відмов у системі на час  $t$ ,  $X(t)$  розглядається як марковський процес, де ймовірність переходів між станами системи визначається ймовірністю недосконалого відлагодження. Час між переходами  $X(t)$  має експоненційний розподіл із значеннями, залежними від кількості помилок у системі. Функція відмов на інтервалі між  $(i-1)$ -ю та  $i$ -ю відмовами визначається виразом:

$$Z(t_i) = [N - p(i - 1)]\lambda, \quad (7.12)$$

де  $N$  – початкова кількість помилок в системі;

$p$  – ймовірність недосконалого відлагодження;

$\lambda$  – інтенсивність відмов.

Модель Фама–Нордмана–Жанга містить явище недосконалого відлагодження, припускаючи, що помилки можуть бути внесені під час фази відлагодження. У цій моделі припускають, що коефіцієнт внесення помилок є лінійною функцією, залежною від часу, а функція виявлення відмов є незростаючою та залежною від часу [19]. Цільова функція має вигляд:

$$m(t) = \frac{a(1-e^{-bt})\left(1-\frac{a}{b}\right)+\alpha t}{1+\beta e^{-bt}}, \quad (7.13)$$

де  $a(t)$  – коефіцієнт внесення помилок під час фази відлагодження;

$a(t)$  – очікувана кількість початкових помилок;  
 $b(t)$  – коефіцієнт виявлення помилок.

Модель Фама–Жанга [20] припускає, що коефіцієнт внесення помилок є експоненційною функцією часу тестування, а функція виявлення помилок є незростаючою. Цільова функція набуває такої форми:

$$m(t) = \frac{1}{(1 + \beta e^{-bt})} ((c + a)(1 - e^{-bt}) - \frac{ab}{b - a}(e^{-at} - e^{-bt})), \quad (7.14)$$

де  $a(t)$  – коефіцієнт внесення помилок під час фази відлагодження;

$a(t)$  – очікувана кількість початкових помилок;

$b(t)$  – коефіцієнт виявлення помилок.

У моделі недосконалого відлагодження Ямади [21] припускають, що функція кількості помилок є експоненційною, а функція виявлення відмов – сталою. Цільова функція представлена виразом:

$$m(t) = \frac{ab(e^{at} - e^{-bt})}{a + b}, \quad (7.15)$$

де  $a(t)$  – коефіцієнт внесення помилок під час фази відлагодження;

$a(t)$  – очікувана кількість початкових помилок;

$b(t)$  – коефіцієнт виявлення помилок.

Інша модель недосконалого відлагодження Ямади [21] розглядає незмінну функцію внесення помилок та сталу функцію виявлення відмов. Цільова функція має вигляд:



$$m(t) = a(1 - e^{-bt})\left(1 - \frac{a}{b}\right) + \alpha at, \quad (7.16)$$

де  $a(t)$  – коефіцієнт внесення помилок під час фази відлагодження;

$a(t)$  – очікувана кількість початкових помилок;

$b(t)$  – коефіцієнт виявлення помилок.

Капур та Гарг [22] представили процес недосконалого відлагодження в моделі Гоеля–Окумото. Вони припустили, що відношення кількості видалених помилок до кількості помилок, що залишились в системі, зменшується через ефект недосконалого відлагодження. Чанг та Лью [23] запропонували модель негауссового простору станів для формулювання ефекту недосконалого відлагодження в надійності ПЗ. Шир [24] розробив модель зростання надійності ПЗ, що містить ефект недосконалого відлагодження та проблему розладнання. Капур та ін. [25] представили дискретну модель зростання надійності ПЗ з логістичним коефіцієнтом видалення помилок та концепцію двох типів недосконалого відлагодження. Прасад та ін. [26] використовували недосконале відлагодження та проблему розладнання в своїй моделі зростання надійності ПЗ, що базувалась на добре відомому експоненційному розподілі.

Фам та Нордман [19] сформулювали узагальнену модель надійності програмного забезпечення на основі неоднорідного пуассонового процесу та вивели аналітичний вираз для цільової функції. Моделі надійності ПЗ на основі неоднорідного пуассонового процесу з урахуванням недосконалого відлагодження ґрунтуються на таких припущеннях:

1) під час видалення помилки, що стала причиною відмови, існує ймовірність внесення нових помилок;

2) ймовірність виявлення помилки в програмі пропорційна кількості помилок, що залишились у системі.

### **7.12. Моделі надійності програмного забезпечення з урахуванням розподілу зусиль тестування**

Для кращого опису поведінки надійності ПЗ використовують моделі надійності з використанням функцій зусиль тестування [27]. Зусилля тестування визначаються як людино-години, кількість тестових випадків, час роботи процесора, які були затрачені під час фази тестування, а їх розподіл в часі визначається виглядом функції, використаної в певній моделі надійності. При цьому що більше зусиль тестування було витрачено під час розроблення програмного засобу, то менша кількість помилок залишається в коді програми і, відповідно, меншою є інтенсивність відмов ПЗ.

Найчастіше в моделях надійності ПЗ використовують такі функції розподілу зусиль тестування:

- постійна,
- експоненційна,
- логістична,
- функції розподілів Релея та Вейбулла [28–30].

Експоненційний та релеїв розподіли можна розглядати як часткові випадки розподілу Вейбулла.

#### *Постійна функція.*

За цією функцією розглядають зусилля тестування як сталу величину. У класичних моделях зростання надійності ПЗ дослідники вважають, що зусилля тестування програмного забезпечення є константою:

$$w(t)=w_0, \quad (7.17)$$

Кумулятивні зусилля тестування можна отримати за виразом:

$$W(t) = wt, \quad (7.18)$$

Як видно з (7.18), загальні зусилля тестування прямують до безмежності. У випадку, якщо функція зусиль тестування не розглядається, можна вважати, що використовується постійна функція з одиничним значенням:

$$w(t) = 1, \quad (7.19)$$

*Розподіл Вейбулла.*

У випадках, коли розподіл зусиль тестування погано описується експоненційною кривою чи кривою Релея, використовують функцію розподілу Вейбулла, яка має вигляд [30]:

$$w(t) = N\beta m t^{m-1} e^{-\beta t^m}, \quad (7.20)$$

де  $N$  – очікувана загальна кількість зусиль тестування, необхідна для тестування ПЗ;

$\beta$ ,  $m$  – параметри масштабу та форми відповідно.

Значення  $m > 3$  не використовуються при описі розподілу зусиль тестування ПЗ, оскільки в цьому випадку форма кривої не відповідає реальним процесам розроблення ПЗ.

Кумулятивні зусилля тестування в цьому випадку записують як:

$$W(t) = N(1 - e^{-\beta t^m}), \quad (7.21)$$

*Експоненційний розподіл.*

Така крива є частковим випадком кривої Вейбулла при  $m=1$ .

Експоненційна крива є зручною для опису середовища тестування, в якому показник зусиль тестування монотонно знижується [31].

Експоненційну функцію зусиль тестування описують виразом:

$$w(t) = N\beta e^{-\beta t} \quad (7.22)$$

а кумулятивну функцію зусиль тестування – виразом

$$W(t) = N(1 - e^{-\beta t}), \quad (7.23)$$

де  $N$  – очікувана загальна кількість зусиль тестування, необхідна для тестування ПЗ;

$\beta$  – параметр масштабу.

#### *Розподіл Релея.*

Крива Релея є частковим випадком функції розподілу зусиль тестування Вейбулла при  $m=2$ . Функція зусиль тестування Релея спочатку збільшується, досягаючи свого піка, і потім спадає, зменшуючи зусилля тестування асимптотично до нуля [30]:

$$w(t) = N\beta t e^{-\frac{\beta}{2}t^2}, \quad (7.24)$$

Кумулятивні зусилля тестування в такому випадку визначаються як:

$$W(t) = N(1 - e^{-\frac{\beta}{2}t^2}), \quad (7.25)$$

У (7.24) і (7.25):

$N$  – очікувана загальна кількість зусиль тестування, необхідна для тестування ПЗ;

$\beta$  – параметр масштабу.

*Логістичний розподіл.*

Логістичну криву вперше запропоновано в [32] як альтернативу кривій Релея. Вони мають подібну поведінку, окрім початкової стадії процесу тестування. Логістичну криву зусиль тестування записують як:

$$w(t) = \frac{NA\eta}{\left(e^{\frac{\eta t}{2}} + Ae^{-\frac{\eta t}{2}}\right)^2}, \quad (7.26)$$

де  $A$  – параметр-константа;

$\eta$  – міра споживання затрат на тестування;

$N$  – кумулятивні зусилля тестування.

Кумулятивні зусилля тестування у випадку логістичного розподілу мають вигляд:

$$W(t) = \frac{N}{1 + Ae^{-\eta t}}, \quad (7.27)$$

*Логарифм логістичної функції зусиль тестування.*

Бохарі та Ахмад [32] запропонували логарифм логістичної функції зусиль тестування для прогнозування поведінки помилок та відмов ПЗ. Вони показали, що логарифм логістичної функції зусиль тестування добре підходить і є достатньо гнучким для оцінювання надійності програмних продуктів різних класів [10].

Вигляд логарифма логістичної функції розподілу зусиль тестування є таким:

$$w(t) = \frac{N\left(\frac{t}{\lambda}\right)^{-\beta}}{\left(1 + \left(\frac{t}{\lambda}\right)^{-\beta}\right)^2 t}, \quad (7.28)$$

де  $N$  – кумулятивні зусилля тестування;

$l > 0$  – параметр масштабу;

$b > 0$  – параметр форми.

Функція кумулятивних зусиль тестування в такому випадку має вигляд:

$$W(t) = \frac{N}{1 + \left(\frac{t}{\lambda}\right)^{-\beta}} \quad (7.29)$$

Найпростіша модель зростання надійності ПЗ з урахуванням недосконалого відлагодження, що використовує функцію розподілу зусиль тестування, повинна відповідати таким критеріям:

- процес видалення помилок розглядається як неоднорідний процес Пуассона;
- ПЗ є суб'єктом, в якому у випадковий час може статись відмова, спричинена проявом помилок, які залишились у системі;
- кількість помилок, виявлених на проміжку часу  $(t, t + \Delta t)$  при поточних затратах на тестування, є пропорційною до кількості помилок, що залишились в системі;
- коефіцієнт пропорційності є однорідним і не залежить від часу;
- поведінка зусиль тестування залежить від часу і може бути змодельована логістичним розподілом;
- кожного разу, коли виникає відмова, помилка, що її спричинила, негайно і коректно видаляється, при цьому жодних нових помилок в систему не вноситься;
- виправлення помилок займає незначний час і виявлені помилки гарантовано видаляються [33].

Отже, модель зростання надійності ПЗ, що містить функцію зусиль тестування, можна описати таким виразом:

$$\frac{dm(t)}{dt} \frac{1}{w(t)} = r[a - m(t)], \quad (7.30)$$

де  $w(t)$  – функція зусиль тестування;

$r$  – коефіцієнт виявлення відмов;

$a$  – коефіцієнт затрат зусиль тестування в логістичній функції;

$m(t)$  – цільова функція, що обчислює очікувану кількість програмних відмов на час  $t$ .

Вираз (7.30) містить дві компоненти, що впливають на кількість виявлених помилок: функцію зусиль тестування  $w(t)$  та коефіцієнт виявлення відмов  $r$ . Розв'язуючи диференціальне рівняння (7.30) з граничною умовою  $m(0)=0$ , отримаємо:

$$m(t) = a(1 - \exp[-r(W(t) - W(0))]) = a(1 - \exp[-rW(t)]), \quad (7.31)$$

де  $W(t)$  – кумулятивні зусилля тестування (наприклад, процесорні години, об'єм тестових випадків, людино-години та ін.) на час  $t$ .

Вираз (7.31) є моделлю на основі неоднорідного процесу Пуассона з цільовою функцією, яка враховує затрати на зусилля тестування. Затрачені зусилля тестування вказують, наскільки ефективно помилки виявляються в ПЗ. Тому ця функція відіграє важливу роль в моделюванні надійності ПЗ і може бути описана розподілами різного типу [33].

Експоненціальна модель Ямади [34] враховує зусилля тестування та містить експоненціальну функцію зусиль тестування. Ця модель є увігнутою, а цільова функція має вигляд:

$$m(t) = a(1 - e^{-ra(1 - e^{-bt})}), \quad (7.32)$$

Модель Ямади–Релея [34] враховує зусилля тестування та містить релесву функцію зусиль тестування. Ця модель належить до S-подібних, цільова функція має вигляд:

$$m(t) = a(1 - e^{-ra(1 - e^{-\frac{bt^2}{2}})}), \quad (7.33)$$

Раджу [35] дослідив інтеграцію логарифм-логістичної функції зусиль тестування в увігнуту S-подібну модель зростання надійності ПЗ на основі неоднорідного процесу Пуассона для кращого опису процесу виявлення відмов при недосконалому відлагодженні. Ямада та ін. [36] модифікували модель Гоеля–Окумото та об'єднали її з концепцією зусиль тестування. Так вони створили модель зростання надійності ПЗ, на основі неоднорідного процесу Пуассона для кращого опису процесу відмов ПЗ. Хуанг [33] запропонував нову модель зростання надійності ПЗ що використовує логістичну функцію зусиль тестування. Логістична функція зусиль тестування – це крива оцінювання розробки та зусиль тестування ПЗ з хорошими характеристиками прогнозування. В роботі [33] Хуанг узагальнив логістичну функцію зусиль тестування. Узагальнена функція зусиль тестування адекватніше описує процес розроблення ПЗ, тому її використання є більш достовірним для опису розподілу затрат на тестування ПЗ.



## 8. Моделі надійності ПЗ [34]

Всі моделі надійності можна класифікувати по тому, який з перерахованих процесів вони підтримують (пророчать, прогнозні, що вимірюють і т.д.). Потрібно відзначити, що моделі надійності, які в якості вихідної інформації використовують дані про інтервали між відмовами, можна віднести і до вимірює, і до оцінюючим в рівній мірі. Деякі моделі, засновані на інформації, отриманої в ході тестування ПЗ, дають можливість робити прогнози поведінки ПЗ в процесі експлуатації.

*Аналітичні* моделі дають можливість розраховувати кількісні показники надійності, ґрунтуючись на даних про поведінку програми в процесі тестування (вимірюють та оцінюють моделі).

*Емпіричні* моделі базуються на аналізі структурних особливостей програм. Вони розглядають залежність показників надійності від числа міжмодульних зв'язків, кількості циклів в модулях і т.д. Часто емпіричні моделі не дають кінцевих результатів показників надійності, проте вони включені в класифікаційну схему, оскільки розвиток цих моделей дозволяє виявляти взаємозв'язок між складністю аналітико-синтетичне опрацювання документів (АСОД) і його надійністю. Ці моделі можна використовувати на етапі проектування ПЗ, коли здійснюється розбивка на модулі і відома його структура.

Аналітичні моделі представлені двома групами: динамічні моделі і статичні. У *динамічних* поведінку ПЗ (поява відмов) розглядається в часі. У *статичних* моделях поява відмов не пов'язують з часом, а враховують тільки залежність кількості помилок від числа тестових прогонів (по області помилок) або залежність кількості помилок від характеристики вхідних даних (по області даних).

Для використання динамічних моделей необхідно мати дані про появу відмов у часі. Якщо фіксуються інтервали кожного

відмови, то виходить безперервна картина появи відмов у часі (група динамічних моделей з безперервним часом). З іншого боку, може фіксуватися тільки число відмов за довільний інтервал часу.

### 8.1. Динамічні моделі надійності

#### *Модель Шумана*

Вихідними даними для моделі Шумана, яка відноситься до динамічних моделей дискретного часу, збираються в процесі тестування АСОД протягом фіксованих або випадкових тимчасових інтервалів. Кожен інтервал - це стадія, на якому виконується послідовність тестів і фіксується деяке число помилок.

Модель Шумана може бути використана при певному образі організованою процедурою тестування. Використання моделі Шумана припускає, що тестування поводить у кілька етапів. Кожен етап являє собою виконання на повному комплексі розроблених тестових даних. Виявлення помилки реєструється, але не виправляється. По завершенні етапу на основі зібраних даних про поведінку ПЗ на черговому етапі тестування може бути використана модель Шумана для розрахунку кількісних показників надійності. При використанні моделі Шумана передбачається, що вихідне кількість помилок в програмі постійно, і в процесі тестування може зменшуватися в міру того, як помилки виявляються і виправляються.

Передбачається, що до початку тестування в ПЗ мається  $E_t$  помилок. У плінні часу тестування  $t_l$  в системі виявляється  $E_c$  помилок у розрахунку на команду в машинній мові.

Таким чином, питомий число помилок на одну машинну команду, що залишилася в системі після  $t_l$  часу тестування, одно:

$$E(t)_r = \frac{E_t}{I_t} E(t)_c, \quad (8.1)$$

де  $I_t$  - загальне число машинних команд, яке передбачається в рамках етапу тестування.

Припускаємо, що значення функції частоти відмов  $Z(t)$  пропорційно числу помилок, що залишилися в ПЗ після витраченого на тестування часу  $t$ .

$$Z(t) = C \cdot E(t), \quad (8.2)$$

де  $C$  - деяка константа,

$t$  - час роботи ВП без відмови, ч.

Тоді, якщо час роботи ПЗ без відмови  $t$  відраховується від точки  $t = 0$ , а  $t_1$  залишається фіксованим, функція надійності, або ймовірність безвідмовної роботи на інтервалі часу від  $0$  до  $t$ , дорівнює:

$$\begin{aligned} R(t, t_1) &= e^{\left(c \left(\frac{E_t}{I_t} - E(t_1)c\right)t\right)} \\ R(t, t_1) &= e^{\left(c \left(\frac{E_t}{I_t} - E(t_1)E(t_1)c\right)t\right)}, \end{aligned} \quad (8.3)$$

$$t_{cp} = \frac{1}{\left(c \left(\frac{E_t}{I_t} - E(t_1)c\right)t\right)}, \quad (8.4)$$

З величин, що входять у формули (8.2) і (8.3), не відомі початкове значення помилок в ПЗ ( $E_t$ ) і коефіцієнт пропорційності -  $C$ . Для їх визначення удаються до наступних міркувань.

У процесі тестування збирається інформація про час і кількості помилок на кожному прогоні, т.е загальний час тестування  $t_1$  складається з часу кожного прогону:

$$t_1 = t_1 + t_2 + t_3 + \dots + T_n \quad (8.5)$$

Припускаючи, що інтенсивність появи помилок постійна і дорівнює  $O$ , можна обчислити її як число помилок в одиницю часу:

$$O = \frac{\sum_{i=1}^k A_i}{t_1}, \quad (8.6)$$

де  $A_i$  - кількість помилок на  $i$ -му прогоні.

$$t_{cp} = \frac{t_i}{\sum_{i=1}^k A_i}, \quad (8.7)$$

Маючи дані для двох різних моментів тестування  $t_A$  і  $t_b$ , які вибираються довільно з урахуванням вимоги, щоб  $E_c(t_b) > E_c(t_A)$ , можна зіставити рівняння (8.4) і (8.6) при  $t_A$  і  $t_b$ :

$$\frac{1}{O(t_{1A})} = \frac{1}{c \left( \frac{E_t}{I_t} - E(t_{1A})c \right)}, \quad (8.8)$$

Обчислюючи відношення (8.7) і (8.8), отримаємо

$$E_t = \frac{E_T \left( \frac{o(t_{1b})}{o(t_{1A})} E(t_{1A})_C - E(t_{1b})_C \right)}{\frac{o(t_{1b})}{o(t_{1A})}^{-1}}, \quad (8.9)$$

Підставивши отриману оцінку параметрів  $E_t$  у вираз (8.8), отримаємо оцінку для другого невідомого параметра:

$$C = \frac{o(t_{1A})}{\frac{E_t - E(t_{1A})_C}{I_t}}, \quad (8.10)$$

Отримавши невідомі  $E_t$  і  $C$ , можна розрахувати надійність програми по формулі (8.3).

Гідність цієї моделі полягає в тому, що можна виправляти помилки, вносячи зміни в текст програми в ході тестування, не розбиваючи процес на етапи, щоб задовольнити вимогу сталості числа машинних інструкцій.

### *Модель La Padula*

За цією моделлю виконання послідовності тестів проводитиметься в  $m$  етапів. Кожен етап закінчується внесенням змін (виправлень) у ПЗ.

Зростаюча функція надійності базується на числі помилок, виявлених в ході кожного тестового прогону.

Надійність ПЗ в течій  $i$ -го етапу:

$$R(t) = R(\infty) - \frac{A}{(i)}, \quad (8.11)$$

де  $i = 1, 2, \dots, n$ ,

$A$  - параметр зростання;

Гранична надійність ПЗ:

$$R(\infty) = \lim_{i \rightarrow \infty} R(i), \quad (8.12)$$

Ці невідомі величини можна знайти, вирішивши наступні рівняння:

$$\left( \sum_{i=1}^m \left( \frac{S_i - m_i}{S_i} - R(\infty) \right) + \frac{A}{i} \right) = 0, \quad (8.13)$$

де  $S_i$  - число тестів;

$m_i$  - число відмов під час  $i$ -го етапу;

$m$  - число етапів;

$i = 1, 2 \dots m$ .

Визначуваний по цій моделі показник є надійність АСОД на  $i$ -му етапі.

$$R(i) = R(\infty) - \frac{A}{i}, \quad (8.14)$$

де  $i = m + 1, m + 2 \dots$

Перевага даної моделі полягає в тому, що вона є прогнозою  $i$ , ґрунтуючись на даних, отриманих в ході тестування, дає можливість передбачити ймовірність безвідмовної роботи програми на подальших етапах її виконання.

## 8.2. Статичні моделі надійності

Статичні моделі принципово відрізняються від динамічних насамперед тим, що в них не враховується час появи помилок в процесі тестування і не використовується ніяких припущень про

поведінку функції ризику. Ці моделі будуються на твердому статичному фундаменті.

### *Модель Міллса*

Використання цієї моделі передбачає необхідність перед початком тестування штучно вносити в програму (засмічувати) деяка кількість відомих помилок. Помилки вносяться випадковим чином і фіксуються в протоколі штучних помилок. Фахівець, що проводить тестування, не знає ні кількості помилок, ні характеру внесених помилок до моменту оцінки показників надійності за моделлю Міллса. Передбачається, що всі помилки (як природні, так і штучно внесені) мають рівну ймовірність бути знайденими в процесі тестування.

Тестуючи програму протягом деякого часу, збирається статистика про помилки. У момент оцінки надійності по протоколу штучних помилок всі помилки діляться на власні і штучні. Співвідношення:

$$N = (S \cdot n) / V , \quad (8.15)$$

дає можливість оцінити  $N$  - початкову кількість помилок в програмі. У даному співвідношенні, яке називається формулою Міллса,  $S$  - кількість штучно внесених помилок,  $n$  - число знайдених власних помилок,  $V$  - число виявлених до моменту оцінки штучних помилок.

Друга частина моделі пов'язана з перевіркою гіпотези від  $N$ . Припустимо, що в програмі є  $n$  власних помилок, і внесемо в неї ще  $S$  помилок. В процесі тестування були виявлені всі  $S$  внесених помилок і  $n$  власних помилок.

Тоді за формулою Міллса ми припускаємо, що спочатку в програмі було  $N = n$  помилок. Імовірність, з якою можна

висловити таке припущення, можливо розрахувати за наступним співвідношенням:

$$C = 1, \quad \text{як що } n > K$$

$$C = \frac{S}{S + K + 1}, \quad \text{як що } n \leq K, \quad (8.16)$$

Таким чином, величина  $C$  є мірою довіри до моделі і показує ймовірність того, наскільки правильно знайдено значення  $N$ .

Ці два пов'язаних між собою за змістом співвідношення утворюють корисну модель помилок: перше пророкує можливе спочатку були в програмі помилок, а друге використовується для встановлення довірчого рівня прогнозу. Однак формула (8.16) для розрахунку  $C$  не може бути у випадку, коли не виявлені всі штучно розсіювання помилки. Для цього випадку, коли оцінка надійності проводиться до моменту виявлення всіх  $S$  розсіяних помилок, величина  $C$  розраховується за модифікується формулою

$$C = 1, \quad \text{як що } n > K$$

$$C = \frac{\frac{S}{V-1}}{\left(\frac{S+K+1}{V+K}\right)}, \quad \text{як що } n \leq K, \quad (8.17)$$

де чисельник і знаменник формули при  $n \leq K$  є Біноміальними коефіцієнтами виду

$$\frac{a}{b} = \frac{a!}{b!(a-b)!}, \quad (8.18)$$

Наприклад, якщо стверджується, що в програмі немає помилок, а до моменту оцінки надійності виявлено 5 з 10



розсіяних помилок і не виявлено жодної власної помилки, то ймовірність того, що в програмі справді немає помилок, буде дорівнює:

$$C = \frac{\frac{10}{4}}{\frac{11}{5}} = \frac{10! \cdot 5! \cdot 6!}{4! \cdot 6! \cdot 11!} = 0,45, \quad (8.19)$$

Якщо при тих же вихідних умовах оцінка надійності проводиться в момент, коли виявлені 8 з 10 штучних помилок, то ймовірність того, що в програмі не було помилок, збільшується до 0.73. В дійсності модель Міллса можна використовувати для оцінки  $N$  після кожної знайденої помилки. Пропонується під час усього періоду тестування відзначати на графіку число знайдених помилок і поточне значення для  $N$ . Перевагою моделі є простота застосування математичного апарату, наочність і можливість використання в процесі тестування.

Однак вона не позбавлена й ряду недоліків, найістотніші з яких - це необхідність внесення штучних помилок (цей процес погано формалізується) і досить вільне допущення величини  $K$ , яке ґрунтується виключно на інтуїції і досвіді людини, що проводить оцінку, тобто допускається великий вплив суб'єктивного фактора.

### *Модель Липова*

Липов модифікував модель Міллса, розглянувши ймовірність виявлення помилки при використанні різного числа тестів. Якщо зробити те ж припущення, що і модель Міллса, тобто що власні і штучні помилки мають рівну ймовірність бути знайденими, то ймовірність виявлення  $n$  власних і  $V$  внесених помилок дорівнює:

$$Q(n, V) = \frac{m}{n+V} \cdot q^{n+V \cdot (1-q)^{m-n - \frac{V \cdot \frac{N \cdot S}{n \cdot V}}{\left(\frac{N+S}{n+V}\right)}}, \quad (8.20)$$

де  $m$  - кількість тестів, які використовуються при тестуванні;  
 $q$  - ймовірність виявлення помилки в кожному з  $m$  тестів,  
 розраховується за формулою:

$$q = \frac{n+V}{n}, \quad (8.21)$$

де:  $S$  - загальна кількість штучно внесених помилок;

$N$  - кількість власних помилок, що міститься у ПЗ до початку тестування.

Для використання моделі Липова повинні виконуватися наступні умови:

$$\begin{aligned} N &\geq n \geq 0 \\ S &\geq V \geq 0 \\ m &\geq n+V \geq 0, \end{aligned} \quad (8.22)$$

Модель Липова доповнює модель Міллса, давши можливість оцінити ймовірність виявлення певної кількості помилок до моменту оцінки.

## **Висновки.**

Моделі зростання надійності використовують на етапі тестування ПЗ. Після введення програми в експлуатацію при визначенні її надійності враховують додавання нових та усунення застарілих програмних модулів, виправлення виявлених помилок тощо. На цьому етапі використовують еволюційні моделі надійності. Тобто на різних етапах життєвого циклу ПЗ використовують різні типи моделей надійності ПЗ, кожен з яких, крім того, містить значну кількість моделей, побудованих з урахуванням різноманітних припущень про методології та технології розроблення ПЗ, поведінку його відмов тощо. Все це значно ускладнює створення єдиної узагальненої класифікації моделей надійності ПЗ, і це завдання сьогодні все ще потребує свого вирішення.

Для кращого опису поведінки надійності ПЗ використовують моделі надійності з використанням функцій зусиль тестування, які визначаються як людино-години, кількість тестових випадків, час роботи процесора, які були затрачені під час фази тестування. При цьому що більше зусиль тестування було витрачено під час розроблення програмного засобу, то менша кількість помилок залишається в коді програми і, відповідно, меншою є інтенсивність відмов ПЗ. Розглянуто найпоширеніші функції розподілу зусиль тестування, такі як постійна, експоненційна, логістична, функції розподілів Релея та Вейбулла. Наведено приклади моделей надійності ПЗ, які містять розподіл зусиль тестування. Показано, що такі моделі адекватніше описують інтенсивність відмов сучасного складного і масштабного ПЗ.

В основу більшості моделей надійності ПЗ покладено припущення про ідеальне відлагодження, коли під час процесу тестування виявляють відмову ПЗ, знаходять відповідну помилку та виправляють її, але при цьому нові помилки не вносяться.

Однак це припущення є спрощенням реального процесу розроблення програмних систем, оскільки в процесі розроблення не завжди вдається видалити помилку із системи, під час видалення однієї помилки можуть бути внесені інші, відмова може бути спричинена кількома взаємопов'язаними помилками тощо.

Розглянуто найвідоміші моделі надійності ПЗ, в яких враховано явище недосконалого відлагодження. На жаль, як в практичних, так і в теоретичних дослідженнях надійності ПЗ явище недосконалого відлагодження використовують недостатньо широко. З урахуванням ускладнення технологій та засобів розроблення програмних продуктів та зростання складності сучасних програмних систем неврахування недосконалого відлагодження під час аналізу надійності ПЗ призводить до того, що моделі надійності стають все менш адекватними реальному об'єкту дослідження, а результати, отримані з їх використанням, перестають бути достовірними.

## Список літератури.

1. Фадин А. А. и др. / под ред. А. С. Маркова и В. Л. Цирлова. – М.: Сводный отчёт по безопасности программного обеспечения в России и мире за 2010 год // НПО “Эшелон”. – 2011. – 34 с.
2. Смагин В. А. Основы теории надежности программного обеспечения. – СПб.: ВКА им. А. Ф. Можайского, 2009. – 336 с.
3. Черкесов Г. Н. Надежность аппаратно-программных комплексов. – СПб.: “Питер”, 2005. – 479 с.
4. William F. Software reliability modeling survey – Naval Surface Warfare Center, 1996.
5. Luy M. R. Handbook of software reliability engineering - IEEE Computer Society Press, 1996.
6. Ch. Ali Asad, Muhammad Irfan Ullah, Muhammad Jaffar-Ur Rechman An approach for software reliability model selection – IEEE Computer Society Press, 2004.
7. Shooman M. L. Operational Testing and Software Reliability Estimation During Program Developments – IEEE Computer Society, 1973.
8. Coutinho J. deS. Software Reliability Growth – IEEE Symposium on Computer Software Reliability, 1973.
9. Васілевський, О. М. Нормування показників надійності технічних засобів : навчальний посібник / О. М. Васілевський, О. Г. Ігнатенко. – Вінниця : ВНТУ, 2013. – 160 с.
10. Musa J. D., Okumoto, K. Software Reliability Models: Concepts, Classification, Comparisons, and Practice – Electronic Systems Effectiveness and Life Cycle Costing, 2000.
11. Moranda P. L., Jelinski Z. Final Report on Software Reliability Study – McDonnell Douglas Astronautics Company, 1972.
12. Гуда А. Н., Калинин Т. С., Чернов А. В. Реализация надежного программного обеспечения задач технической

диагностики информационно-управляющих систем // Известия высших учебных заведений. Северо-Кавказский регион.

Технические науки. № 4. – 2011. – С. 26–31.

13. Чернов А. В., Паращенко И. Г. Классификация моделей надежности программного обеспечения // Электронный научный журнал “Инженерный вестник Дона”, 2007–2015.

14. Белявский Г. И., Чернов А. В. Математические модели линейных контролируемых дискретных динамических систем // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. – 2009. – № 2. – С. 145–151.

15. ДСТУ 3918-99 (ISO/IEC 12207: 1995) Інформаційні технології. Процеси життєвого циклу програмного забезпечення. – К.: Держстандарт України, 2002. – 49 с.

16. Cobra Rahmani. Exploitation of Quantitative Approaches to Software Reliability / Cobra Rahmani, Azad Azadmanesh // University of Nebraska at Omaha, 2008. – 32 p.

17. Pham H. Software Reliability Models for Critical Applications/H. Pham, M. Pham//EGG–2663 Technical Report. Idaho National Engineering Laboratory, EG&G Idaho Inc. -1991. -98p.

18. A. L. Goel, Software Reliability Models: Assumptions, Limitations, and Applicability, // IEEE Trans. Software Eng., vol. 11, pp. 1411–1423, 1985.

19. Pham H. System software reliability // Springer-Verlag ondon Limited, 2006. – 440 p.

20. H. Pham, L. Nordmann, and X. Zhang, A general imperfect software debugging model with s-shaped fault detection rate // IEEE Trans. Reliability, vol. 48, pp. 169–175, June 1999.

21. H. Pham and X. Zhang, “An NHPP software reliability models and its comparison” // International J. Of Reliability, Quality and Safety Engineering, vol. 14, no. 3, pp. 269–282, 1997.

22. S. Yamada, K. Tokuno, and S. Osaki, "Imperfect debugging models with fault introduction rate for software reliability assessment," // *International J. Syst. Science*, vol. 23, no. 12, 1992.
23. Kapur, P. K., and Garg, R. B. 1990. Optimal release policy for software reliability growth models under imperfect debugging. *Oper. Res. RAIRO*. 24(3), 295–305.
24. Chang, Y. C., and Liu, C. T. 2009. A generalized JM model with applications to imperfect debugging in software reliability // *Appl. Math. Model.* 33, 3578–3588.
25. Shyr, H.J. 2003. A stochastic software reliability model with imperfect-debugging and changepoint.// *J. Syst. Software*. 66(2), 135–141.
26. Kapur, P. K., Singh, O. M. P., Shatnawi, O., and Gupta, A. 2006. A discrete NHPP model for software reliability growth with imperfect fault debugging and fault generation. // *Int. J. Perform. Eng.* 2(4), 351–368.
27. Prasad, R.S., Raju, O.N., and Kantam, R.R.L. 2010. SRGM with imperfect debugging by genetic algorithms. // *Int. J. Software Eng. Appl.* 1(2), 66–79.
28. Ce Zhang, Gang Cui, Hongwei Liu, Fanchao Meng, Shixiong Wu. "A Unified and Flexible Framework of Imperfect Debugging Dependent SRGMs with TestingEffort" // *Journal of Multimedia*, Vol. 9, no. 2, FEBRUARY 2014.
29. Chin-Yu Huang. An Assessment of Testing-Effort Dependent Software Reliability Growth Models [Text]/ Chin-Yu Huang, Sy-Yen Kuo, Michael R. Lyu // *IEEE Transactions on Reliability*, Vol. 56, No. 2, June 2007. – C. 198–211.
30. Sy-Yen Kuo. Framework for Modeling Software Reliability, Using Various Testing-Efforts and Fault-Detection Rates [Text] / Sy-Yen Kuo, Chin-Yu Huang, Michael R. Lyu // *IEEE Transactions on Reliability*, Vol. 50, no. 3, September 2001. – C. 310–320.

31. Ahmad N. Software Reliability Growth Models with Log-logistic Testing-Effort Function: A Comparative Study [Text]/ N. Ahmad, Md. Zafar Imam // International Journal of Computer Applications (0975 – 8887) Vol. 75. – No.12, August 2013. – С. 6–10.
32. Yamada S. A testing-effort dependent software reliability model and its application [Text] / Shigeru Yamada, Hiroshi Ohtera, Hiroyuki Narihisa // Microelectron. Reliab, Vol. 27, No. 3, 1987. – С. 507–522.
33. Parr F. An Alternative to the Rayleigh Curve Model for Software Development Effort // IEEE Transactions on Software Engineering, Vol.6, no. 3, May 1980, pp. 291–296.
34. В. Яковина, В. Мацелюх. Огляд і аналіз моделей надійності програмного забезпечення. Вісник Національного університету Львівська політехніка. Комп'ютерні науки та інформаційні технології. Вип. 864, С. 130-140.
35. C. Y. Huang, M. R. Lyu, “Optimal Release Time for Software Systems Considering Cost, Testing-Effort, and Test Efficiency,” // IEEE Trans. on Reliability, Vol. 54, No. 4, pp. 583–591, Dec. 2005.
36. H. Pham, “Software reliability and cost models: perspectives, comparison and practice” // European J. of Operational Research, vol. 149, pp.475–489, 2003.
37. Raju, O.N. 2011. Software reliability growth models for the safety critical software with imperfect debugging. // Int. J. Comput. Sci.Eng. 3(8), 3019–3026.
38. S. Yamada, J. Hishitani, and S. Osaki, “Software reliability growth model with Weibull testing effort: a model and application,” // IEEE Trans. Rel., vol. R-42, pp. 100–105, 1993.
39. Електронний ресурс. URL:  
<https://uadoc.zavantag.com/text/300/index-1.html>



### Контрольні запитання.

1. Проаналізуйте який тип програмного забезпечення є більш надійний:

2. Системне програмне забезпечення;
3. Операційна система
4. Сервісні програмні засоби
5. Як ви оцінюєте – що впливає на ступінь надійності

прикладних програм?

6. У чому полягає практичне значення побудови систематики техніки?

7. У чому полягає користь від класифікації технічних систем?

8. Чи поділяєте ви розділ на категорії ПЗ за ступенем оригінальності, викладений у лекційному матеріалі?

9. Проаналізуйте яка властивість надійності є більш інформативною для споживача: безвідмовність; довговічність; збережність; ремонтопридатність.

10. Як на вашу думку оцінюється ремонтпридатність ПЗ?

11. У п. 1 та 2 питань є різні визначення надійності. У чому полягає різниця і чим вона зумовлена?

12. Інтенсивність відмов ПЗ  $10^{-5}$ . Це добре чи погано?

13. До якої причини відмов ПЗ можна віднести дію електромагнітного поля? За яких умов дія електромагнітного поля здатна привести до відмови ПЗ?

14. У чому особливості програмного забезпечення як об'єкта аналізу надійності?

15. Зобразіть залежності зміни надійності програмних та апаратних засобів ІС від часу.

16. Дайте визначення поняття «надійність програмного забезпечення» відповідно до стандарту ДСТУ ISO/IEC 9126-1:2013.

17. Назвіть основні причини помилок програмного забезпечення.

18. Які джерела помилок (загрози надійності) програмного забезпечення?

19. У чому полягають відмінності програмних та апаратних відмов?

20. Наведіть класифікацію моделей надійності програмних засобів.

21. Як здійснити оцінку надійності програмного забезпечення при використанні аналітичних моделей?

22. На яких припущеннях побудовано модель Шумана?

23. У чому полягає відмінність між статичними і динамічними моделями надійності програмних засобів?

24. Назвіть основні напрямки підвищення надійності ІС.

25. Яка роль кластерів у підвищенні надійності ІС?

26. Назвіть сучасні методи кластеризації.

27. Що таке ПЗ Docker і для чого воно використовується?

28. Що таке ПЗ Apache Benchmark і для чого воно використовується?

29. Опишіть основні принципи і особливості протоколу НТТР.

30. Чи кожен web-застосунок має використовувати протокол НТТР? Відповідь обґрунтуйте. (Вказівка: див. Протоколи 7-го рівня мережевої моделі OSI).

31. Поясніть принцип роботи тестування на навантаження, практичний сенс виконання такого типу тестування та яку інформацію можна отримати про систему внаслідок його проведення.

32. Поясніть своє розуміння поняття відмови для ПЗ та за допомогою яких критеріїв таке поняття може бути заданим. Чи можуть бути системи де відмови допустимі?

33. Наведіть приклади систем, де прийнятні вимоги до часу відповіді можуть бути з більшим або меншим часом, ніж «правило 3-х секунд». Поясніть як та чому величину прийнятного часу може бути зменшено або збільшено з точки зору логіки використання застосунку.

34. У яких нормативних документах може бути регламентовано та описано умови відмов та критерії до допустимої швидкодії системи?

## Приклади тестових завдань

Дайте відповіді на наведені питання згідно прикладу.

1. *Вибрати вірні відповіді до тестових запитань, наведених нижче.*

Практичне значення побудови систематики техніки зв'язано з:

- a) створенням єдиної науково-обґрунтованої класифікації патентів і стандартів.
- b) створенням національних і міжнародних активних інформаційних ресурсів автоматизованих банків даних і банків знань на електронних носіях.
- c) створенням пошукових систем, які використовують Інтернет.
- d) актуальністю ціноутворення з точки зору складності технічних систем.

2. *Вибрати не вірні відповіді у переліку ознак у твердженні:*

Технічні системи класифікуються за такими ознаками:

- a) за функцією;
- b) за типом операцій;
- c) за принципом здійснення робочої дії;
- d) за характером функціонування;
- i) за рівнем складності;
- f) за способом виготовлення;
- g) за способом упорядкування більш низьких рівнів технічної системи;

- h) за винахідницьким рівнем;
- i) за способом роботи оператора;
- j) за ступенем оригінальності;
- к) за типом виробництва (одиничне, серійне, масове);
- l) за експлуатаційними властивостями.

*3. Вибрати не вірні відповіді у переліку ознак у твердженні:*

За ступенем оригінальності ПЗ технічні системи можна розділити на категорії, що подані нижче.

- a) запозичене ПЗ;
- b) придбане;
- c) допрацьоване ПЗ;
- d) протестоване;
- i) модифіковане ПЗ;
- f) нові технічні системи

*4. Вибрати вірні відповіді у переліку ознак у твердженні:*

a) Системне програмне забезпечення призначене для керування роботою складових комп'ютера та обміном даними між ними, діагностування та усунення недоліків у роботі комп'ютера, автоматизації процесу обробки даних, організації обміну даними між користувачем і комп'ютером.

b) Системне програмне забезпечення призначене для керування роботою складових комп'ютера та обміном даними між ними.

c) Системне програмне забезпечення призначене для керування роботою складових комп'ютера, діагностування та усунення недоліків у роботі комп'ютера, автоматизації процесу обробки даних, організації обміну даними між користувачем і комп'ютером.

*5. Вибрати не вірні відповіді у переліку ознак у твердженні:*

Системне (базове) програмне забезпечення слугує для організації й ефективної роботи комп'ютера та охоплює:

а) операційні системи (ОС), основною функцією яких є керування ресурсами (фізичними та логічними) і процесами обчислювальних систем;

б) мережеве програмне забезпечення, призначене для керування спільними ресурсами в розподілених обчислювальних системах;

с) сервісні програми, до яких належать файлові менеджери; утиліти, тобто системні програми для виконання службових функцій (антивіруси, архіватори, програми для обслуговування дисків тощо).

*6. Вибрати не вірні відповіді у переліку ознак у твердженні:*

Операційна система – це комплекс програм, що забезпечують:

а) керування роботою пристроїв комп'ютера та обмін даними між ними;

б) зберігання даних в оперативній пам'яті та на зовнішніх носіях;

с) виконання інших програм;

д) збір статистичних даних про роботу програм;

е) контроль ресурсів комп'ютера та операційної системи.

*7. Вибрати не вірні відповіді у переліку ознак у твердженні:*

Класифікація сервісних програмних засобів наступна:

а) Диспетчери файлів (файлові менеджери).

б) Засоби обміну.

- c) Засоби стиснення даних (архіватори).
- d) Засоби діагностики.
- e) Програмні утиліти.

8. *Вибрати не вірні відповіді у переліку ознак у твердженні:*

До прикладних програм загального призначення належать:

- a) текстові процесори;
- b) програми архівації;
- c) табличні процесори;
- d) системи інсталяції;
- e) програми створення презентацій.

9. *Вибрати не вірні відповіді у переліку ознак у твердженні:*

До прикладних програм спеціального призначення належать:

- a) системи ведення бухгалтерського обліку;
- b) системи захисту даних;
- c) правові БД;
- d) БД службового користування.

10. *Вибрати вірні відповіді у переліку ознак у твердженні:*

a) Транслятор – це програма, за допомогою якої текст програми перекладається (транслюється) на машинний код.

b) Транслятор - це програма, за допомогою якої текст програми, написаний довільною мовою програмування, перекладається (транслюється) на машинний код.

c) Транслятор - це програма, за допомогою якої текст програми, написаний довільною мовою програмування, перекладається (транслюється) на машинний код для певного

типу контролерів.

*11. Вибрати вірні відповіді у переліку ознак у твердженні:*

а) Транслятори бувають двох типів (інтерпретатори; компілятори).

б) Транслятори бувають трьох типів (інтерпретатори; формалізатори; компілятори).

с) Транслятори бувають чотирьох типів (інтегратори; інтерпретатори; формалізатори; компілятори).

*12. Вибрати вірні відповіді до тестових запитань, наведених нижче.*

а) Згідно ДСТУ ISO 9000:2007 надійність – це збірний термін, який вживають для описування характеристики готовності та чинників, що її зумовлюють: характеристик безвідмовності, ремонтопридатності та забезпеченості технічного обслуговування й ремонту.

б) Згідно ДСТУ ISO 9000:2007 надійність – це термін, який вживають для описування характеристики надійності: безвідмовності, ремонтопридатності та забезпеченості технічного обслуговування й ремонту.

с) Згідно ДСТУ ISO 9000:2007 надійність – сукупність характеристик безвідмовності, ремонтопридатності та забезпеченості технічного обслуговування й ремонту.

д) Згідно ДСТУ ISO 9000:2007 надійність – це збірний, комплексний термін, для описування характеристики готовності об'єкту до виконання роботи та чинників, що її зумовлюють: характеристик безвідмовності, ремонтопридатності та забезпеченості технічного обслуговування й ремонту.



*13. Вибрати вірні відповіді у переліку ознак у твердженні:*

а) Згідно ДСТУ 2860-94 надійність – властивість об'єкта зберігати в часі у встановлених межах значення всіх параметрів, які характеризують здатність виконувати потрібні функції відповідно до технічних умов.

б) Згідно ДСТУ 2860-94 надійність – властивість об'єкта зберігати в часі у встановлених межах значення всіх параметрів, які характеризують здатність виконувати потрібні функції в заданих режимах та умовах застосування, технічного обслуговування, зберігання і транспортування.

с) Згідно ДСТУ 2860-94 надійність – властивість об'єкта зберігати в часі значення всіх параметрів, які характеризують здатність виконувати потрібні функції в заданих режимах та умовах застосування, технічного обслуговування, зберігання і транспортування.

*14. Вибрати вірні відповіді у переліку ознак у твердженні:*

а) Надійність включає в себе такі властивості: безвідмовність, довговічність, ймовірність безвідмовної роботи, ремонтпридатність та інтенсивність відмов.

б) Надійність включає в себе: безвідмовність, довговічність, збережність та ремонтпридатність.

с) Надійність включає в себе: безвідмовність, довговічність, збережність та інтенсивність відмов.

*15. Вибрати вірні відповіді у переліку ознак у твердженні:*

а) Безвідмовність – властивість об'єкта виконувати потрібні функції в певних умовах протягом заданого

інтервалу часу чи наробітку.

б) Безвідмовність - властивість об'єкта виконувати потрібні функції в умовах, заданих у технічному завданні, протягом заданого інтервалу часу.

с) Безвідмовність – властивість об'єкта виконувати потрібні функції в певних умовах протягом заданого інтервалу часу при збереженні з встановлюваного значення імовірність безвідмовної роботи.

*16. Вибрати вірні відповіді у переліку ознак у твердженні:*

а) Інтенсивність відмов  $\lambda(t)$  є показником безвідмовності не ремонтпридатних і не відновлюваних об'єктів;

б) Інтенсивність відмов  $\lambda(t)$  є показником довговічності не ремонтпридатних і не відновлюваних об'єктів;

с) Інтенсивність відмов  $\lambda(t)$  є показником безвідмовності ремонтпридатних і відновлюваних об'єктів;

д) Інтенсивність відмов  $\lambda(t)$  є показником надійності об'єктів.

*17. Дайте відповідь на питання:*

Інтенсивність відмов  $\lambda(t)$  невідновлюваного ПЗ після випробування протягом 1000 годин складає  $3 \times 10^{-5}$ . Кількість працездатних примірників ПЗ після випробування складає 97 шт. Скільки примірників ПЗ відмовили при випробуванні? Як називаються такі випробування?

*18. Дайте відповідь на питання:*

Як визначити технічний ресурс ПЗ? До якої властивості надійності відноситься цей параметр?

19. Дайте відповідь на питання:

Чим відрізняється термін служби від ресурсу?

- a) Тривалістю;
- b) Розмірністю;
- c) Напрацюванням.

20. Вибрати вірні відповіді у переліку ознак у твердженні:

Причинами відмов ПЗ є:

- a) конструкційні, споживчі, експлуатаційні відмови;
- b) споживчі, експлуатаційні, інформаційні відмови;
- c) конструкційні, виробничі, експлуатаційні відмови.

21. Вибрати вірні відповіді у переліку ознак у твердженні:

a) Гамма-відсотковий ресурс є показником безвідмовності не ремонтпридатних і не відновлюваних об'єктів;

b) Гамма-відсотковий ресурс є показником довговічності не ремонтпридатних і не відновлюваних об'єктів;

c) Гамма-відсотковий ресурс є показником довговічності довільних об'єктів.

22. Оцінка надійності програми за Холстедом основана на:

a) властивості людського мозку, що з певної кількості уявних розрізень, необхідних для написання програми, частина є помилковою.

b) теорія ймовірності, що стверджує: з певної кількості розрізень, необхідних для написання програми, половина є помилковою.

c) теорія Хаосу, яка стверджує: рівні хаотичності і

впорядкованості визначаються масштабом, який може бути з ймовірністю 50% помилковим.

23. Точність прогнозу, одержуваного за допомогою моделей, що базуються на теорії Холстеда, знаходиться в межах:

- a) 5 – 10 %
- b) 10 - 12 %
- c) 11 – 15%

24. Модель Шумана відноситься до:

- a) аналітичних моделей дискретного часу
- b) динамічних моделей сталого часу
- c) динамічних моделей дискретного часу

25. При використанні моделі Шумана вважаються справедливими такі припущення:

- a) число помилок у програмі постійно і в процесі коригування нові помилки не вносяться;
- b) до початку тестування є помилки;
- c) загальна кількість машинних команд постійна в рамках етапу тестування.

26. Кластер високої готовності це:

- a) це сервер, що працює як єдина система для надання високої доступності сервісів та додатків для клієнтів;
- b) група серверів, що працюють як єдина система для надання високої доступності сервісів та додатків для клієнтів;
- c) група серверів, що працюють окремо для надання високої доступності сервісів та додатків для клієнтів.

27. Сучасні методи кластеризації використовують:

- a) асиметричну кластеризацію;

- b) симетричну кластеризацію;
- c) кластеризацію  $N + 1$ ;
- d) спільну кластеризацію.

28. *Оберіть вірне твердження:*

a) Concurrency – один з кращих підходів до симуляції навантаження на web-застосунки, оскільки дозволяє ефективно імітувати використання великою кількістю користувачів;

b) Concurrency не є абсолютно точним через те що у реальному житті кожен окремий користувач не робить велику кількість запитів у дуже малий проміжок часу;

c) Concurrency не є абсолютно точним через те що не враховує ймовірність розподілу запитів за Гаусовим законом.

29. *Бізнес-застосунки використовують правило 3-х секунд, згідно з яким:*

a) користувач не повинен чекати на завантаження ресурсу більше ніж 10 секунд;

b) користувач не повинен чекати на завантаження ресурсу більше ніж 3 секунди;

c) користувач не повинен чекати на завантаження ресурсу.

30. *Apache Benchmark це:*

a) інструмент для вимірювання швидкодії та проведення тестування на навантаження для HTTP-застосунків (web-застосунків);

b) інструмент для вимірювання швидкодії тестування на навантаження для HTTP-застосунків (web-застосунків);

c) інструмент для вимірювання навантаження для HTTP-

застосунків (web-застосунків).

*31. Docker це:*

- a) інструментарій для управління ізольованими Linux-середовищами;
- b) інструментарій для підготовки відтворюваного середовища, яке можна перевіряти на швидкодію та відмови;
- c) інструментарій для управління ізольованими Linux-контейнерами.