ПЕРЕЛІК ОСНОВНИХ УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ І СКОРОЧЕНЬ

БМ	Байєсівська мережа
ВВ	Випадкова величина
ЖЦ	Життєвий цикл
МЗНП	Модель зростання надійності пуассонівського типу
ММП	Метод максимальної правдоподібності
ПЗ	Програмний засіб (або програмне забезпечення системи)
ПЗН	Програма забезпечення надійності
ПК	Програмний комплекс
ПрЗ	Програмне застосування
ПП	Програмний продукт
ПС	Програмна система
ТІВ	Таблиця ймовірності значень випадкової величини у вершині
ТМ	Програмно-технологічний модуль
УОФ	Умовна одиниця функціональності

АНОТАЦІЯ

Тема дипломної роботи: “ Дослідження моделей та методів підтримки прийняття рішень при прогнозуванні якості програмних систем на ранніх стадіях життєвого циклу ”. // Дипломна робота // Любий Андрій Вікторович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем та програмної інженерії, група СІм-61 // Тернопіль, 2015 // с. – 127 , рис. – 41 , табл. –30, аркушів А1 – 8 , додат. –2 , бібліогр. – 54 .

Ключові слова: ЯКІСТЬ, ПРОГНОЗУВАННЯ, ПРОГРАМНА СИСТЕМА, НАДІЙНІСТЬ, ПРИЙНЯТТЯ РІШЕНЬ.

У першому розділі дипломної роботи виконано постановку задач дослідження. Визначено, що пріоритетною характеристикою якості ПС є надійність (а саме, її підхарактеристика завершеність). Проаналізовано причинно-наслідкові зв’язки відмов ПС, дефектів у робочих продуктах, помилок розробників та вад застосовуваних процесів. Визначено переваги та недоліки існуючих підходів до встановлення цільових вимог до надійності окремих компонентів ПС. Проаналізовано особливості прогнозування якості програмних систем.

У другому розділі дипломної роботи наведено опис обгрунтованих моделей та розробленого методу, призначених для вирішення задач інженерії якості на ранніх стадіях ЖЦ. Запропоновано метод раннього прогнозування надійності програмних застосувань (ПрЗ), що базується на модифікації експоненційної моделі Дж. Муси, яка, на відміну від інших моделей зростання надійності, не потребує даних про відмови і, саме тому, придатна для прогнозування надійності ще до початку системного тестування, за характеристиками ПрЗ, умов його розроблення та експлуатації.

У третьому розділі спроектовано архітектуру програмного засобу підтримки прогнозування якості ПС на ранніх стадіях ЖЦ, реалізовано її у вигляді окремого компоненту програмного комплексу управління якістю

програмними системами. Проведено експериментальні дослідження розробленого методу.

У четвертому розділі розглянуто питання економічної доцільності проведення НДР та впровадження програмного комплексу.

У дипломній роботі також розглянуто питання охорони праці, безпеки в надзвичайних ситуаціях та екології.

ABSTRACT

The theme of the thesis: " Research of decision-making support models and methods in predicting the quality of software in the early stages of their life cycle." //Master work // Lubyy Andriy Victorovych/ Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and software engineering, group CIm-61 // Ternopil, 2015 // p. - 127, fig. – 41, table. -30, sheets A1 - 8, Add – 2, Ref. - 54.

KEYWORDS: QUALITY FORECASTING, SOFTWARE SYSTEMS, RELIABILITY, DECISION MAKING.

In the first chapter of the thesis written statement of research tasks. Determined that the aircraft priority characteristic quality is reliability (ie completeness of pidharakterystyka). Analyzed causal relationships aircraft failures, defects in work products, errors and defects developers used processes. Advantages and disadvantages of existing approaches to target the requirements for the reliability of the individual components of the aircraft. The features of forecasting quality software systems.

In the second chapter of the thesis is a description based models and the methods intended to solve the problems of quality engineering in the early stages of life cycle. A method for the early prediction of the reliability of software applications (PrZ) based on the modified exponential model J. Musa, which, unlike other models increase reliability, does not require data denial, and it is suitable for predicting reliability before the system testing , the characteristics PrZ, conditions for its development and operation. The third part is designed architecture of the software support quality prediction of AC in the early stages of life cycle, it is implemented as a separate component of the software system quality management software systems. Experimental study developed method. The fourth part deals with the question of economic feasibility of conducting research and implementation of software. In the thesis work also deals with issues of safety, security and environmental emergencies.

ЗМІСТ

ВСТУП

Актуальність теми. Якість програмного забезпечення є запорукою його конкурентоздатності на ринку та ключовим аспектом при використанні замовником і кінцевими користувачами. Однак, для випуску якісного програмного продукту необхідно застосовувати технології забезпечення якості, починаючи з ранніх етапів життєвого циклу (ЖЦ) і впродовж усіх стадій виконання проекту.

Подолання наслідків помилок, які були допущені на ранніх стадіях ЖЦ і своєчасно не виправлені негативно впливають на час та вартість розробки проекту, оскільки вони призводять до появи і поширення дефектів у робочих продуктах, складність усунення яких з часом лише зростає. Тому запобігання дефектів і їх виявлення безпосередньо на тих стадіях ЖЦ, на яких вони були внесені, є основою забезпечення якості програмних систем. У стандартах з якості передбачено процеси гарантування якості та управління якістю, однак їх реалізація не можлива без вдосконалення процесів ЖЦ, зокрема прогнозування, верифікації та вимірювання.

Зважаючи на те, що моделі ЖЦ удосконалюються з акцентом на пришвидшення процесу розробки програмних систем (ітераційні моделі, технологія Agile та ін.) це вимагає розробки і впровадження у практику забезпечення якості програмних систем методів і засобів інтелектуального аналізу, зокрема методу аналізу ієрархій та апарату байєсівських мереж. Тому, задача дослідження та розробки моделей, методів і засобів прогнозування якості програмних систем на ранніх стадіях життєвого циклу і прийняття відповідних рішень для керування якістю є актуальною.

Метою роботи є дослідження моделей, методів і засобів прогнозування якості програмних систем на ранніх стадіях ЖЦ, удосконалення процесу прогнозування якості ПС для прийняття ефективних рішень з управління якістю.

Об’єктом дослідження є процеси прогнозування та забезпечення якості програмних систем на ранніх стадіях життєвого циклу.

Предметом дослідження є моделі, методи і засоби прогнозування якості програмних систем та прийняття рішень.

Задачі, які необхідно вирішити у магістерській роботі полягають у наступному:

– аналіз сучасного стану забезпечення якості програмних систем на ранніх стадіях життєвого циклу програмних систем;

– дослідження методів щодо кількісного вираження рівня якості програмного забезпечення та обґрунтування вибору моделі розподілу надійності по ієрархічній структурі ПС;

– аналіз та розробка методу прогнозування якості ПС на ранніх стадіях життєвого циклу із заданим рівнем надійності кожного компонента ПС;

– проектування та реалізація програмного комплексу підтримки запропонованого методу та обґрунтованої моделі;

Наукова новизна одержаних результатів при виконанні дипломної роботи полягає в наступному:

– уперше, розроблено метод прогнозування якості програмних систем на ранніх стадіях життєвого циклу на основі оцінювання надійності кожного компонента програмної системи, що дало змогу врахувати вплив на якість не тільки конструктивних особливостей, але й умов розробки та функціонування системи і дозволило приймати ефективні рішення щодо управління якістю ПС;

– уперше обґрунтовано модель розподілу надійності за модулями ієрархічної структури програмних систем з урахуванням їх незалежності, а також встановлених за допомогою МАІ взаємозв’язків оцінок внеску різних компонентів цієї структури в забезпечення необхідної якості ПС.

Методи дослідження. При виконання дипломної роботи магістра використовувались наступні методи:

– метод аналізу ієрархій – при обґрунтуванні моделі розподілу якості за структурою програмної системи;

– теорія ймовірностей та математичної статистики – при обґрунтуванні моделі прогнозування якості;

– теорія надійності програмних засобів і теорія прогнозування – при розробці методу прогнозування якості програмних систем на ранніх стадіях життєвого циклу

– проектування та реалізація – при проектуванні архітектури та реалізації програмного комплексу підтримки запропонованого методу.

Практична цінність результатів дослідження. Практична цінність роботи полягає у створенні архітектури та реалізації програмного комплексу підтримки методу прогнозування якості на ранніх стадіях життєвого циклу.

Публікації. Результати дослідження апробовано на VIII Всеукраїнській студентській науково - технічній конференції "Природничі та гуманітарні науки. Актуальні питання" (23-24 квітня 2015 р.) та IV Міжнародній науково-технічній конференції молодих учених та студентів «Актуальні задачі сучасних технологій» (25-26 листопада 2015 року) Тернопільського національного технічного університету імені Івана Пулюя у вигляді тез конференцій..

РОЗДІЛ 1
АНАЛІЗ ЗАДАЧ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ПРОГРАМНИХ СИСТЕМ НА РАННІХ СТАДІЯХ ЖИТТЄВОГО ЦИКЛУ

1.1. Аналіз особливостей моделювання якості програмних систем

Серед задач забезпечення якості програмних систем можна виділити три основних класи: запобігання появі дефектів у програмному продукті, виявлення та усунення дефектів безпосередньо на тих стадіях життєвого циклу, на яких вони були внесені, а також підвищення якості за рахунок тестування. Більшість досліджень в області інженерії якості присвячено саме останньому із зазначених аспектів [15-17]. В даній роботі приділена увага першим двом аспектам забезпечення якості і запропоновані моделі та методи інженерії якості, які застосовуються на ранніх етапах виконання програмних проектів.

У міжнародному стандарті ISO/IEC 12207:2002 у процесах ЖЦ представлений новий процес – “керування якістю”, мета якого – моніторинг якості за кількісними показниками. Саме його поява спричинює пошук існуючих та розроблення нових моделей та методів, які б забезпечували підтримку прийняття ефективних рішень з керування якістю. Відмінності цього процесу від процесу забезпечення (фактично, нормоконтролю) якості, представленого у національному стандарті ДСТУ 3918-99 наведені на рис. 1.1.

Сукупність яких саме властивостей продукту буде асоціюватися з поняттям його якості, тобто, якою саме буде модель якості, залежить від особливостей предметної області, призначення ПС та категорій її користувачів, а також від кінцевої мети моделювання якості з позицій того, хто його виконує. Більшість відомих автору робіт з моделювання якості, насамперед, робіт В.В. Ліпаєва, стосуються проблем забезпечення якості великих виробничих систем, що працюють у реальному часі [18-20]. В даній роботі метою моделювання якості є не лише забезпечення властивостей програмних систем, але й покращення керованості проектами за критерієм якості (тобто за основу взятий

погляд на якість менеджера проекту). Модель якості ПС представляє собою множину взаємопов'язаних характеристик, які утворюють базис для специфікації вимог до якості та її оцінювання.

Забезпечення гарантій якості (SQA, Software Quality Assurance)	**Керування якістю (SQM, Software Quality Management)**
Мета процесу: гарантувати, що продукти та процеси узгоджуються з вимогами до них і відповідають планам **Включає** два основні види діяльності: • впровадження стандартів та відповідних процедур в розроблення ПЗ • оцінку дотримання цих стандартів та процедур **Задачі виконання** – гарантувати (пересвідчитися у тому), що: • плани непротирічні та виконувані • проміжні робочі продукти узгоджуються з планами • призначені для постачання продукти відповідають вимогам • застосовні процеси узгоджуються з договором та дотримуються планів • середовище та методи розроблення узгоджуються з договором • прийняті метрики продуктів та процесів і прийоми вимірювання (за наявності) відповідають затвердженим стандартам та процедурам • штат виконавців має знання та досвід	**Новий**. Включений у ISO/IEC 12207 в 1998 році поряд з "керування проектом", "керування ризиком", "вимірювання". **Мета процесу**: здійснювати *моніторинг* якості. Гарантувати, що продукт буде задовольняти споживача. **Включає** два основні види діяльності: • визначення кількісних цілей якості, заснованих на виявлених і передбачуваних потребах користувачів • керування досягненням цілей **Задачі виконання** - гарантувати, що: • цілі якості встановлені для всіх робочих продуктів в контрольних точках • визначена стратегія досягнення цілей (метрики, критерії приймання, вимоги до процесу вимірювання) • для кожної цілі визначені і виконуються дії з - вбудування якості (надання продуктам властивостей, що забезпечують якість) - контролю якості (SQA, V&V). Якщо цілі не досягнуті – регулювання.

Рис. 1.1. Відмінності двох основних процесів в інженерії якості

Дворівневу узагальнену модель якості, рекомендовану стандартом ISO/IEC 9126:2002, подано на рис. 1.2.

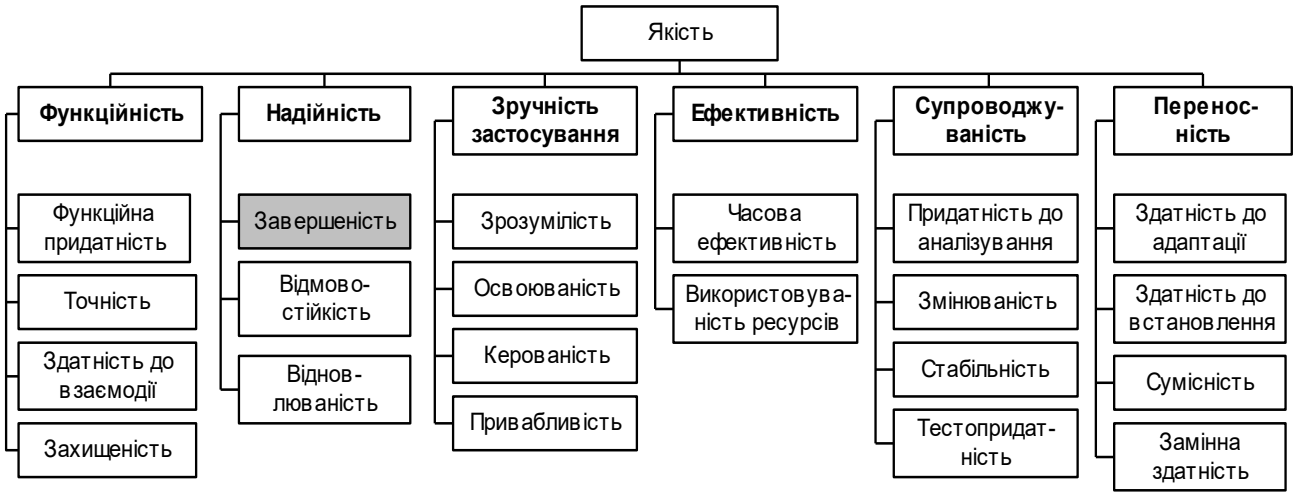


Рис. 1.2. Узагальнена модель якості ПС

Кольором виділено найбільш вагому характеристику якості саме для програмних систем оброблення даних – завершеність (безвідмовність), яка відображає властивості ПС щодо уникнення відмови через приховані дефекти.

Якість ПС вимірюється за допомогою метрик якості. За визначенням стандарту ISO/IEC 9126-2 метрика якості ПС представляє собою “модель вимірювання атрибута, що пов'язується з однією характеристикою якості ПС. Це комбінація конкретного методу вимірювання (способу отримання значень) атрибута сутності і шкали вимірювання” [21].

За видом (станом) об'єкта вимірювання (працююча версія програми або сукупність документів та програмного коду) метрики поділяються на зовнішні, внутрішні та експлуатаційні метрики ПС.

Внутрішні метрики призначені для оцінювання внутрішньої якості – множини властивостей продукту, яка визначає його здатність задовольняти встановленим або реальним потребам при використанні в певних умовах та забезпечує можливість користувачам, менеджерам та розробникам оцінювати якість проміжних і кінцевих продуктів ПС безпосередньо за їх властивостями, без виконання на комп'ютері.

Зовнішні метрики використовують виміри працюючого на комп'ютері програмного продукту, отримані в результаті вимірювання його поведінки в ході роботи. Вони призначені для оцінювання зовнішньої якості – міри у якій продукт

відповідає встановленим (заявленим) і передбачуваним вимогам під час його використання у певних умовах.

Метрики експлуатаційної якості вимірюють, до якої межі програмний продукт, встановлений і використовуваний в певному середовищі експлуатації задовольняє потреби користувачів стосовно ефективного, продуктивного та безпечного вирішення задач, а також справляє загальне позитивне враження. Вони допомагають оцінити не властивості самої ПС, а видимі результати її експлуатації – експлуатаційну якість. В.В. Ліпаєв у роботі [22] використовує інший термін для означення цього рівня якості, а саме “качество программного средства в использовании”. Характеристиками якості ПС у використанні є продуктивність, результативність, безпека функціонування та задоволеність від використання програмного продукту [21]. В даній роботі пропонується експлуатаційну якість ПС пов’язувати із загальною задоволеністю користувачів надійним функціонуванням ПС, якій відповідає характеристика якості ПС у використанні – задоволеність (“satisfaction”).

Для того, щоб ефективно керувати якістю за визначеною характеристикою або множиною характеристик, необхідно розробити таку модель якості, яка б встановлювала взаємозв’язок мір і відповідних метрик внутрішньої, зовнішньої та експлуатаційної якості ПС, тобто поєднувала погляди на якість як розробників системи, так і її безпосередніх користувачів.

У табл. 1.1 подано систематизований опис внутрішніх і зовнішніх метрик підхарактеристики завершеність, за якою моделюється якість ПС досліджуваного класу.

Таблиця 1.1

Метрики підхарактеристики завершеність

Назва метрики	**Вимірювані величини**	**Позна-чення**	**Формула та дані**	**Інтерпретація**
Внутрішні метрики «завершеності»				
Інтенсив-ність виявлення дефектів	Кількість знайдених дефектів	x_1	$F = \frac{x_1}{M(x)}$	$F \geq 0$ - висока якість перевірки $x_1=0$ не гарантує відсутності дефектів
	Очікувана кількість дефектів	$M(x)$		
Інтенсив-ність усунення дефектів	Кількість відкоригованих дефектів у робочому продукті (проект, код)	x_2	$F = x_2$	$F \geq 0$ - чим більше F, тим менше зали-шилося дефектів
	Частка (%) відкоригованих дефектів у кількості виявлених		$F = \frac{x_2}{x_1}$	$0 \leq F \leq 1$ - чим ближче до 1, тим краще
Точність перевірок	Кількість категорій дефектів, запланованих до перевірки, в плані перевірки	x_3	$F = \frac{x_3}{N(x)}$	$F \geq 0$ - чим більше F, тим вище точність перевірки
	Кількість категорій дефектів, які треба охопити перевіркою для забезпечення належного обсягу перевірки	$N(x)$		
Зовнішні метрики «завершеності»				
Оцінка щільності прихова-них дефектів	Загальна очікувана кількість дефектів у ПП	y_1	$F = \frac{\lvert y_1 - y_2 \rvert}{V}$	$F \geq 0$ - залежно від стадії перевірки. На пізніших стадіях – чим менше, тим краще
	Загальна кількість реально виявлених дефектів у ПП	y_2		
	Розмір (об’єм) ПП	V		
Щільність відмов (стосовно тестових ситуацій)	Кількість подій відмов (за період)	y_3	$F = \frac{y_3}{y_4}$	$F \geq 0$ - залежно від стадії тестування. На пізніших стадіях – чим менше, тим краще
	Кількість виконаних тестів	y_4		

Продовження таблиці 1.1

Назва метрики	**Вимірювані величини**	**Позначення**	**Формула та дані**	**Інтерпретація**
Зовнішні метрики «завершеності»				
Щільність дефектів	Загальна кількість реально виявлених дефектів у ПП	y_2	$F = \frac{y_2}{V}$	$F \geq 0$ - залежно від стадії тестування. На пізніших стадіях – чим менше, тим краще
	Розмір ПП	V		
	Кількість подій відмов	y_3		
Інтенсивність усунення дефектів	Кількість відкоригованих дефектів у ПП при тестуванні	y_6	$F = \frac{y_6}{y_2}$ $F = \frac{y_6}{y_1}$	$0 \leq F \leq 1$ - чим ближче до 1, тим краще, менше дефектів залишилося
	Загальна кількість реально виявлених дефектів у ПП	y_2		$F \geq 0$ - чим ближче до 1, тим краще, менше дефектів залишилося
	Загальна очікувана кількість дефектів у ПП	y_1		
Середній час між відмовами	Період оперування Сума по періодах безвідмовної роботи	T_1 T_2	$F_1 = \frac{T_1}{A}$ $F_1 = \frac{T_2}{A}$	*F1, F2 > 0* - чим довше, тим краще, довший період між відмовами
	Кількість виявлених відмов за період	A		

Як видно з табл. 1.1, загально визнаним “мірилом” завершеності ПС є внутрішні метрики – кількість (щільність) дефектів у робочих продуктах ПС, а також зовнішні метрики – очікувана кількість передбачених прихованих дефектів у ПС після випуску в експлуатацію, реальна кількість виявлених дефектів у ПС під час експлуатації, кількість усунутих (відкоригованих) дефектів та щільність дефектів. Може також безпосередньо використовуватися зовнішня міра завершеності – ймовірність того, що протягом заданого часу t експлуатації програмного застосування у визначених умовах не виникне послідовність вхідних даних у сценарії використання ПЗ, яка призведе до його відмови. Якість ПС у використанні на рівні експлуатаційної якості зазвичай вимірюється експертним шляхом.

На початку розроблення ПС специфікуються значення (область значень) зовнішніх метрик, які слугуватимуть критерієм досягнення рівня якості при випробовуванні ПС, а потім визначаються найбільш придатні (з позиції прогнозу значень зовнішніх метрик) внутрішні метрики і планується поетапне досягнення зовнішніх вимог до якості. Очевидно, що зовнішні вимоги до якості ПС треба встановлювати з позицій забезпечення максимально можливої експлуатаційної якості програмного продукту.

В даній роботі пропонується трирівнева модель якості ПС оброблення даних, яка забезпечує взаємозв'язок мір завершеності (рис. 1.3).

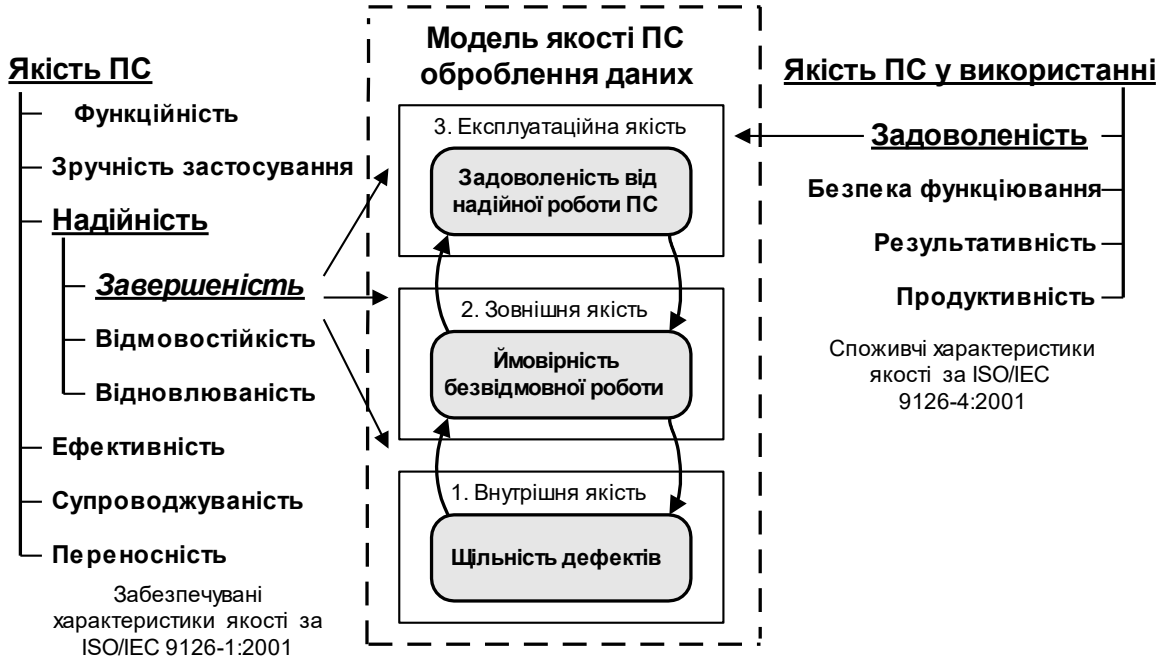


Рис. 1.3. Трирівнева модель якості ПС оброблення даних

При виборі метрик для моделі якості треба враховувати особливості моделей ЖЦ, оскільки вони впливають на вибір вимірюваних робочих продуктів, процесів, ресурсів, а також на підхід до керування якістю ПС.

Довгий час моделі якості будувалися на основі каскадної (водоспадної) моделі ЖЦ, як, наприклад, у В.В. Ліпаєва [22], О.Ф. Кулакова [23], що було виправдане умовами розроблення ПС, які існували. Але сучасна практика та

досвід роботи у проектах з динамічними ЖЦ ПС за моделлю “Обдумування – Взаємодія – Навчання” свідчать про такі їх особливості:

– кожне ПЗ має власний ЖЦ, у якому чітко виділяються дві стадії – розроблення (від аналізу вимог до автономного тестування) та системне тестування (разом з усіма подальшими стадіями за каскадною моделлю);

– стадія розроблення включає кілька послідовних етапів робіт, кожний з яких представляє собою цикл “розроблення-перевірка-контроль стану ПЗ”. Дефекти, які залишилися у ПЗ по завершенні одного етапу розроблення, переносяться на наступний;

– кожне ПЗ розробляється окремо невеликою групою учасників, яка включає проектувальника, програміста та призначеного верифікатора (або тестувальника-верифікатора);

– тривалість одного етапу встановлюється у планах проекту. Контроль стану ПЗ та прогнозування його якості виконується по завершенні кожного етапу у контрольних точках проекту.

Модель динамічного ЖЦ подано на рис. 1.4.

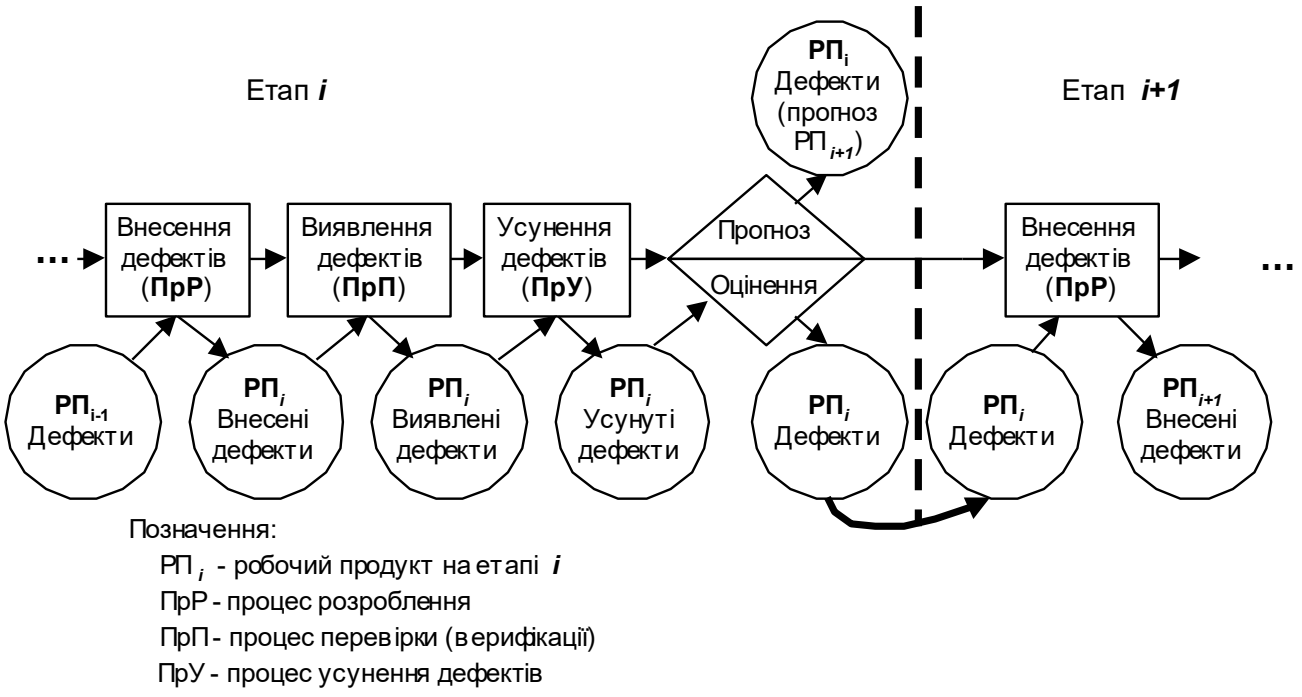


Рис. 1.4. Модель динамічного ЖЦ розроблення ПЗ

1.2. Причинно-наслідкові зв’язки чинників якості

На жаль, прийоми потужної теорії надійності технічних засобів (ТЗ) практично не застосовуються для моделювання завершеності ПЗ, з огляду на іншу природу і механізм їх відмов. На відміну від ТЗ, програмний засіб є інтелектуальним продуктом зі складними взаємозв’язками компонентів. Він не має фізичного зносу і процес його відмов цілком залежить від вмісту дефектів і операційного профілю експлуатації. Інтенсивність відмов ПЗ спадає по мірі усунення дефектів, досягаючи стабільного рівня приблизно через чотири роки після введення в дію (рис 1.5 а та б).

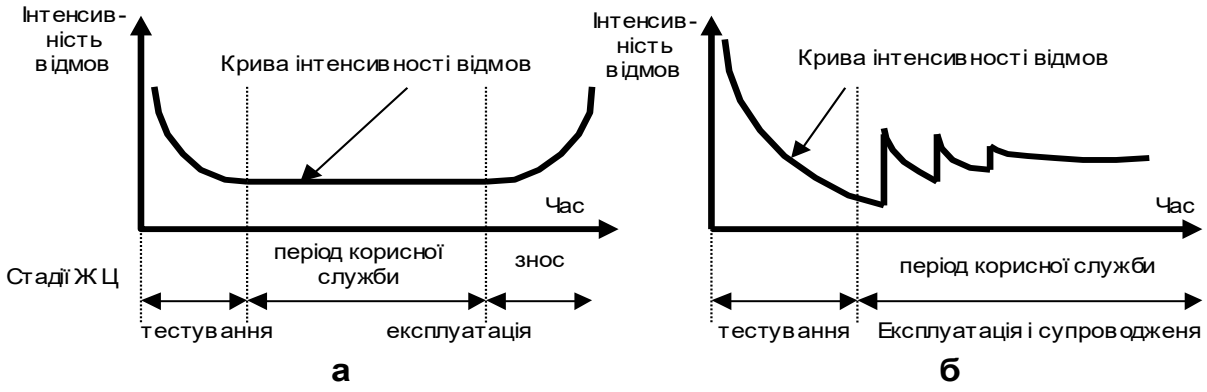


Рис. 1.5. Графіки інтенсивності відмов технічних засобів та ПЗ

Дефекти у ПЗ завжди пов’язані з помилками розробників, які припускаються на кожній стадії розроблення ПЗ. На рис. 1.6 показана поведінка кривої дефектів у ПЗ, а також діаграма їх розподілу у розрізі стадій ЖЦ (за каскадною моделлю).

Відмова – подія переходу ПЗ з працездатного стану в непрацездатний або отримання результатів, які знаходяться поза областю допустимих значень. Відмови ПЗ можуть бути обумовлені як зовнішніми причинами (помилками елементів середовища функціонування, в тому числі людини-оператора), так і внутрішніми причинами – дефектами в ПЗ.

Дефект у ПЗ – запис (текстовий фрагмент) елемента програми (коду) або документа (робочого продукту), використання якого може призвести до неправильної інтерпретації цього елемента комп'ютером (помилковому стану програми) або людиною.

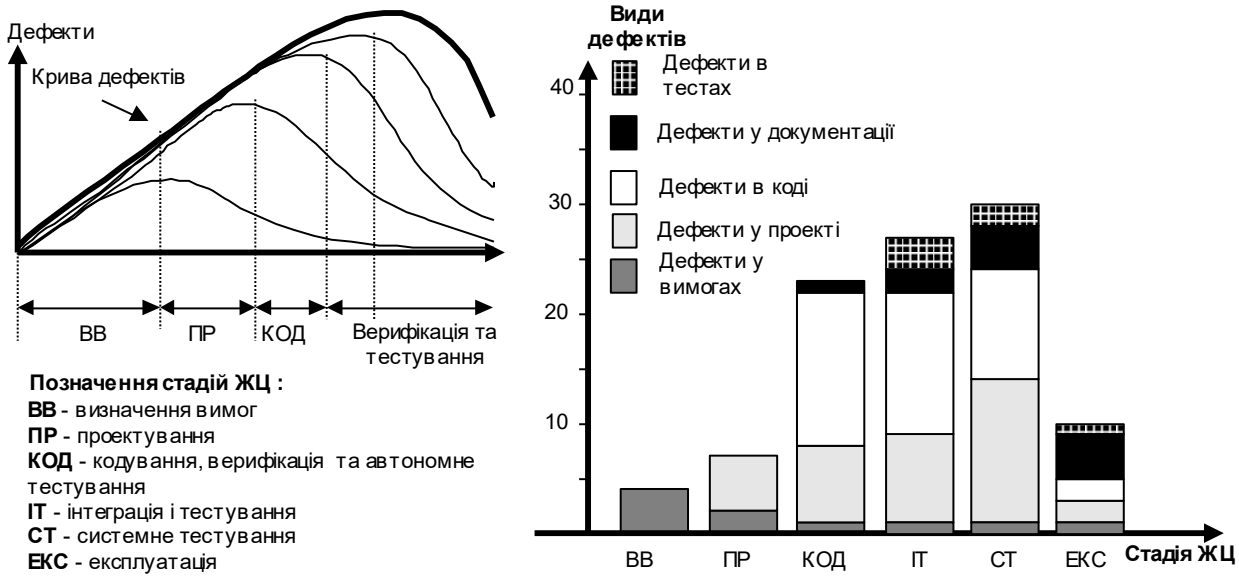


Рис. 1.6. Поведінка кривої дефектів у ПЗ

На рис. 1.7. визначені причинно-наслідкові зв'язки між відмовами ПЗ в експлуатації, дефектами у ньому, помилками розробників і вадами в процесах створення ПЗ.

Дефект завжди є наслідком помилки виконавця процесу на будь-якому етапі розроблення ПЗ. Дефекти властиві для специфікації вимог, проектним документам, текстам коду, експлуатаційній документації тощо. Вихідні робочі продукти одних процесів, які містять не усунені при перевірці дефекти, служать вхідними для інших процесів, а дефекти в них – джерелом помилок виконавців цих процесів. Крім того, помилки виконавців можуть бути спричинені вадами у визначенні процесів (неправильна послідовність дій, неправильно вибраний інструмент тощо), які сприяють неправильній інтерпретації початкової інформації людиною і прийняттю невірних рішень, або просто недостатньою професійною зрілістю виконавців процесів. Помилки виконавців, в свою чергу,

призводять до дефекту в робочому продукті, і цикл «помилка – дефект – помилка» повторюється.

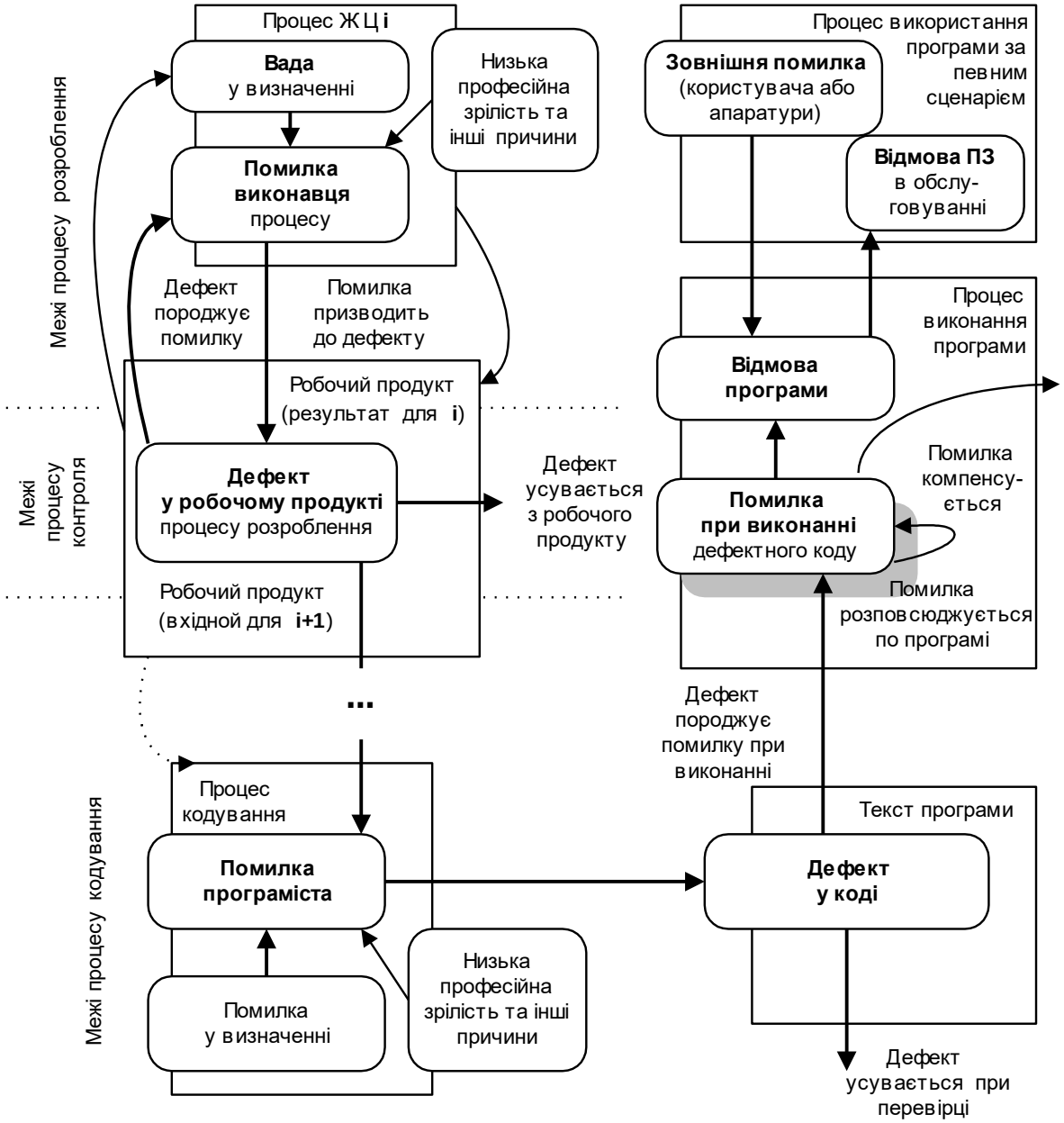


Рис. 1.7. Зв’язок помилок, дефектів та відмов ПЗ

Дефекти в коді, не виявлені під час перевірок тексту коду, є джерелом потенційних помилок і відмов ПЗ. «Спрацює» дефект чи ні, залежить від того, за яким сценарієм працюватиме з програмою користувач, і - чи «зіткнеться» оброблювач коду з неправильним елементом. Якщо дефект «спрацьовує» – виникає ланцюг помилок, що передаються від модуля до модуля. Якщо помилка

не компенсується в програмі (завдяки вбудованим у неї засобам відмовостійкості або внаслідок випадкового «спрацювання» іншого дефекту), – вона може призвести до відмови ПЗ.

Дослідження процесів внесення помилок, усунення дефектів і запобігання відмов – задачі окремого розділу інженерії якості, який безпосередньо пов’язаний із забезпеченням завершеності (безвідмовності) ПЗ систем, – інженерії надійності ПЗ (ІНПЗ).

Головна мета інженерії надійності ПЗ полягає у забезпеченні заданого рівня надійності ПЗ шляхом вироблення і застосування комплексного підходу до проблеми, що враховує особливості ПЗ, передбачуваного середовища його функціонування, а також середовища розроблення. В роботі [25] сформульовані задачі, поетапне вирішення яких сприятиме досягненню мети ІНПЗ. Основними з них є такі:

– визначення чинників, що впливають на надійність і обґрунтування цільових вимог до надійності ПЗ на певному етапі його життєвого циклу;

– визначення стратегії і застосування методів забезпечення досяжності цих вимог на кожному етапі робіт за проектом ПЗ;

– визначення об'єктів, що підлягають перевірці, і забезпечення вимірності робочих продуктів, процесів та ресурсів проекту з метою їх вдосконалення.

Задачі інженерії якості та надійності в програмному проекті вирішуються групою якості проекту (разом з групою верифікації та тестування) за відповідними планами. Об'єм вирішуваних задач може визначатися вимогами договорів на розроблення ПЗ або встановлюватися в Програмі забезпечення надійності [26].

Успіх рішення задач ІНПЗ залежить не тільки від забезпеченості проекту ресурсами, але і від рівня професійної зрілості кадрів, досконалості (потужності) окремих процесів ЖЦ і технологічної зрілості організації-розробника ПЗ. Відповідні огляди сучасних підходів до вирішення проблеми підвищення та оцінки зрілості зроблені у роботі [27].

Задачі інженерії якості ПЗ:

– встановлення обґрунтованого цільового значення завершеності для кожного програмного засобу у складі ПС;

– прогнозування досяжності встановленого рівня якості на ранніх стадіях розроблення ПС;

– визначення стратегії підвищення якості ПС з урахуванням ресурсів проекту.

1.3. Аналіз підходів до встановлення кількісних вимог до якості

Існує чимало моделей, призначених для оцінювання завершеності ПС на різних стадіях ЖЦ. Але не так багато моделей, які б допомагали замовникам та розробникам встановлювати обґрунтовані кількісні показники завершеності кожного програмного застосування системи на початку розроблення. Очевидно, що залежно від класу системи (критична / не критична), частоти застосування окремих компонентів користувачами різних категорій у ділових процесах, наслідків відмов окремих компонентів ПС для користувачів, а також інших чинників (вартості розроблення, ціни втрат тощо), вимоги до завершеності різні.

У табл. 1.2 наведено стислий огляд підходів до встановлення кількісних вимог до завершеності ПС. Описано критерії їх розподілу по компонентах ПС, переваги та недоліки підходів (моделей). Подано посилання на джерела інформації стосовно описаних підходів. Оскільки в спеціалізованій літературі традиційно досліджується надійність, а не її підхарактеристика – завершеність.

Таблиця 1.2

Підходи до розподілу кількісних вимог до надійності ПС

Назва (автор) підходу	Сутність підходу (чинники) розподілу надійності	Переваги та недоліки
Не критеріальний	Призначення однакового цільового значення надійності всім компонентам (модулям) (без розподілу)	Встановлює необґрунтовані вимоги до надійності. Якщо вимоги надто високі, вони можуть бути не досяжними або надто ресурсоємними
Технічні підходи		
Ахмед Абд-Аллах [28]	Побудова блок-схеми надійності системи з апаратно-програмними компонентами	Успадкоємлені з теорії надійності технічних систем. Прийнятні лише для вбудованого ПЗ та за умови наявності розрахункових значень характеристик надійності для всіх складових системи
А. Нейфелдер [29]	Рівно пропорційне розподілення з урахуванням послідовної/ паралельної роботи модулів	
Підходи на основі специфікації використання програмних застосувань		
Р.Чеінг [31]	Розподілення з урахуванням ймовірностей міжмодульних переходів у сценаріях роботи ПЗ	Історично перша робота (1980 рік), яка дала поштовх до розвинення підходу
Дж.Муса [32]	Розподілення за функціональним або операційним профілем. Використовує ієрархічну будову (дерево) для розподілення характеристик використання системи у відповідності до категорій користувачів, режимів системи, функцій та операцій.	Переваги – можливість урахування особливостей використання велико масштабних систем. Побудова сценаріїв тестування з максимальним урахуванням характеристики середовища експлуатації ПС, економія ресурсів тестування. Недоліки – чутливість моделей до зміни операційного профілю
Дж.Мунсон [33]	Розподілення на основі виконав-чого профілю ПрЗ. Використовує моделювання функцій ПрЗ стохастичним процесом запуску та відпрацювання модулів, які їх реалізують.	Аналіз переходів між модулями дозволяє побудувати шаблони виконання кожної функції ПрЗ. Недоліки – потребує вбудовування додаткового коду в модулі для оцінювання частоти звернення.
Дж. Рейгопал [34]	Розподілення на основі марківських моделей передачі керування між модулями ПрЗ	Враховує погляд розробників на можливі сценарії роботи системи. Недоліки – складність побудови моделей для великих систем.
Дж. Вайтеккер [35]	Розподілення на основі марківських моделей переходу ПрЗ із одного стану в інший.	Враховує погляд користувачів на сценарії використання системи. Недоліки – ті самі.
П. Раннісан [36]	Розподілення на основі ієрархічних моделей станів. Моделі мають ієрархічну та поведінкову складові. Поведінкові складові є ланцюгами Маркова (з дискретним часом), а ієрархічні – структурами-деревами.	Враховує погляд користувачів на сценарії використання системи. Але, оскільки модель є композицією марківських моделей, загальний простір станів системи не визначається. Недоліки – ті самі.

Продовження таблиці 1.2

Назва (автор) підходу	Сутність підходу (чинники) розподілу надійності	Переваги та недоліки
Підходи на основі специфікації використання програмних застосувань		
Г. Хечт [37]	Розподілення за чинниками операційної критичності та режимами відмов. Ґрунтується на методі SFMEA (Software Failure Mode and Effect Analysis)	Призначений для систем, розроблюваних за допомогою UML, використовує всі переваги інструментів програмної підтримки UML для SFMEA.
Розподілення з урахуванням властивостей модулів		
Н.Олссон [38]	Розподілення на основі оцінювання складності модулів та їх схильності до помилок	Недоліки – не враховує погляд користувачів на сценарії використання системи.
Розподілення з урахуванням ресурсів проекту		
М. Ліу [39]	Розподілення з урахуванням витрат часу на тестування (мінімізація часу тестування - критерій оптимізації). Ставиться задача нелінійної оптимізації, яка потребує застосування спеціалі-зованих математичних пакетів.	Придатна для комплексного розподілення встановленого цільового значення інтенсивності відмов для ПС з багатьма ПрЗ. Недолік – не враховує погляд користувачів на важливість компонентів ПС.
М. Хелендер [40]	Розподілення надійності на основі операційного профілю ПС з урахуванням витрат на досягнення цілей надійності. Ставиться задача нелінійної оптимізації, яка потребує застосування спеціалізованих математичних пакетів.	Поєднує два підходи до планування надійності та вартості розробки: мінімізація вартості за умови обмеження надійності та максимізація надійності за умови обмеження бюджету. Недолік – чутливість моделі до зміни операційного профілю

На основі аналізу переваг та недоліків розглянутих підходів сформульовані такі вимоги до моделі розподілу надійності по компонентах програмних систем досліджуваного класу:

– ґрунтування на погляді користувачів ПС на важливість (частоту) виконання окремих функцій у загальному діловому процесі організації;

– врахування думки розробників ПС стосовно важливості окремих програмних застосувань для виконання функцій ПС;

– дотримання диференційованого підходу до компонентів (модулів) ПС із урахуванням інтенсивності звернення до них у операційних сценаріях та збалансованості цільових значень надійності компонентів;

– забезпечення максимального рівня загальної задоволеності замовника постаченим програмним продуктом через досягнення встановлених цільових значень надійності компонентів ПС, адекватних потребам користувачів.

Модель за традицією названа моделлю розподілу надійності, хоча більш правильною була б назва – модель вимог до надійності компонентів ПС.

1.4. Аналіз підходів до прогнозування якості програмних систем

У літературі термін прогнозування якості у контексті надійності (reliability prediction) пов’язують, як правило, з її оцінюванням за моделями зростання надійності на прикінцевих етапах робіт за проектом.

У роботі [41] запропоновано розрізняти дві фази прогнозування надійності: «раннє» прогнозування (до початку тестування) і традиційне «пізнє» прогнозування.

Раннє прогнозування надійності ПС полягає у побудові проекції значень вимірів надійності, отриманих за внутрішніми метриками на певній стадії ЖЦ, на значення вимірів надійності, обчислюваних за зовнішніми метриками на будь-якій наступній стадії та в кінці розроблення ПС.

Для раннього прогнозування використовуються дані про робочі продукти, процеси та ресурси проекту ПС, які збираються та оброблюються в ході процесу вимірювання. Їх систематизований перелік подано у [42], де процес вимірювання визначений як центральний процес у сучасній парадигмі якості ПС. На основі зібраних даних будуються прогнозні моделі дефектів, які використовуються протягом усього ЖЦ ПС.

Результат раннього прогнозування надійності ПС – значення інтенсивності відмов ПС на початку (або в будь-який момент) тестування системи. Це мінімальна надійність, яка згодом може тільки підвищуватися завдяки усуненню дефектів у ПС. Мета раннього прогнозування надійності ПС – визначити, які удосконалення можуть бути зроблені в застосовуваних методах і процесах

розроблення ПС, щоб забезпечити мінімальну щільність дефектів (тобто максимальну надійність) до моменту початку системного тестування.

«Пізнє» прогнозування надійності ПС пов'язується із застосуванням аналітичних моделей зростання надійності (МЗНП) на стадії системного тестування після збору певної кількості даних про відмови ПС. Всі відомі моделі зростання надійності ПС фактично охоплюють період після раннього прогнозування, коли надійність підвищується внаслідок тестування і виправлення дефектів [43]. Якщо результати прогнозування за цими моделями показують, що надійність ПС низька, єдина можливість її підвищення - продовження тестування і усунення дефектів протягом розрахованого періоду часу. Таке “відкладене” прогнозування надійності звичайно може зменшити ризик відмови ПС у експлуатації, але не зменшить ризику проекту.

Для своєчасного керування ризиком бажано з самого початку виконання проекту оцінювати можливість досягнення заданого рівня надійності ПС і, за необхідності, регулювати процеси і ресурси розроблення.

Раннє прогнозування надійності базується на аналізі та урахуванні чинників надійності в ході ЖЦ. Оскільки, надійність ПС – одна з основних характеристик його якості, всі чинники, що впливають на якість ПС загалом, визначені, наприклад, у [42], впливають і на його надійність. Багато з них є, в свою чергу, елементами таксономії ризику проектів ПС і ризику відмов ПС, яка подана в [44].

Н. Фентон та його колеги піддали обґрунтованій критиці більшість лінійних регресійних моделей прогнозування кількості (щільності) дефектів у ПЗ за окремими чинниками (наприклад, складністю та розміром модулів) [45], а також перевірили справедливість гіпотез стосовно прогнозуючої здатності цих чинників та існуючих причинно-наслідкових залежностей дефектів та відмов у ПЗ [46]. Результати їх досліджень подано у табл. 1.3.

Таблиця 1.3

Результати перевірки гіпотез про вплив окремих чинників

№ п.п.	Формулювання гіпотези	Підтвердженість
1.а	Більшість дефектів, які виявляються під час завершальних випробувань, міститься у малій кількості модулів	Так. Підтверджується правило 20:60
1.б	Справедливість гіпотези 1.а пояснюється тим, що "відповідальні" за відмови модулі за розміром складають левову частку об'єму коду, що постачається	Ні
2.а	Невелика кількість модулів містить більшість дефектів, які проявляються під час експлуатації ПС	Так. Підтверджується правило 20:80
2.б	Справедливість гіпотези 2.а пояснюється тим, що "відповідальні" за відмови модулі за розміром складають левову частку об'єму постачуваного коду	Ні
3.	Чим більше дефектів проявляється під час функціонального тестування, тим більше дефектів проявлятиметься під час системного тестування	Незначна підтримка
4.	Чим більше дефектів проявляється під час завершальних випробувань, тим більше дефектів проявлятиметься під час експлуатації	Ні. Строге відхилення
5.а	Модулі, менші за розміром, здатні спричинити менше відмов, ніж більші за розміром	Ні
5.б	Метрики розміру (наприклад, кількість рядків коду) є добрими провісниками кількості прихованих дефектів у модулі перед випуском ПС	Незначна підтримка
5.в	Метрики розміру (наприклад, кількість рядків коду) є добрими провісниками кількості прихованих дефектів у модулі після випуску в експлуатацію	Ні
6.	Метрики структурної складності є кращими, ніж метрики розміру, провісниками того, що модулі матимуть більше дефектів або спричинюватимуть більше відмов	Ні
7.	Щільність дефектів на відповідних стадіях функціонального, системного, інтеграційного тестування та експлуатації кожної версії ПС залишається постійною (поведінка кривої щільності дефектів однакова)	Так
8.	ПС, які розробляються в подібних середовищах, мають подібну щільність дефектів на відповідних стадіях тестування та в експлуатації	Так

Аналізуючи представлені результати, можна зробити висновок, що, хоча правило Парето (гіпотези 1.а та 2.а) підтверджується, гіпотези про існуючий тісний причинно-наслідковий зв'язок між дефектами (відмовами) та розміром (складністю) модулів практично не мають підтримки. В той же час, підтверджуються гіпотези про те, що щільність дефектів у різних версіях ПС, що

постачаються організацією-розробником, є незмінною. Це свідчить про залежність щільності дефектів від досконалості середовища (процесів) її розроблення та тестування, а також залежність кількості відмов від умов експлуатації ПС.

Тому пошук моделей, придатних для раннього прогнозування надійності, обмежився класом мультиплікативних моделей, які ґрунтуються, перш за все, на чинниках середовища розроблення та функціонування ПС.

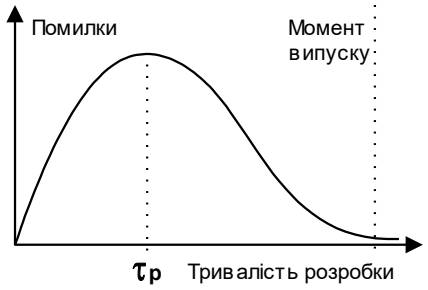
Огляд проаналізованих моделей раннього прогнозування надійності зроблено у роботах [45, 47], а їх стислий опис подано у табл. 1.4. Для кожної моделі вказано чинники надійності, прогнозовані характеристики, основні рівняння моделі, а також стадії проекту, на яких виконується збір даних для прогнозування. У п. 4 таблиці є посилання на оцінки розміру ПЗ в умовних одиницях функціональності (УОФ) (або FP, від functional points). Огляд методів визначення розміру за відповідною методологією аналізу функціональної складності ПЗ - FPA (functional points analysis) – подано у роботі [48].

Таблиця 1.4

Характеристики моделей раннього прогнозування надійності

Назва моделі	Чинники для прогнозування надійності	Прогнозована метрика та основні рівняння моделі
1. Модель RLM (розробник Air Force Rome Laboratory) Дані збираються на стадіях від визначення концепції до кодування	• Z – тип застосування • Y – середовище розробки • D_1 – вимоги до відмовостійкості • D_2 – вимоги до трасовуваності рішень • D_3 – вимоги до якості (SQA) • D_4 – рівень мови коду • D_5 – розмір програми • D_6 – структурованість ПЗ • D_7 – об’єм повторно використовуваного коду • D_8 – складність ПЗ • D_9 – відповідність стандартам	**Щільність дефектів** (D_0)*:* $D_0 = Z \cdot Y \cdot \prod_{i=1}^{9} D_i$ **Початкова інтенсивність відмов** (λ_0) $\lambda_0 = \varphi \cdot K \cdot (D_0 \cdot KSLOC)$ φ – лінійна частота виконання коду (середня інтенсивність виконання, поділена на об’єм виконуваного коду) K – коефіцієнт виявлення дефектів: $1.4 \cdot 10^{-7} \le K \le 10.6 \cdot 10^{-7}$ KSLOC – одиниця розміру ПЗ (тисяча рядків початкового коду)

Продовження табл. 1.4

Назва моделі	Чинники для прогнозування надійності	Прогнозована метрика та основні рівняння моделі
2. Модель Гефні та Девіса Дані збираються на стадіях від стадії визначення вимог до випуску	• E – інтенсивність внесення дефектів • Стадія ЖЦ • Розмір ПЗ (у SLOC) • Інтенсивність виявлення дефектів за кривою Релея:  τ_p – момент часу виявлення дефекту, який відповідає піку на кривій Релея, де має бути виявлено 39% дефектів	**Щільність виявлених дефектів** на певній стадії ЖЦ: $\Delta D_g = E \cdot [\exp(-B \cdot (g-1)^2) - \exp(-B \cdot g^2)]$ g – коефіцієнт виявлення дефектів в залежності від стадії (g = 1 - вимоги, g = 2 - проектування, g = 3 - кодуван-ня, g = 4 - автономне тестування, g = 5 - приймальне тестування. $B = 1/2\tau_p^2$ Кількість дефектів, які залишилися не виявленими на стадії t: $D_g = E \cdot \exp(-B \cdot g^2) \cdot SLOC_g$ $SLOC_g$ – кількість рядків вихідного коду, обчислена на стадії g
3. Модель Малаї-Дентона [49] Збір даних від стадії кодування до випуску	D_i – модифікатори щільності • D_1 - етап тестування ПС (автономне, системне тощо) • D_2 - зрілість групи програмістів (кваліфікація, досвід) • D_3 – зрілість процесу розроблення (за CMM) • D_4 – структура розроблюваної ПС (мова, складність, модульність, частка повторно використовуваного коду) • D_5 – плинність вимог	**Щільність дефектів** (D_0)*:* $D_0 = C \cdot \prod_{i=1}^{5} D_i$ C – константа пропорційності для калібрування моделі (кількість дефектів у KSLOC), $6 \le C \le 20$ Модифікатор $D_4 = 1+0.4a$ a – частка коду на мові асемблера (передбачається, що код на асемблері містить на 40% більше дефектів)

Продовження табл. 1.4

Назва моделі	Чинники для прогнозування надійності	Прогнозована метрика та основні рівняння моделі
4. Модель COCUAL-MO (Боема-Чулані [50]**)** Є розширенням моделі COCOMO II для визначення витрат. Збір даних від стадії визначення концепції до випуску	Складається з двох підмоделей: моделі внесення дефектів та моделі виявлення дефектів. Параметри моделі: • I_s – розмір ПС (у KSLOC або УОФ) • A_j – базове значення щільності дефектів для артефакту j (j=1 – для вимог, j=2 – для проекту, j=3 – для коду) • D_{ij} – модифікатор щільності дефектів, пов'язаних з чинником i (i=1, 2,…,21) для артефакту j (j = *1,2,3*) за категоріями – платформа, продукт, персонал, проект (зміст чинників описаний у роботі [2, глава 8]). • M_{ij} – частка дефектів артефакту j, яка видаляється завдяки застосуванню методології i (i=*1,2,3*). Класифікація методологій виявлення дефектів: - автоматизований аналіз та формальні методи (від простого синтаксичного контролю до верифікації коду); - колективні перегляди (від неформальних методів перевірки до формальних інспекцій); - тестування (від "повністю відсутнього" до керованого процесу).	**Кількість нетривіальних дефектів вимог, проекту, коду (N$_j$)** де j = 1 для вимог, j = 2 для проекту, j = 3 для коду). Вираз для оцінки загальної кількості внесених дефектів: $N = \sum_{j=1}^{3} A_j \cdot I_s^{B_j} \cdot \prod_{i=1}^{21} D_{ij}$ B_i = 1 (в поточній версії моделі). Рівняння для оцінної кількості внесених дефектів у артефакті j, N_j: $N_j = A_j \cdot I_s^{B_j} \cdot \prod_{i=1}^{21} D_{ij}$ Для врахування впливу кожного чинника i на щільність дефектів у артефакті j використовується шкала значень від VH – дуже високий до VL – дуже низький. Значення, більші 1, свідчать про збільшення щільності дефектів внаслідок впливу чинника, значення, менші 1 – про його зменшення. 1 – номінальне значення. Для відображення рівня ефективності методології і стосовно виявлення дефектів артефакту j використову-ються значення від VL – дуже низька ефективність, до EH – екстра висока ефективність. Значення на інтервалі [0,1), де 0 – найнижча ефективність. Рівняння для оцінки загальної кількості дефектів, N_j, які залишилися не усуненими для артефакту j: $N_j = C_j \cdot I_{sj} \cdot \prod_i (1 - U_{ij})$ C_j – базове значення залишкової кількості дефектів j-го артефакту. U_{ij} – доля усунених дефектів j-го артефакту завдяки методології i.

Подані у табл. 1.4 моделі прогнозування щільності дефектів мають значні переваги над лінійними регресійними моделями, оскільки:

1) отримані значення прогнозованої кількості дефектів не можуть бути нульовими або від'ємними (що можливо для адитивних моделей);

2) ці моделі дозволяють врахувати одночасно значну кількість чинників надійності;

3) моделі можуть застосовуватися в разі, коли є лише неповна інформація. За умовчанням прогнозоване значення для невідомого параметра (чинника) приймається рівним одиниці, що відповідає усередненій оцінці впливу чинника;

4) кількісні характеристики артефактів проекту для цих моделей отримані шляхом аналізу великої кількості експериментальних даних (наприклад, у моделі 1 – 59 досліджених проектів, а у моделі 4 - 163).

Загальні недоліки розглянутих моделей такі:

1) кількісні дані у моделях придатні для застосування лише у такому проекті, характеристики і умови виконання якого подібні до тих, що їх мали досліджені проекти;

2) моделі потребують перекалібрування з урахуванням попереднього власного (історичного) досвіду організації-розробника, яка їх застосовує;

3) моделі орієнтовані на каскадний ЖЦ розроблення ПС та застосовуються на відповідних стадіях ЖЦ. Вони не розраховані на сучасні адаптивні методології розроблення;

4) застосування моделей потребує попереднього впровадження у проекти процесу вимірювання, а, отже, можливе лише на четвертому рівні зрілості організації-розробника ПС (за моделлю СММ). Думка про 10% покращення кожного наступного проекту у порівнянні з минулим – практично не справджується.

Використовуючи огляд переваг та недоліків існуючих підходів до раннього прогнозування кількості (щільності) дефектів у ПС робимо такі висновки:

1) для забезпечення можливості збору на накопичення історичних даних про артефакти проекту треба:

а) впровадити процеси вимірювання та оцінювання у ЖЦ проектів,

b) вибрати (розробити) методи класифікації збираних даних відповідно до дій (стадій, процесів), які можуть спричинювати дефекти;

2) для забезпечення ефективного раннього прогнозування надійності у проектах треба:

a) побудувати модель раннього прогнозування, яка б ураховувала переваги моделей, поданих у табл. 1.4, зокрема моделей 3 та 4, дозволяла виконувати прогнозування в умовах невизначеності щодо кількісних оцінок впливу чинників на надійність, а також ураховувала потреби динамічного ЖЦ, тобто не була “прив’язана” до конкретних стадій у каскадній моделі;

b) вдосконалити процеси ЖЦ, зокрема верифікації, керування проектом, керування якістю та навчання, шляхом надання методичної підтримки.

1.5. Прогнозування якості в циклі керування проектом

Результати прогнозування якості на ранніх стадіях ЖЦ ПС передусім використовуються для оцінки ресурсів проекту (зокрема, ресурсів тестування зростання надійності), необхідних для забезпечення досягнення встановленого рівня цільової характеристики якості. Оцінка ресурсів може показати можливе відхилення від графіків і вартості розроблення, що спричинить прийняття одного з трьох рішень: про перерозподіл ресурсів, вдосконалення процесів ЖЦ або перерозподіл якості (надійності).

У роботі автора [47] зроблено висновок про те, що прогнозування надійності повинно бути ітеративним процесом застосування комбінації моделей “раннього” та “пізнього” прогнозування надійності ПС з послідовним уточненням параметрів цих моделей по ходу розроблення програмного продукту.

Результатами прогнозування надійності ПС є:

– передбачена кількість дефектів, очікувана на кожній стадії ЖЦ ПС;

– передбачена інтенсивність відмов розроблюваних програмних засобів;

– пропозиції по вдосконаленню ПС і процесів її розроблення, які можуть бути надані внаслідок інтеграції процесу прогнозування в ЖЦ ПС.

Загальну процедуру прогнозування надійності схематично подано на рис. 1.8.

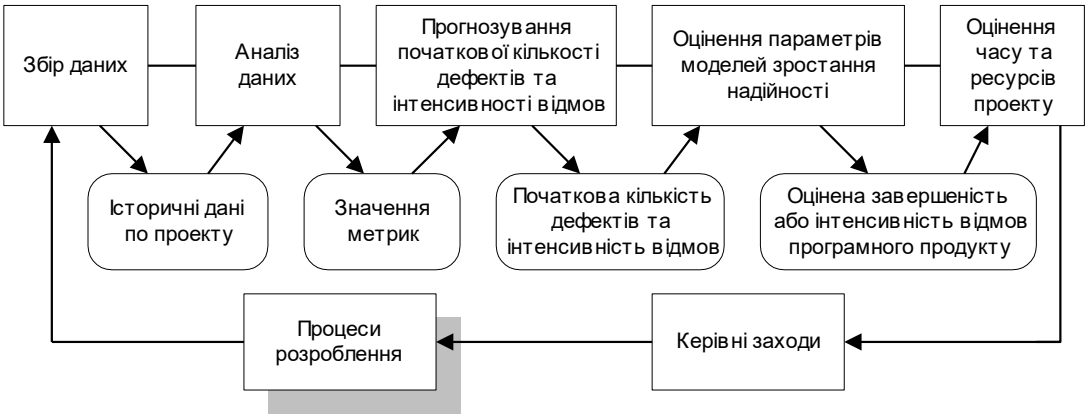


Рис. 1.8. Цикл прогнозування надійності ПС

По ходу виконання проекту значення вимірюваних величин оновлюються (уточнюються) і перед початком системного тестування прогнозні значення замінюються фактичними і використовуються для статистичного оцінювання параметрів застосовуваної моделі зростання надійності.

Оцінка кількості дефектів у ПС до початку системного тестування, отримана за моделлю раннього прогнозування, може використовуватися нарівні або замість оцінки параметрів методами максимальної правдоподібності, найменших квадратів тощо.

Переваги оцінювання параметрів моделі зростання надійності до початку тестування такі:

– початкові тестові дані, невеликі за обсягом, спричинюють нестабільні, або навіть неможливі значення параметрів моделі зростання надійності. В цих випадках значення, оцінені за допомогою апріорно отриманої

інформації, можуть використовуватися для контролю або для стабілізації прогнозів, доповнюючи інформацію, отриману за даними тестування;

– якщо для оцінювання значень параметрів використовуються ітеративні технології, значення, які отримуються з їхньою допомогою, можуть залежати від початкових оцінок, з яких розпочинаються обчислення. Використання апріорних значень як початкової оцінки забезпечує звуження пошуку наближених значень параметрів моделей;

– апріорні значення, оцінені за історичними даними, можуть використовуватися для виконання попереднього планування та розподілення ресурсів до початку тестування.

По закінченні кожної стадії розроблення ПС керівництву повинен надаватися звіт про оцінювання надійності, що відображає міру досягнення поставлених цілей, використані ресурси, виявлені недоліки у самих процесах ЖЦ та рекомендації з їх удосконалення.

Результати прогнозів надійності, які не відповідають початковим вимогам до надійності, свідчать про необхідність втручання в процес розроблення з боку керівництва.

Кінцева мета створення і застосування прогнозних моделей – організація керування проектом на кількісній основі. У сукупності з іншими прогнозними моделями, наприклад моделями прогнозу об'єму і складності ПС [48], а також вартості її розроблення на всіх стадіях ЖЦ [51], ці моделі дають керівництву проекту реальні важелі (критерії) керування ризиком проекту. Забезпечуючи баланс проекту по трьох основних чинниках – тривалість розроблення, вартість і якість (надійність) ПС, процес розроблення можна зробити дійсно керованим.

1.6. Висновки

У розділі 1 визначена пріоритетна характеристика якості ПС – надійність (а саме, її підхарактеристика «завершеність», безвідмовність).

Сформульовано загальний підхід до забезпечення якості, заснований на вирішенні трьох груп задач інженерії якості ПС на ранніх стадіях ЖЦ:

– обґрунтування цільових вимог до якості (завершеності) ПС;

– визначення стратегії прогнозування досяжності цих вимог на кожному етапі робіт за проектом ПС;

– забезпечення вимірності робочих продуктів, процесів та ресурсів проекту з метою їх вдосконалення.

Основні наукові та практичні результати, які досягнуто у даному розділі:

1. Сформульовано вимоги до моделі розподілу завершеності ПС, що належать до класу систем оброблення даних.

2. Визначено, що основною прогнозованою мірою надійності ПС на ранніх стадіях має бути кількість (щільність) дефектів.

3. Зроблено висновки про необхідність впровадження процесів вимірювання та оцінювання у проекти ПС; побудову моделі раннього прогнозування з урахуванням потреб динамічного ЖЦ (без “прив’язки” до стадій каскадної моделі ЖЦ); необхідність вдосконалення процесів ЖЦ (зокрема верифікації та керування проектом) шляхом надання методичної та інструментальної підтримки.

РОЗДІЛ 2
ОБГРУНТУВАННЯ МОДЕЛІ ТА РОЗРОБКА МЕТОДУ ПРОГНОЗУВАННЯ ЯКОСТІ ПРОГРАМНИХ СИСТЕМ НА РАННІХ СТАДІЯХ ЖИТТЄВОГО ЦИКЛУ

2.1. Модель прийняття рішень з управління якістю ПС

Процес прийняття рішень стосовно забезпечення якості розроблюваної ПС в ході керування проектом включає вирішення комплексу таких задач:

1) визначення цільового кількісного значення встановленої характеристики якості ПС (рівня якості), яке слугуватиме критерієм відповідності ПС потребам користувачів;

2) прогнозування досяжності встановленого рівня якості з урахуванням поточного стану виконуваних процесів та розроблених продуктів у програмному проекті. Пошук та аналіз можливих альтернативних рішень для досягнення рівня за умови усунення «слабких місць» у процесах ЖЦ. Якщо рівень недосяжний – його перегляд та узгодження з замовником ПС;

3) вибір найкращого рішення з позицій досягнення визначеного рівня якості та оцінювання можливості його реалізації у проекті з урахуванням обмежених ресурсів. В разі потреби – перегляд цільового значення ;

4) визначення стратегії, методів та засобів забезпечення якості компонентів ПС, а також перевірки правильності вироблених робочих продуктів ПС (документів проекту, коду тощо), які відповідають встановленому рівню якості та обмеженням ресурсів. Організація збору даних про хід виконання проекту, стан процесів, ресурсів та розроблюваних робочих продуктів ПС для накопичення досвіду;

5) аналіз досвіду, перегляд (уточнення) суджень стосовно важливості тих чи інших чинників успішного досягнення встановленого рівня якості та

коригування стану виконання проекту. В разі потреби - повторне виконання задач (п. 1 або п. 2).

У [52] запропоновано розглядати склад та послідовність визначених задач (п. 1 – п. 5) як основні функціональні вимоги до системи підтримки прийняття рішень (СППР) з питань керування процесом забезпечення якості розроблюваних ПС, модель якої подано на рис. 2.1.

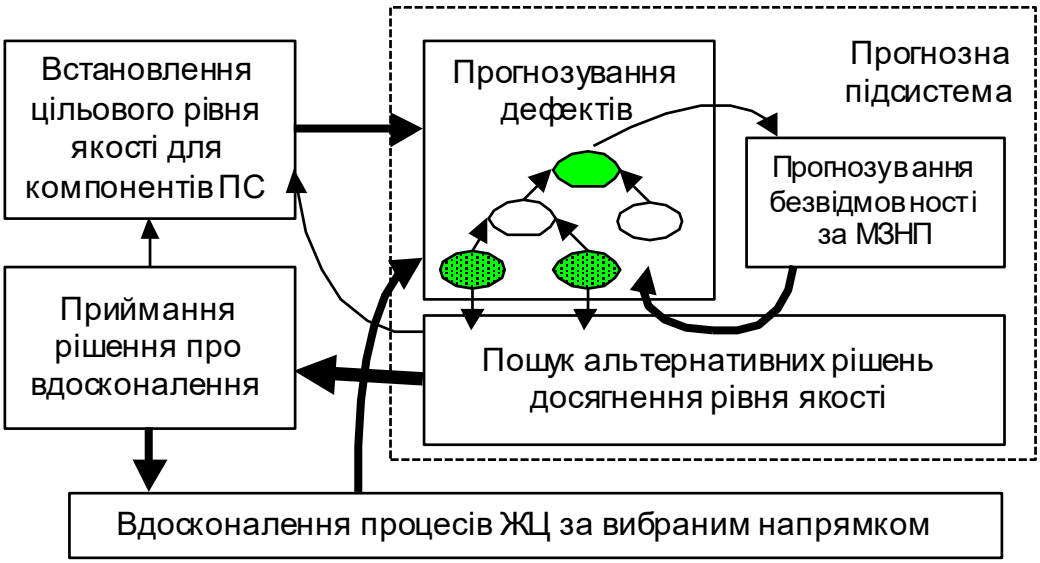


Рис. 2.1. Концептуальна модель прийняття рішень з керування якістю

Далі у розділі описано модель та метод, розроблені для підтримки вирішення поставлених задач.

2.2. Модель розподілу надійності по компонентах ПС

2.2.1. Метод аналізу ієрархічної структури ПС

З огляду на той факт, що плинність вимог до ПС є джерелом значної кількості проблем (до 40%) при її супроводженні, для підтримки актуальності схеми розподілу надійності треба з самого початку проекту враховувати погляд

користувачів на важливість надійного функціонування окремих компонентів ПС у бізнес процесі.

Для розподілення встановленого цільового значення рівня надійності за компонентами ПС, які відносяться до класу інформаційних систем, запропоновано чотирирівнева ієрархічна структура, кожний рівень якої відповідає рівню бачення проблеми якості відповідною категорією учасників проекту ПС, а саме:

– замовника, який зацікавлений у загальній якості ПС при її використанні ($Q_{пс}$) та підвищенні ефективності бізнес процесу завдяки надійній роботі впровадженої ПС;

– користувачів, які пов’язують загальну якість системи з надійним виконанням множини функцій ПС ($F_1, ..., F_k$);

– менеджерів (та аналітиків), які пов’язують надійне виконання кожної функції F_i з надійною роботою множини розроблюваних програмних застосувань ($Z_1, ..., Z_l$), призначених для автоматизованої підтримки функцій;

– проектувальників (та програмістів), які пов’язують надійність кожного програмного засобу Z_i з надійністю множини розроблюваних, а також повторно використовуваних незалежних модулів ($M_1, ..., M_m$), до яких є звернення у програмних засобів. Припущення незалежності відповідає сучасним концепціям об’єктно-орієнтованого та компонентного підходів до розроблення ПС.

Ієрархічна декомпозиція є природним засобом спрощення проблеми в системах оброблення даних, не пов'язаних з функціонуванням в реальному масштабі часу. Вона властива сучасним CASE-технологіям, які застосовуються для побудови ПС. Наприклад, Oracle Designer’2000 надає засоби для автоматизованого опису ділового процесу в організації-замовнику та побудови діаграм ієрархії функцій у цьому процесі, а також для визначення на нижньому рівні ієрархії модулів-кандидатів трьох типів – модулів-форм, модулів-звітів та модулів-меню.

Приклад ієрархічної структури ПС подано на рис. 2.2

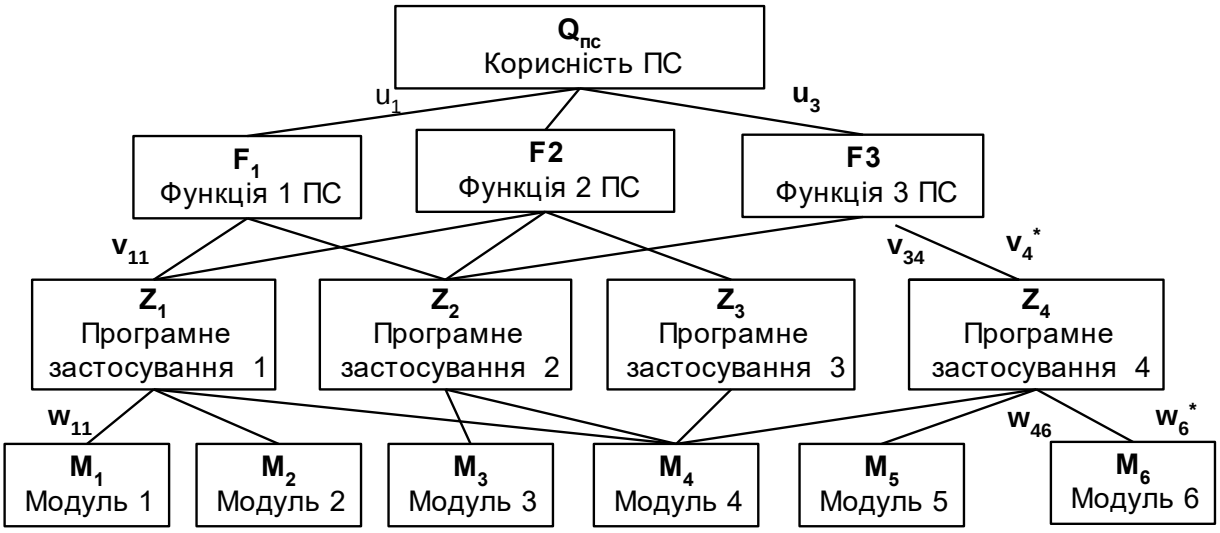


Рис. 2.2. Чотирирівнева ієрархічна структура ПС

Основна мета побудови і аналізу ієрархії ПС – отримати параметри моделі розподілу надійності на кожному її рівні з урахуванням важливості компонентів кожного з рівнів 2 – 4 для загальної якості (надійності) ПС.

Для визначення ваги окремих компонентів у ієрархії пропонується використати метод аналізу ієрархій (MAI) [54]. Загальний процес аналізу побудованих ієрархічних структур та метод їх реалізації описано у розд. 3. За цим методом визначаються:

1) вектори локальних пріоритетів функцій, програмних засобів та модулів, а саме:

$U = (u_1, u_2, ..., u_k)$ – вектор коефіцієнтів відносної ваги функцій у Q_{nc};

$V_i = (v_{i1}, v_{i2}, ..., v_{il})$, $i = 1, \ldots k$ – вектори коефіцієнтів відносної ваги програмних застосувань для кожної функції;

$W_i = (w_{i1}, w_{i2}, ..., w_{im})$, $i = 1, \ldots l$ – вектори коефіцієнтів відносної ваги модулів для кожного програмного застосування;

2) вектори загальних (глобальних) пріоритетів програмних застосувань та модулів. Загальна вага *i*-го програмного засобу розраховується за формулою

$$V^*_i = \sum_{j=1}^{k} u_j \cdot v_{ij}$$

Для всіх програмних застосувань та по відношенню до всіх функцій вектор загальних вагових коефіцієнтів визначається так:

$$V^* = U \cdot \begin{pmatrix} V_1^* \\ V_2^* \\ \dots \\ V_l^* \end{pmatrix} \quad \text{або} \quad (v_1^* \quad v_2^* \quad \dots \quad v_l^*) = (u_1 \quad u_2 \quad \dots \quad u_k) \begin{pmatrix} v_{11} & v_{12} & \dots & v_{1l} \\ v_{21} & v_{22} & \dots & v_{2l} \\ \dots & \dots & \dots & \dots \\ v_{k1} & v_{k2} & \dots & v_{kl} \end{pmatrix} \quad (2.1)$$

Так само визначається вектор загальних вагових коефіцієнтів для всіх модулів, до яких є звернення у програмних застосуваннях:

$$(w_1^* \quad w_2^* \quad \dots \quad w_m^*) = (v_1^* \quad v_2^* \quad \dots \quad v_l^*) \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \dots & \dots & \dots & \dots \\ w_{l1} & w_{l2} & \dots & w_{lm} \end{pmatrix} \quad (2.2)$$

Отримані загальні вагові коефіцієнти для множини функцій, програмних застосувань та модулів далі використовуються при побудові моделі розподілу цільових вимог до надійності. Кожний з цих вагових коефіцієнтів є оцінкою ступеню важливості надійної роботи відповідного компоненту ПС для забезпечення її загальної експлуатаційної якості за критерієм надійності.

2.2.2. Розподіл цільових вимог до надійності по компонентах ієрархії

Міру експлуатаційної якості ПС пропонується визначати як функцію корисності $Q_{nc} = \sum_{i=1}^{k} a_i \cdot R_i$, де a_i – міра важливості функції ПС для бізнес процесу користувача, R_i – надійність виконання функції у заданому періоді t експлуатації системи.

Надійність виконання функцій ПС оцінюють лише під час системного тестування та експлуатації ПС, що з позицій керування якістю надто пізно. Тому

поставлено та вирішено задачу апріорного знаходження оптимального рівня надійності кожного модуля та програмного засобу, який забезпечує максимізацію функції корисності з урахуванням технічних та ресурсних обмежень проекту ПС.

Введемо позначення:

u_i – коефіцієнт відносної ваги функції F_i у досягненні Q_{nc}, i=1,… ,k;

v_{ij} – локальний коефіцієнт відносної ваги j-го програмного засобу у забезпеченні виконання i-ої функції, i = 1,…, k; j=1,…, l;

v^*_j - загальний коефіцієнт відносної ваги програмного засобу Z_j;

q_j - надійність роботи програмного засобу Z_j у період t експлуатації;

w_{js} – локальний коефіцієнт відносної ваги s-го модуля в забезпеченні виконання j-го програмного застосування, s = 1,…, m; j=1,…, l;

w^*_s - загальний коефіцієнт відносної ваги модуля M_s;

r_s – надійність модуля M_s у період експлуатації t;

E_j – множина номерів усіх модулів, які необхідні для виконання програмногог засобу Z_j;

α_s – нижня межа надійності модуля M_s;

β_s – верхня межа надійності модуля M_s;

G – загальна ціна ПС;

C – собівартість створення ПС організацією-розробником;

c_s – накладні витрати, пов’язані з розробленням модуля M_s ;

d_s – витрати, необхідні для підвищення надійності модуля M_s на 0,01;

δ – частка прибутку у ціні ПС.

Запропонована модель розподілу надійності ґрунтується на побудованій системі відповідності вагових коефіцієнтів на різних рівнях ієрархії, тобто

$$Q_{nc} = \sum_{j=1}^{l} v^*_j \cdot q_j = \sum_{s=1}^{m} w^*_s \cdot r_s \qquad (2.3)$$

За умови незалежності модулів надійність qj кожного програмного засобу Z_j $(j = 1, ...,l)$, яке у своїй роботі використовує модулі M_s, $(s = 1,..., m)$, обчислюється за формулою

$$q_j = \prod_{n \in E_j} r_n \tag{2.4}$$

Необхідно знайти таке модульне розподілення надійності, яке максимізує функцію корисності Q_{nc}, або формально

$$Q_{nc}(r_1,...,r_m) = \sum_{j=1}^{l}(v_j^* \cdot \prod_{n \in E_j} r_n) = \sum_{j=1}^{l}(\sum_{i=1}^{k} u_i v_{ij} \cdot \prod_{n \in E_j} r_n) \to \max \tag{2.5}$$

при обмеженнях

$$0 < \alpha_s \le r_s \le \beta_s \le 1, \; s = 1, \; ..., \; m \tag{2.6}$$

$$c_s + d_s \cdot r_s \le (1-\delta) \cdot w_s^* \cdot G \;\text{ або }\; c_s + d_s \cdot r_s \le (1-\delta) \cdot G \cdot \sum_{j=1}^{l}\sum_{i=1}^{k} u_i v_{ij} w_{js} \tag{2.7}$$

$$\sum_{s=1}^{m}(c_s + d_s \cdot r_s) \le C \tag{2.8}$$

Обмеження (2.6) задають допустимі нижні α_s верхні β_s межі надійності модулів, виходячи з міркувань важливості (або ризику відмов) кожного модуля. Якщо у договорі на створення ПС встановлене кількісне значення її надійності, саме це значення приймається для β_s.

Обмеження (2.7), (2.8) є вартісними. Нерівності (2.7) встановлюють відповідність загальних витрат на розроблення модуля, з одного боку, і частки ціни системи G, яка припадає на цей модуль з огляду на його внесок в надійність системи, та з урахуванням частки прибутку δ розробника, з другого боку. Припускається лінійна залежність між вартістю модуля і його надійністю.

Нерівність (2.8) встановлює відповідність сумарних витрат на розроблення всіх модулів і собівартості створення ПС. Витрати на розроблення модуля складаються з накладних (непрямих) та прямих витрат безпосередньо на його розроблення.

Оскільки на практиці під час випробувань та експлуатації ПС оцінюється надійність програмних застосувань, відповідний розподіл надійності для q_j, $j = 1, \ldots l$ отримується з (2.4).

Розв'язок задачі дає розподіл надійності для програмних застосувань і модулів, що робить цей показник визначальним чинником керування якістю ПС. Отримані за моделлю значення надійності модулів і програмних засобів використовуються як відповідні цільові вимоги.

2.2.3. Метод побудови моделі прогнозування дефектів

Залежно від кількості чинників якості, використовуваних для прогнозування, та встановлених між ними взаємозв'язків, можуть будуватися моделі дефектів різної структури та складності. Запропонований метод побудови моделей прогнозування щільності дефектів включає такі етапи:

1) аналіз технічних обмежень інструментів, застосовних для побудови моделі, які можуть позначитися на її структурі;

2) аналіз та вибір найбільш впливових чинників щільності дефектів;

3) визначення топології моделі – байєсівської мережі (структури ациклічного графу, вершини якого асоційовані з чинниками дефектів);

4) аналіз залежностей артефактів проекту та вибір шкал вимірювання змінних у вершинах мережі;

5) побудова мережі, перевірка її правильності та відлагодження;

6) перевірка чутливості мережі на гіпотетичних сценаріях подій.

Далі розглядаються задачі, які вирішуються на кожному етапі застосування методу.

1. Урахування обмежень засобів реалізації при побудові моделі

Обмеження реалізації Байєсівської мережі (БМ) у Hugin Lite 6.5 (до 50 значень змінних у вершинах та до 500 їх комбінацій) обумовили спрощення моделі і позначилися на прийнятих рішеннях стосовно вибору чинників якості для відображення у вершинах БМ, типів вершин, шкал значень та топології БМ:

– кожна вершина має не більше двох батьківських вершин;

– кількість змінних (вершин БМ) та кількість значень кожної змінної вибирається так, щоб загальна кількість значень не перевищувала 50;

– там, де це доцільно, застосовуються закони (функції) розподілу ймовірності ВВ (замість побудови ТІВ), які можуть бути задані параметрично (зокрема, Біноміальний закон розподілу).

З огляду на відсутність ретроспективних даних про проекти ПС досліджуваного класу використовуються якісні порядкові шкали значень (“низький”, “середній”, “високий” тощо) та суб’єктивні оцінки ймовірності цих значень, які є відображенням думок спеціалістів у проблематиці якості.

2. Аналіз артефактів проекту, з якими пов’язані чинники якості

Чинники якості пов’язані з артефактами проектів у категоріях – проект, робочий продукт, середовище (процеси) розроблення та експлуатації. Перелік характеристик артефактів проекту, збір даних про які пропонується включити в процес вимірювання, подано у табл. 2.3. Це дозволить у подальшому уточнювати підсистему прогнозування якості. Характеристики артефактів згруповані по категоріях та представлені нарівні з позначеннями стадій ЖЦ, на яких треба збирати дані для їх обліку.

Таблиця 2.3

Характеристики артефактів проекту для визначення чинників якості[1]

Артефакти проекту за категоріями	Стадії ЖЦ				
	Вв	**Пр**	**Ре**	**Те**	**Су**
1. Характеристики розроблюваного програмного продукту					
1. Тип та сфера застосування	+				
2. Розмір		+	+		
3. Складність		+	+		
4. Структурованість (модульність)			+		
5. Рівень мови програмування			+		
6. Характеристики відмовостійкості		+	+	+	+
7. Характеристики відновлюваності					+
8. Об'єм повторно використовуваного коду		+	+		
2. Характеристики робочих продуктів ПЗ					
9. Щільність дефектів (внесених, усунутих, залишок)	+	+	+		
10. Характеристики якості РП (наприклад, плинність вимог або рівень їх точності (однозначності), рівень складності та можливості реалізації проектних рішень, рівень документованості коду тощо)	+	+	+	+	+
3. Характеристики середовища та процесів розроблення					
Технологічна зрілість середовища розроблення					
11. Застосовувані методи та інструменти розроблення (адекватність потребам проекту)	+				
12. Рівень інтегрованості середовища та методології (CASE) (адекватність потребам)	+				
Зрілість процесів розроблення в проекті					
13. Зрілість організації (наприклад, за моделлю CMM)	+				
14. Рівень досконалості процесів керування (проектом, якістю, ризиком)	+				
15. Рівень досконалості процесів та методів, які забезпечують відстежуваність та своєчасне усунення дефектів		+			
16. Дотримання сучасних стандартів з програмної інженерії		+	+		
17. Професійна зрілість колективу проекту	+				
18. Характеристики процесів внесення дефектів		+	+	+	
19. Характеристики процесів усунення дефектів		+	+	+	+
20. Наявність історичних даних за подібними проектами	+	+	+	+	
4. Характеристики середовища застосування					
21. Операційний профіль (ймовірний розподіл частоти використання ПП різними категоріями користувачів)	+	+	+		
22. Технічні характеристики середовища експлуатації (швидкість процесора)	+	+	+	+	+
23. Якість процесу супроводження (умови відновлюваності, наявність методів, середовища, персоналу супроводження)					+

[1] Стадії ЖЦ: Вв – визначення вимог, Пр – проектування, Ре – кодування (реалізація) та автономне тестування, Те – системне та інші види тестування, СУ – супроводження під час експлуатації

З усієї множини чинників у табл 2.3 пропонується включити в БМ вершини для врахування впливу на залишкову щільність дефектів D_0 чинників 9, 15, 18, 19, чинники 6, 7, 20, 23 – не враховувати, а решту - враховувати комплексно у вершинах, які у сукупності характеризують якість розроблення поточного РП (чинники 11-14), якість перевірки поточного РП (чинник 15, 16) та складність проблеми, яка асоціюється з якістю робочого продукту, що є вхідним для поточної стадії (чинник 10).

3. Урахування особливостей моделі ЖЦ при визначенні топології БМ

Оскільки БМ призначена для прогнозування щільності дефектів у робочому продукті, який створюється під час проектування та кодування програмних застосувань ПС, вона повинна враховувати особливості адаптивної моделі ЖЦ.

4. Аналіз еталонних даних про артефакти проекту та вибір шкал

За орієнтири при виборі шкал, діапазонів та законів розподілу значень ВВ у вершинах БМ пропонується використовувати еталонні дані.

У табл. 2.4 подано еталонні дані, а у табл. 2.5 – дані про процеси виявлення та усунення дефектів, опубліковані С. Каном [81].

Таблиця 2.4

Еталонні дані для вибору шкал та діапазонів значень ВВ

Характеристика	**Показники високої потенційної надійності ПС**	**У середньому**
Щільність дефектів у ПП, який реалізує зовсім нові функції		0,97 дефектів/УОФ 14 дефектів/KSLOC
Щільність дефектів у ПП, який реалізує не нові функції		0,14 дефектів/УОФ 2 дефекти/KSLOC
Щільність виявлених дефектів у ході розроблення (для ПС інформаційно-аналітичної підтримки)		1,2 дефектів/УОФ 17,4 дефектів/KSLOC
Найвища ефективність усунення дефектів	> 95% усіх дефектів усуваються під час розроблення	> 56% усіх дефектів усуваються під час розроблення
Стабільність вимог	< 2.5% змін у базових вимогах	
Ясність вимог	> 97.5 % перевірених вимог	
Щільність дефектів після тестування	0.06 дефектів/УОФ	0.44 дефектів/УОФ 6,4 дефектів/KSLOC
Продуктивність розроблення (УОФ/чол.-міс.)	39 УОФ або 4250 SLOC у чол.-міс.	23 УОФ або 2500 SLOC у чол.-міс.

Продовження табл.2.4

Характеристика	Показники високої потенційної надійності ПС	У середньому
Найменша (по відношенню до середнього) щільність *потенційних* дефектів у артефакті проекту (у розрахунку на УОФ)	У вимогах - 0.20 У проектних рішеннях - 0.25 У коді - 0.25 У документації - 0.20 Не точно усунутих - 0.1 Усього 1.0 дефекта в УОФ	У вимогах - 1.0 У проекті - 1.25 У коді - 1.25 У документації - 1.0 Не точно усунутих - 0.5 Усього 5.0 дефектів в УОФ
Найменша щільність *фактичних* дефектів (у розрахунку на УОФ)	У вимогах - 0.02 У проекті - 0.0125 У коді - 0.003 У документації - 0.01 Не точно усунутих - 0.01 Усього 0.056 фактичних дефектів/УОФ	У вимогах - 0.16 У проекті - 0.10 У коді - 0.024 У документації - 0.08 Не точно усунутих - 0.08 Усього 0.444 фактичних дефектів/УОФ
Ефективність усунення дефектів (здатність усувати дефекти без внесення нових)	СММ рівень 5 - 95%, СММ рівень 4 - 93% СММ рівень 3 - 91%, СММ рівень 2 - 89% СММ рівень 1 - 85%	

Таблиця 2.5

Структура дефектів по стадіях ЖЦ ПС (дефекти/KSLOC)

Стадія ЖЦ	Успадковані	Внесено	Усього присутні	Ефективність усунення	Усунуто	Залишок
Аналіз вимог	Немає	1.2	1.2	Немає	Немає	1.2
Специфікація вимог	1.2	8.6	9.8	74%	7.3	2.5
Проектування	2.5	9.4	11.9	61%	7.3	4.6
Кодування	4.6	15.4	20.0	55%	11.0	9.0
Автономне тестування	9.0	Немає	9.0	36%	3.2	5.8
Системне тестування	5.8	Немає	5.8	67%	3.9	1.9
Випробування	1.9	Немає	1.9	58%	1.1	0.8
Експлуатація	0.8	Немає	Немає	Немає	Немає	Немає

Кожний з чинників дефектів може бути представлений у мережі вершиною одного з типів: *Labelled*, *Boolean*, *Numbered*, *Interval* (в термінології Hugin Lite 6.5). Вибір типу вершини визначається, перш за все, тим, якісні чи кількісні

характеристики відповідного чинника треба описати у вершині. Крім того, беруться до уваги типи всіх вершин, які є безпосередніми предками або нащадками даної вершини.

Нехай, наприклад, треба визначити типи вершин у БМ, поданій на рис. 2.5, яка моделює кількість виявлених дефектів під час перевірки.

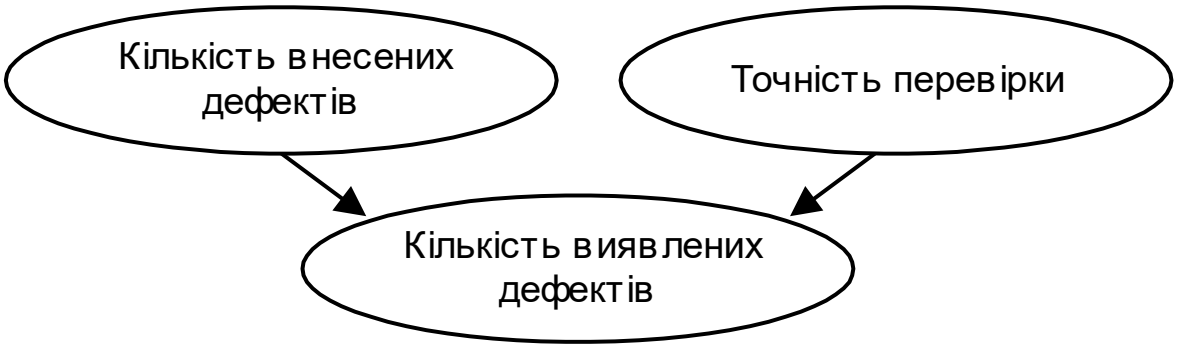


Рис. 2.5. Фрагмент байєсівської мережі

Приклади варіантів вибору шкал значень подано у табл. 2.6.

Таблиця 2.6

Приклади варіантів шкал значень змінних у вершинах БМ

Назва вершини	Шкала	Значення	Пояснення
Варіант 1			
Кількість внесених дефектів	Labelled	{“дуже мало”, “мало”, “середня кількість”, “багато”, “дуже багато”} Маргінальні (безумовні) ймовірності значень	Заповнення таблиць ймовірності вершин виконується вручну. Загальна кількість значень $5+3+5\cdot3\cdot5=83$
Точність перевірки	Labelled	{“висока”, “середня”, “низька” } Маргінальні ймовірності значень	
Кількість виявлених дефектів	Labelled	{“дуже мало”, “мало”, “середня кількість”, “багато”, “дуже багато”} Умовні ймовірності значень - $5\cdot3\cdot5$	
Варіант 2			
Кількість внесених дефектів	Interval	(0-5], (5-10], (10-20], (20-50], (50-100]	Бета-функція розподілу ймовірності
Точність перевірки	Interval	(0-1], (1-2], (2-3], (3-4]	
Кількість виявлених дефектів	Interval	*Beta* ((5 - 1) / 4 * *точність_ перевірки* + (5 - (5 - 1)), (1 - 5 / 4) * *точність_перевірки* + (1 - (1 - 5)), 0, *внесені_дефекти*)	

Продовження табл. 2.5

Назва вершини	Шкала	Значення	Пояснення
Варіант 3			
Кількість внесених дефектів	Numbered	0, 1, 2, 3, 4, 5	Біноміальний розподіл ймовірності
Точність перевірки	Interval	(0–1], (1-2], (2-3], (3-4], (4-5]	
Кількість виявлених дефектів	Numbered	Binomial (внесені_дефекти, точність _перевірки / 5)	

5. Побудова байєсівської мережі та перевірка її правильності

Безпосереднє розроблення БМ включає опис топології та всіх таблиць ймовірності значень змінних у вершинах (або функцій розподілу), її перевірку (компіляцію) і наступну ініціалізацію (навчання). В ініціалізованому стані з вершинами мережі пов’язані апріорні ймовірності значень відповідних змінних за відсутності очевидного свідчення (факту).

Формальна перевірка правильності БМ виконується автоматично при її компіляції. Перевіряється ациклічність графу та сумісність шкал значень у групах вершин БМ.

Крім того, перевірка правильності БМ полягає в аналізі конструкцій у структурі БМ, утворених послідовно, дивергентно або конвергентно з’єднаними вершинами, з метою визначення умов їх незалежності і можливих “блокувань” поширення ймовірності по БМ (рис. 2.6).

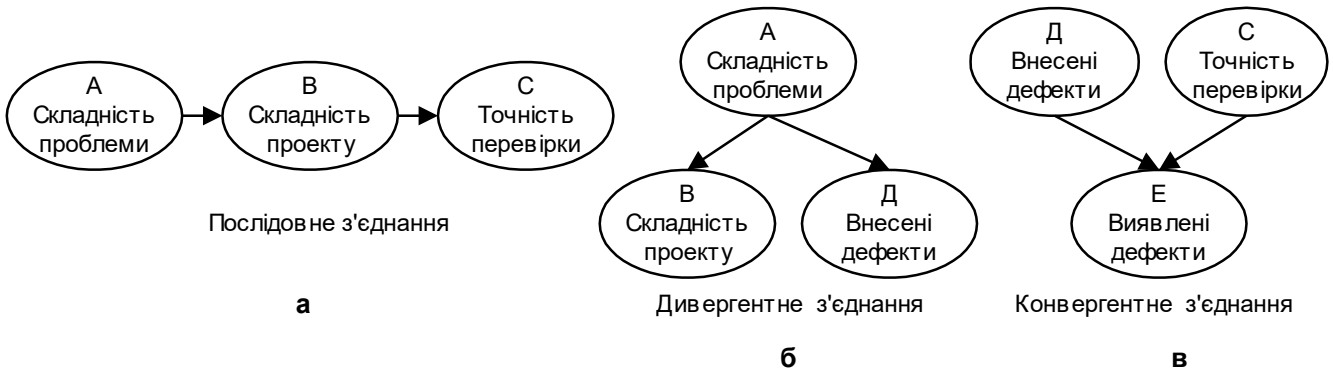


Рис. 2.6. Можливі варіанти з’єднання вершин

Наприклад, на рис. 2.6а шлях від вершини *A* до вершини *C* не заблокований лише тоді, коли нічого не відомо про значення у вершині *B*. Факт, введений у вершину *B* (на рис. 2.6а), робить вершини С та А незалежними (будь-яке значення А не важливе для С).

Аналогічно, факт, введений у *A* (на рис. 2.6б), також робить *B* та *Д* незалежними. Але, якщо нічого не відомо про *A*, але є відомості про *B*, це підвищує довіру до *A*, а також довіру до *Д*.

У випадку конвергентного з'єднання (рис. 2.6в), якщо нічого не відомо про *E*, - *Д* і *C* – незалежні. Але, якщо є відомості про *E*, - *Д* і *C* стають умовно залежними і до них обох підвищується довіра.

6. Перевірка чутливості мережі на гіпотетичних сценаріях подій

Для виконання аналізу чутливості БМ будуються сценарії можливих подій у проекті шляхом введення у вершини, які не мають батьків і відповідають чинникам впливу на дефекти, певних комбінацій фактів, та спостереження їх сукупного ефекту на розподіл ймовірності дефектів. Перевіряється, які чинники мають найбільший вплив на дефекти. В результаті можуть бути визначені чинники, вплив яких незначний і які можна видалити з мережі, а також некоректно визначені розподіли ймовірності ВВ.

Прогнозуюча здатність БМ перевіряється на "найгірших" та "найкращих" сценаріях подій у проекті.

2.2.4. Ієрархія з різною кількістю і складом альтернатив

Топологія Байєсвіської мережі наведена на рис. 2.7, а семантичний опис змінних у її вершинах – у табл. 2.7.

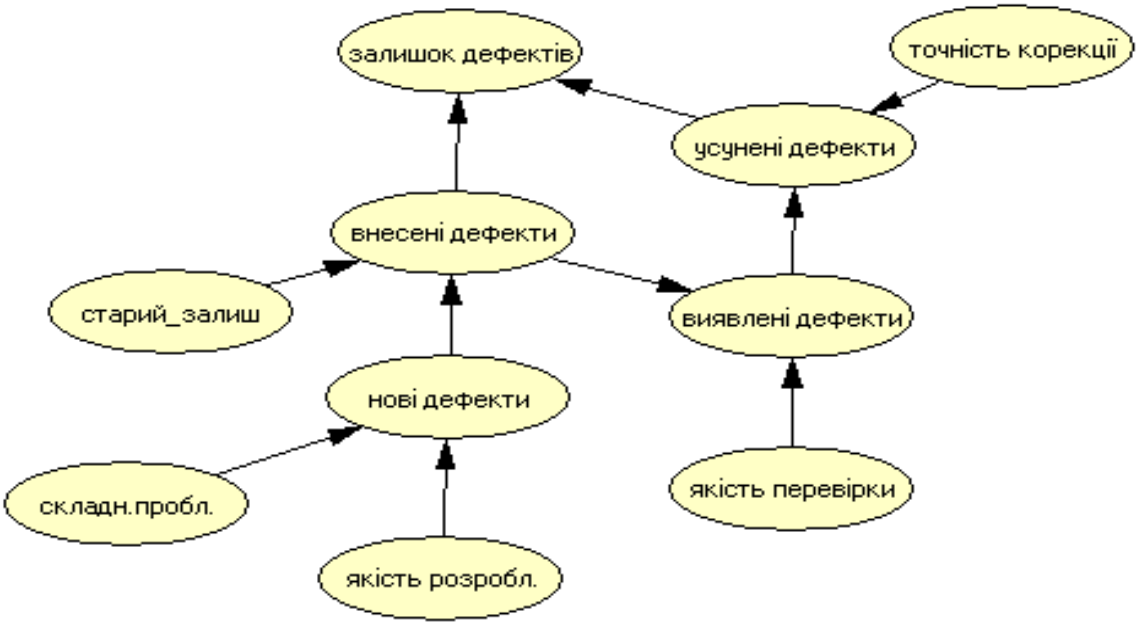


Рис. 2.7. Топологія БМ для прогнозування щільності дефектів у ПрЗ

Таблиця 2.7

Семантичний опис вершин БМ прогнозування дефектів

№ вер.	Назва вершини	Ім'я змінної	Тип шкали	Опис вершини та залежностей
1	Залишок дефектів	output_ defects	Numbe-red (0,…, 6)	Визначається як різниця між кількістю внесених дефектів у поточну версію РП і кількістю усунутих (відкоригованих) дефектів.
2	Внесені дефекти (загалом)	input_ defects_ all	Numbe-red (0,…, 6)	Визначається як сума кількості нових внесених дефектів (у даній фазі) та залишку дефектів (з попередньої фази)
3	Внесені дефекти (нові)	input_ defects_ new	Numbe-red (0,…, 3)	Кількість внесених нових дефектів залежить від складності вхідного РП та якості розроблення
4	Залишок старих дефектів	Output_ defects_old	Numbe-red (0,…, 3)	Кількість дефектів, які, згідно прогнозу, залишилися не виявленими у ПрЗ на минулій стадії розроблення
5	Усунені дефекти	fixed_ defects	Numbe-red (0,…, 6)	Кількість усунутих дефектів залежить від кількості виявлених дефектів, а також точності усунення дефекту (нормованої на [0,1])
6	Виявлені дефекти	discov_ defects	Numbe-red (0,…, 6)	Кількість виявлених дефектів залежить від кількості внесених дефектів, а також від якості перевірки (нормованої на [0,1])
7	Точність усунення	rework_ accuracy	Interval (0-3)	Здатність розробника точно усунути дефект. Чим вище значення, тим більша здатність

Продовження табл. 2.7

№ вер.	Назва вершини	Ім'я змінної	Тип шкали	Опис вершини та залежностей
8	Якість перевірки	verif_ qual	Interval (0-3)	Здатність верифікатора знайти дефекти у РП. Чим вище значення, тим більша здатність
9	Якість розробл.	develop_q uality	Interval (0-3)	Здатність розробника запобігти внесенню дефектів при розробленні. Чим вище значення, тим більша здатність
10	Склад-ність проблеми	problem_c ompl.	Labelled (3)	Асоціюється з ризиком проекту за категорією "складність реалізації вимог" до ПрЗ. Чим більша оцінка ризику, тим вища складність

Таблиці ймовірностей ВВ для вершин 7, 8, 9, 10 подано на рис. 2.8, а функції розподілу ймовірностей ВВ у вершинах 1, 2, 5, 6 – у табл. 2.8.

Rework_Accuracy	
0-1	0.1
1-2	0.6
2-3	0.3

Verif_qual	
0-1	0.1
1-2	0.5
2-3	0.4

Develop_Quality	
0-1	0.1
1-2	0.5
2-3	0.4

Problem_complex	
low	0.2
average	0.4
hign	0.4

Рис. 2.8. Таблиці ймовірності ВВ у вершинах 7, 8, 9, 10

Таблиця 2.8

Функції розподілу ймовірності у вершинах БМ

№ вер.	Функція (закон) розподілу	Пояснення
1	max (input_defects - fixed_defects, 0)	Детермінована функція. Залишкова кількість дефектів
2	input_defects_new+ output_defects_old	Детермінована функція. Загальна кількість внесених дефектів
5	Binomial (discov_defects, rework_accuracy/3)	Кількість усунутих дефектів. Робиться припущення, що кількість усунутих дефектів розподілена біноміально, *B(n,p)*, з параметрами *n* - кількість виявлених дефектів, *p* – ймовірність усунення дефекту. Оскільки *rework_accuracy* приймає значення на інтервалі (0-3), виконується нормалізація значень на інтервалі (0-1)
6	Binomial (input_defects _all, verif_qual / 3)	Кількість виявлених дефектів. Припущення – те саме *B*(input_defects_all, verif_qual)

Значення та розподіли ймовірності визначені у результаті аналізу еталонних даних про вплив різних чинників на щільність дефектів, а також даних у мультиплікативних моделях надійності. Дані, стосовні визначених якісних та

кількісних залежностей чинників були узгоджені із спеціалістами з груп якості (тестування) організацій-розробників, у яких впроваджені процеси верифікації та тестування і збираються дані.

З огляду на технічні обмеження реалізації БМ у Hugin Lite 6.5, побудована мережа не є темпоральною (динамічною), тобто змінна у вершині “Залишок дефектів” не індексована, а прогнозне значення щільності дефектів наприкінці однієї фази проекту автоматично не передається на наступну фазу для його підсумовування зі значенням у вершині “Внесені дефекти”. Для вирішення проблеми врахування залишкових дефектів у мережу введена вершина “Старий залишок”.

Ініціалізована БМ для прогнозування дефектів подана на рис. 2.9.

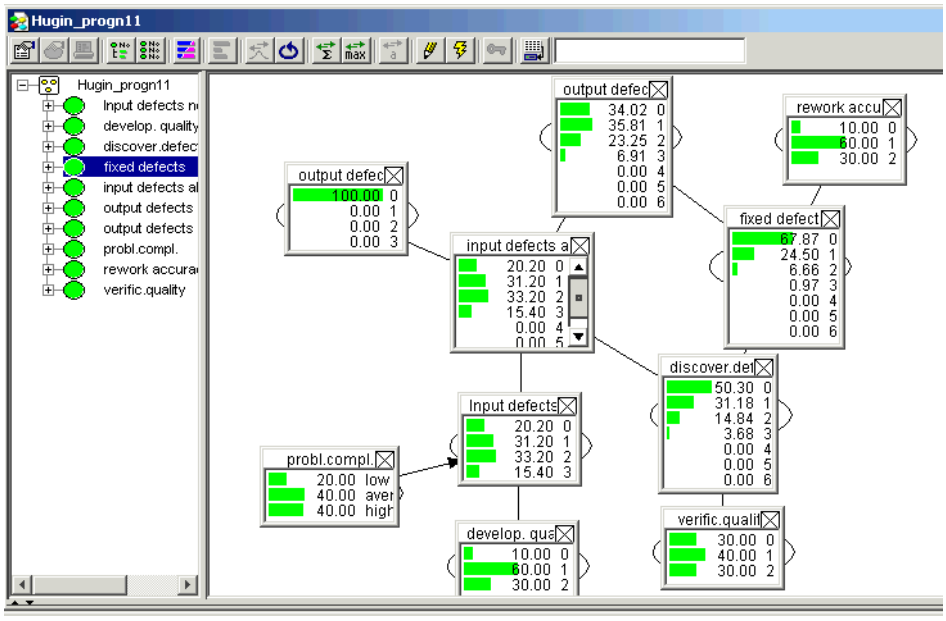


Рис. 2.9. Апріорні розподіли ймовірності у БМ

Чутливість мережі перевірялась на оптимістичному та песимістичному гіпотетичному сценаріях.

Оптимістичний сценарій. Якщо складність проблеми *низька*, а якість розроблення (ймовірність запобігти дефектам) та перевірки (ймовірність знайти дефекти) *високі*, а також *висока точність* усунення дефектів (ймовірність

правильно відкоригувати РП), - щільність дефектів буде *дуже низькою* (з ймовірністю 86%) (рис. 2.10).

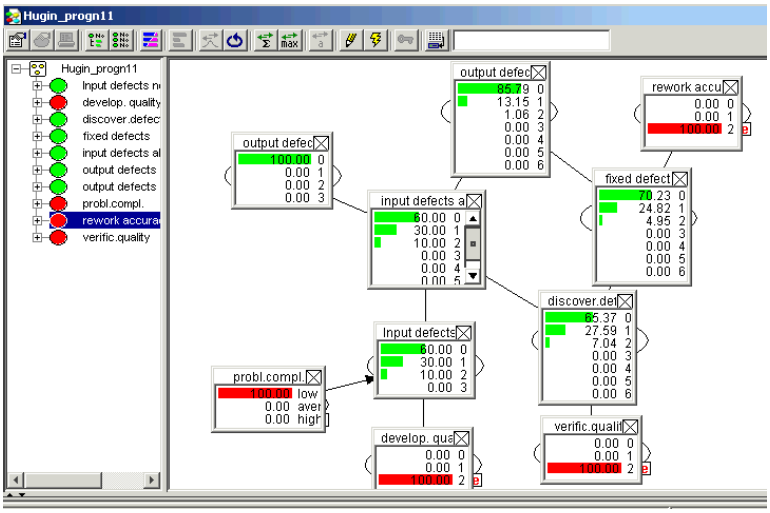


Рис. 2.10. Оптимістичний сценарій

Песимістичний сценарій. Якщо складність проблеми висока, а якість розроблення та перевірки низькі, а також низька точність усунення дефектів, - щільність дефектів буде високою (рис. 2.11).

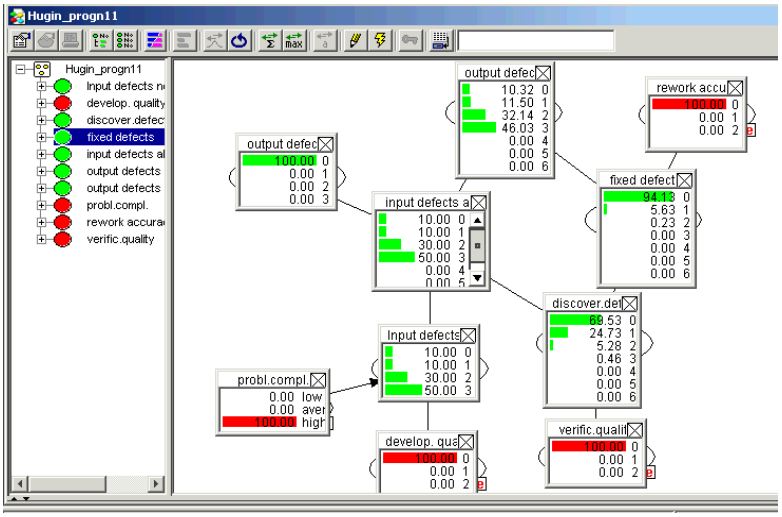


Рис. 2.11. Песимістичний сценарій

Розроблена модель дозволяє на початку поточної фази проекту прогнозувати (з певною мірою впевненості), якою буде щільність дефектів у програмних засобах, якщо не зміняться умови виконання проекту.

2.2.5. Метод аналізу альтернатив досягнення цільового рівня якості

Відповідно до методу прогнозування досяжності цільового значення надійності (безвідмовності) програмних застосувань, прогнозне значення щільності дефектів для кожного програмного засобу Z_i використовується у МЗНП для отримання значення ймовірності безвідмовного функціонування $R_i(t)$ протягом часу t експлуатації. Запропонований метод аналізу альтернатив базується на порівнянні значення $R_i(t)$ з цільовим розподіленим значенням безвідмовності q_i.

Умовою застосування методу є попереднє встановлення таких критеріїв для прийняття рішень з керування якістю i-го ПрЗ:

1. q_i – встановлене цільове значення надійності i-го ПрЗ;

2. D_i^* - максимальне значення щільності дефектів у ПрЗ, яке не буде перешкодою для досягнення цільового значення q_i, визначається з рівняння

$$q_i = \exp[-D_i^* I_i \cdot (1 - \exp(\frac{\lambda_{0i}}{D_i^* I_i} \cdot t))] ;$$

3. L_0 - мінімально припустима ймовірність прогнозного значення щільності дефектів (прийнятний рівень впевненості менеджера);

4. максимально припустимі відхилення:

σ_r – прогнозного значення надійності $R_i(t)$ від цільового значення q_i;

σ_d – прогнозного значення щільності дефектів D_{0i} від значення D_i^*;

σ_p – отриманої найбільшої ймовірності прогнозного значення щільності дефектів $L(D_{0i})$ від L_0.

Робиться припущення, що на ранніх стадіях ЖЦ $q_i \geq R_i(t)$ і $D_{0i} \geq D_i^*$.

Визначені такі альтернативи для прийняття рішень стосовно подальших дій із забезпечення якості ПрЗ:

1) якщо $|L(D_{0i}) - L_0| \geq \sigma_p$ (рис. 2.12а) і

2) (або $|D_{0i} - D_i^*| < \sigma_d$), продовжувати розроблення ПрЗ Z_i “як є”;

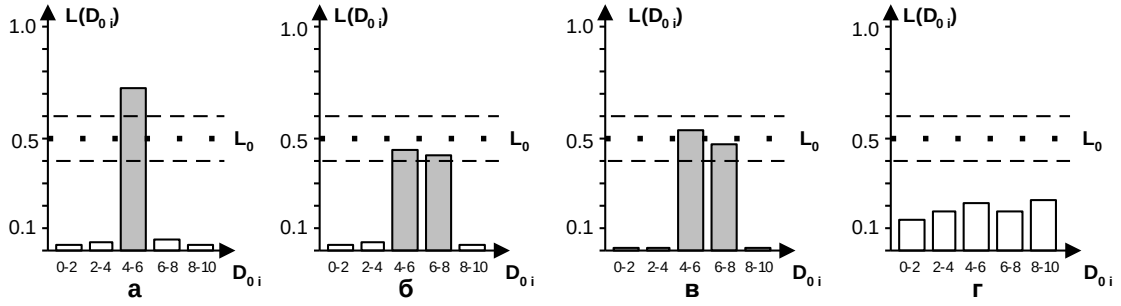


Рис. 2.12. Варіанти розподілу ймовірності значень D_0

3) якщо $|L(D_{0i}) - L_0| \geq \sigma_p$ (рис. 2.12а) і $|q_i - R_i(t)| > \sigma_r$, приймати рішення з огляду на оцінки ресурсів проекту, зокрема часу до завершення розроблення;

4) якщо $|L(D_{0i}) - L_0| < \sigma_p$ (рис. 2.12б або 2.12в), обчислити $R_i(t)$ у діапазоні близьких значень D_{0i} і оцінити відносні відхилення $R_i(t)$ від q_i для кожного з них, з огляду на те, що D_{0i} - це *щільність* дефектів у ПрЗ, розмір якого обчислений в УОФ, тобто діапазон значень $R_i(t)$ може бути досить широким;

5) якщо $|L_0 - L(D_{0i})| > \sigma_p$ (рис. 2.12г), тобто значення D_{0i} розподілені з ймовірностями, нижчими прийнятного рівня впевненості, або з самого початку проекту порушуються вихідні припущення для ранніх стадій ЖЦ ПрЗ - $q_i \geq R_i(t)$ і $D_{0i} \geq D_i^*$, це свідчить про недосконаліть застосовної моделі (малу кількість врахованих чинників дефектів, некоректно визначені апріорні розподіли змінних у вершинах без батьків тощо) та необхідність її уточнення. У цьому випадку менеджерові рекомендується застосувати діючі мультиплікативні моделі прогнозування дефектів.

Якщо в ході виконання проекту після стабілізації коду ПрЗ Z_i (тобто з наближенням до кінця періоду розроблення) не спостерігається тенденція до зниження щільності дефектів (за результатами прогнозувань в контрольних точках проекту t_s, t_{s+1},... отримуються значення D_{0i}, для яких $|D_{0i} - D_i^*| > \sigma_d$), це свідчить про недосконаліть процесів розроблення. Треба приймати рішення з огляду на оцінки залишку часу розроблення, а також часу та трудомісткості

системного тестування. Це може бути рішення про відлучення ПрЗ від розробника та його дострокову передачу на системне тестування, або про покращення процесу розроблення (шляхом кадрового “підсилення” проекту, навчання розробників, вдосконалення окремих процесів ЖЦ тощо).

2.3. Вдосконалення процесів життєвого циклу

Підвищення якості ПС здійснюється шляхом вдосконалення базового процесу ЖЦ ПС. У відповідності до стандарту ДСТУ 3918-95 модель базового процесу обов’язково включає всі процеси з категорії основних процесів розроблення, а також (можливо частково) процеси підтримки та організаційні процеси.

Існує чимало методологій поступового вдосконалювання процесів ЖЦ та підвищення якості ПС, стислий огляд яких подано в [2]. Це, наприклад, методології загального керування якістю (TQM, від Total Quality Management) [82]; розгортання процесів забезпечення якості (QFD, від Quality Function Deployment) [83]; гнучкі технологічні лінії (зокрема, Lean Software development, Quality Improvement Paradigm [84]); сходинки до якості (CMM, SPICE, Bootstrap, Trillium) тощо.

Для практичного застосування пропонуються такі підходи за трьома напрямками:

1) вдосконалення процесів розроблення:

– модель зрілості організації-розробника ПС (CMM (Capability Maturity model)) [27];

– еталонна модель у стандартах ДСТУ ISO 15504:2002 (SPICE), [2];

– модель бездефектного розроблення - Cleanroom [43];

2) підвищення професійної зрілості кадрів [2]:

– персональний процес розробника (PSP, Personal Software process) [2];

– колективний процес розроблення (TSP, Team Software process) [2];

3) вдосконалення процесу верифікації ПС [2]:

– наскрізний контроль; колегіальні перевірки, інспекції [89].

Вибір саме цих методологій обумовлений переконливими результатами порівняльного аналізу їх ефективності/вартості, отриманими Д. Ріко за даними 72 звітів провідних організацій-розробників ПС [9].

В умовах розроблення ПС за динамічним ЖЦ ключову роль у забезпеченні якості ПС на ранніх стадіях ЖЦ відіграє процес верифікації.

Отримання об'єктивних даних про стан продуктів, процесів і ресурсів розроблення ПС покладається на процес вимірювання – центральний процес в сучасній парадигмі якості. Результати вимірювань використовуються всіма іншими підтримувальними і організаційними процесами ЖЦ з метою прогнозування якості програмного продукту і оцінки ефективності виконання процесів.

Модель процесу вимірювання подано на рис. 2.13. Дії та задачі процесу регламентуються ISO/IEC 15939:2002 [10]. Оскільки стандарт не забезпечує науково-методичної підтримки виконання дій у процесі, автором запропоновано використовувати відому методологію “Ціль-питання-міра (метрика)” (GQM, Goal-Question-Measure.

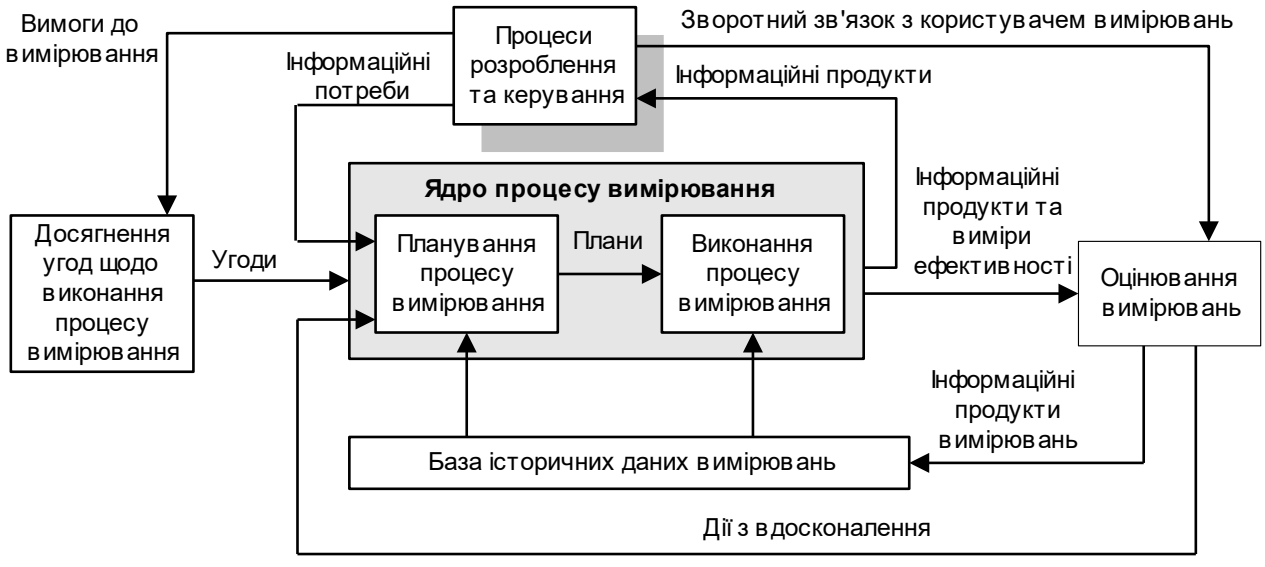


Рис. 2.13. Модель процесу вимірювання

Форми процесу вимірювання, рекомендовані для застосування, наведені в додатку Б.

2.4. Висновки до розділу

1. У розділі запропоновані методи та моделі, призначені для підтримки прийняття рішень з керування якістю в ході програмного проекту.

2. Поставлена та вирішена задача апріорі визначити необхідний оптимальний рівень надійності кожного модуля та програмного застосування шляхом максимізації функції загальної корисності ПС, позбуваючися необхідності отримання математичного формулювання надійності функцій ПС. Обґрунтовується, що модель розподілу надійності є вирішуваною задачею нелінійної оптимізації, де безвідмовність модулів – невідомі змінні.

3. Запропоновано метод прогнозування надійності програмних застосувань - найбільш важливої характеристики якості для ПС оброблення даних. В основі методу – прогнозування щільності дефектів у програмних засобах на початку кожного етапу проекту. Розроблена графічна модель прогнозування щільності дефектів із застосуванням байєсівської мережі, що забезпечує врахування причинно-наслідкових зв’язків чинників та ймовірнісні судження менеджерів про поточну ситуацію в проекті.

Хоча ранні прогнози, за своєї природи, менш точні, ніж оцінки по фактичних значеннях, - вони служать стимулом до покращення якості в процесі розроблення, оскільки дають можливість оцінити ефективність застосовних методів розроблення ПС в контексті майбутніх відмов. Якщо організація чекатиме, поки будуть зібрані дані про відмови ПС, може бути вже дуже пізно робити істотні підвищення її якості.

РОЗДІЛ 3
РОЗРОБКА ЗАСОБІВ ПІДТРИМКИ МОДЕЛЕЙ ТА МЕТОДУ ПРОГНОЗУВАННЯ ЯКОСТІ ПРОГРАМНИХ СИСТЕМ

3.1. Розробка програмно-технологічного модуля підтримки процесу прогнозування якості програмних систем

3.1.1. Структура програмно-технологічного модуля

При процесному підході до розроблення ПС структурним об’єктом, який інкапсулює методи і алгоритми виконання дій у процесах ЖЦ, є програмно-технологічний модуль [3]. В ньому упорядковується сукупність цілеспрямованих операцій, які:

– виконуються за встановленою технологією, що підтримує відповідний метод виконання процесу (дії);

– об'єднані технологічним маршрутом їх виконання (операційним сценарієм);

– мають відомі передумови виконання, входи та виходи;

– забезпечені інформаційною, інструментально-програмною та методичною підтримкою виконання.

Загальна структура технологічного модуля (ТМ) наведена на рис. 3.1.

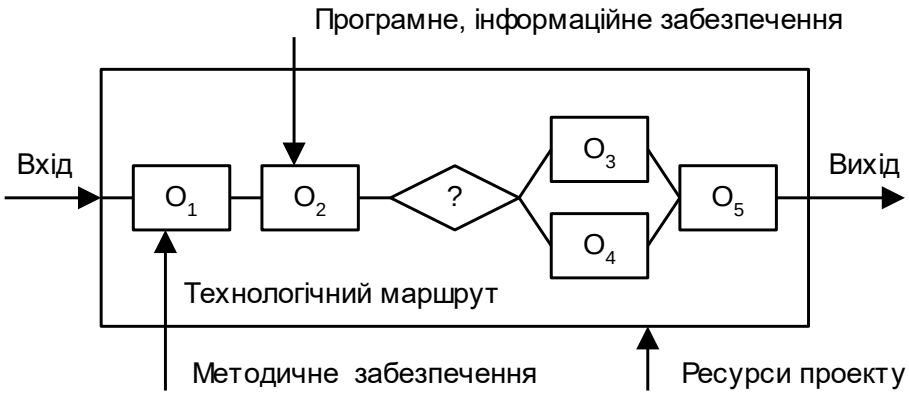


Рис. 3.1. Структура програмно-технологічного модуля

ТМ будується на стадії технологічної підготовки розроблення ПС виходячи із знання профілю процесу і конкретних цілей і умов виконання процесу [2].

Для вирішення задач інженерії якості розроблені такі ТМ: розподілу надійності по компонентах ПС; прогнозування надійності ПС на ранніх стадіях ЖЦ ПС та аналізу стану забезпечення якості; підтримки процесів верифікації та вимірювання.

Інформаційно-програмне забезпечення всіх ТМ виконує розроблений програмний комплекс.

3.1.2. Основні функції програмного комплексу

Програмний комплекс забезпечує застосування всіх технологічних модулів для підтримки прийняття рішень у сфері інженерії якості. Архітектурно-функціональну схему програмного комплексу наведено на рис. 3.2.

ПК розроблений засобами Visual Basic 6.0, Visual Basic for Application (VBA) та використовує бібліотеки функцій таких компонентів загальносистемного середовища:

– HUGIN Lite 6.5 API для реалізації графічної моделі прогнозування дефектів;

– MATLAB 6 для вирішення задач матричної алгебри, а також нелінійної оптимізації при реалізації моделі розподілу надійності по компонентах ПС;

– Oracle – ADO та OLE Object для організації обміну даними в інтегрованому інформаційному середовищі (БД Oracle, таблиці Excel), а також реалізації функцій верифікації у середовищі Oracle Designer’2000.

Основні функціональні відмінності програмного комплексу від відомих наборів інструментів аналізу якості, наприклад, Rational Suite такі:

– інформаційно-програмне забезпечення повного циклу прийняття рішень з керування якістю (функції розподілу надійності по компонентах ПС, прогнозування надійності, аналіз альтернатив);

– інформаційно-програмна інтеграція з програмного комплексу HUGIN Lite 6.5 та MATLAB 6 і використання їх функцій безпосередньо у ПК;

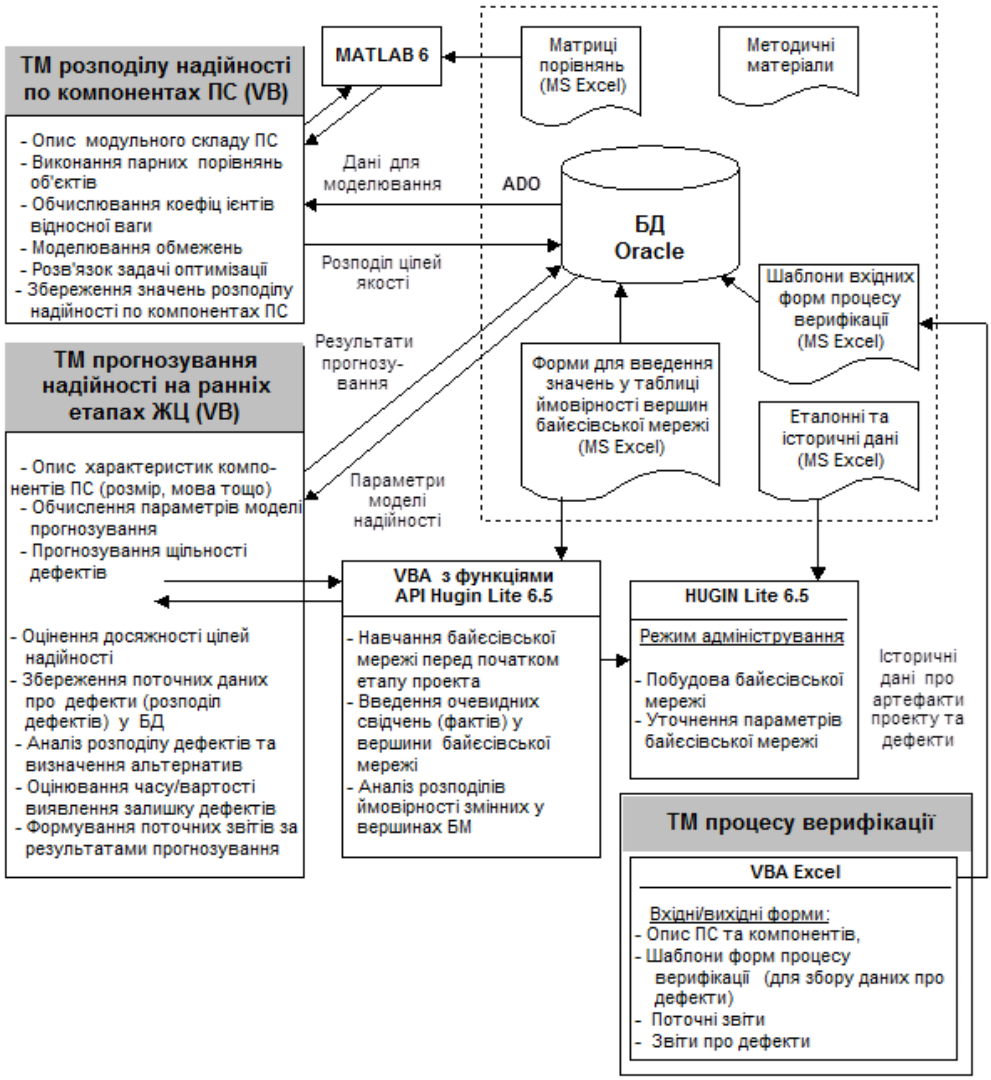


Рис. 3.2. Архітектурно-функціональна схема програмного комплексу

– використання функцій та утиліт Oracle Repository Reports Runtime для отримання звітів за даними репозиторію Oracle Designer’2000;

– прогнозування щільності дефектів за двома моделями (RLM та розробленою експериментальною графічною моделлю);

– ведення бази історичних даних про дефекти, виявлені та усунені під час розроблення.

Головне вікно програмного комплексу подано на рис. 3.3, а функції головного меню стисло описані в табл. 3.1.

Рис. 3.3. Головне вікно програмного комплексу

Таблиця 3.1

Функції головного меню програмного комплексу

Функція	**Опис функції**	**Область дії (лист книги)**
Доступ	Встановлення доступу до БД Oracle	Будь-який лист
Опис	Опис ПС, опис програмного засобу (мова, розмір, розробник, верифікатор), модулів (вартісні обмеження тощо), характеристик процесу верифікації	“Опис_ПС”
Експорт	Аналіз форми (залежно від назви листа) та завантаження даних у таблиці БД, пов’язані з формою	Будь-який лист, крім “Службовий”
Розподіл	Перехід у середовище “ТМ розподіл надійності” для визначення параметрів моделі розподілу надійності та розв’язку задачі	“Матриці”
Прогноз	Перехід у середовище “ТМ прогнозування надійності” для оцінки значень параметрів моделей RLM та БМ і виконання прогнозів	“Параметри_ прогноз”
Аналіз	Аналіз альтернатив	“Стан”
Звіти	Формування регламентованих звітів за даними у БД, відповідно до множини запитів	“Запити”
Очистити	Очистка вмісту листа Excel	Крім “Службовий”

3.1.3. Інформаційне забезпечення

Інформаційне середовище програмного комплексу складають таблиці БД Oracle, шаблони форм Excel.

Таблиці БД Oracle призначені для:

– накопичення загальних даних про проект ПС (дати початку, завершення, контрольних точок для збору та аналізу даних, функції тощо);

– опису модульного складу розроблюваних ПС і характеристик модулів та програмних застосувань (розмір початкового коду, розмір об’єктного коду, мова (середовище) програмування, цільове значення надійності, фактично оцінена надійність, вартість, ризики тощо);

– опису середовища функціонування (характеристики комп’ютерів (швидкість процесора), специфікації встановлюваних ПрЗ тощо);

– збереження даних, отриманих за моделями розподілу надійності та прогнозування надійності для кожного ПрЗ.

Оперативні таблиці БД та класифікатори описано у табл. 3.2.

Таблиця 3.2

Оперативні таблиці програмного комплексу

Назва таблиці	Призначення
Software	Ідентифікація системи
Funct_PS	Опис функцій системи
Prz_PS	Опис програмних застосувань системи
PassportModule	Опис модулів системи (паспорти)
SoftComprOf	Структурний склад системи
Etaps_PRZ	Стан ПрЗ на визначеному етапі розроблення
Weight_Loc	Локальні коефіцієнти відносної ваги об’єктів системи
Weight_Glob	Глобальні коефіцієнти відносної ваги об’єктів системи
Ver_Consist	Результати перевірки узгодженості парних порівнянь об’єктів
VerifDefectDescr	Опис дефектів, виявлених під час верифікації
Processor_char	Характеристики техніки, постачуваної замовникові
Severity_defect	Шкала серйозності дефектів (класифікатор)
Stan_defect	Стан дефекту (класифікатор)
Obj_hierarchy	Типи об’єктів у ієрархії системи (класифікатор)
Kl_ODC	Категорії дефектів за методом ODC (класифікатор)

Шаблони форм Excel призначені для:

– збору даних про дефекти та їх класифікації;

– збору даних про парні порівняння функцій, ПрЗ та модулів ПС;

– коригування таблиць імовірності вершин моделі дефектів у ПрЗ;

– введення об’єктивних свідчень про поточний стан проекту ПрЗ (формування параметрів моделі RLM та графічної моделі);

– отримання з БД та графічної інтерпретації розподілу надійності та прогнозованих значень надійності.

Функції технологічного модуля виконуються менеджером проекту після визначення функціональних та технічних вимог до ПС та побудови *першого прототипу*, призначеного для деталізації та узгодження з замовником вимог до окремих ПрЗ, а також визначення модульного складу та об’єму повторно використовуваного коду [97]. Склад ТМ описано в табл. 3.5.

Таблиця 3.5

Склад технологічного модуля “Розподіл надійності”

Складові ТМ	Опис складових ТМ розподілу надійності
Базовий метод та модель	MAI та модель розподілу надійності
Інформаційна підтримка	СКБД Oracle. Опис ПС, модульного складу та вимог у репозитарії Oracle Designer’2000
Інструментальна підтримка	Середовище VBA, Visual Basic 6.0, Matlab 6.
Вхід	Функціональні вимоги до ПС. Вимоги до надійності
Вихід	Цільові значення надійності, експортовані до БД
Операції	1. Опитування користувачів і парне порівняння функцій. Складення матриці порівнянь функцій (у Excel). Експорт до БД. 2. Опитування розробників ПС і парне порівняння ПрЗ. Складення матриць порівнянь ПрЗ (у Excel). Експорт до БД. 3. Вибір (проектування) модулів. Опитування розробників і парне порівняння модулів. Складення матриць порівнянь модулів (у Excel). Експорт до БД. 4. Коригування опису ПС, матриць (у середовищі ТМ). 5. Визначення обмежень моделі. 6. Виконання розподілу та збереження цільових даних надійності для кожного ПрЗ.

Головне вікно ТМ “Розподіл надійності” з розгорнутою формою для оброблення та збереження у БД результатів парних порівнянь об’єктів зображене на рис. 3.5. Функції головного меню стисло описані в табл. 3.6.

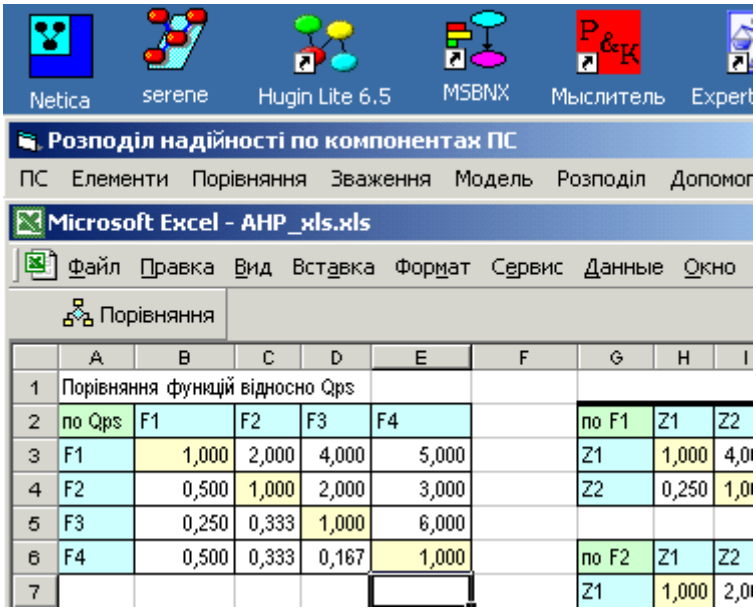


Рис. 3.5. Головне вікно ТМ з формою для парних порівнянь об’єктів

Таблиця 3.6

Функції головного меню ТМ “Розподіл надійності”

Назва елемента меню	**Опис функції**
ПС	Опис ПС
Елементи	Опис елементів ПС (програмних застосувань, модулів)
Порівняння	Оброблення матриць парних порівнянь та збереження даних порівнянь у БД
Зважування	Визначення власних значень і власних векторів, індексів та коефіцієнтів узгодженості. Формування висновків про необхідність повторного порівняння
Модель	Визначення узагальнених вагових коефіцієнтів для ПрЗ та модулів (ієрархічний синтез). Встановлення обмежень.
Розподіл	Розв’язок задачі оптимізації. Збереження результатів розрахунку цільових розподілених значень надійності для модулів у БД, визначення цільових значень надійності для кожного ПрЗ
Допомога	Довідкова підсистема з описом функцій ПК

Особливість реалізації моделі розподілу надійності полягає в тому, що підготовчі функції, зокрема збір даних порівняння компонентів ПС, виконуються

в середовищі Excel, а знаходження власних значень та векторів, а також розв'язок задачі оптимізації – в середовищі MATLAB.

Для отримання λ_{max} та W застосовується алгоритм, представлений фрагментом коду на VBA (кроки алгоритму описані у вигляді коментарів до коду):

```
……
    matlabinit                  ' Запуск MATLAB
'Експорт даних до MATLAB
    MLPutVar "M", MZ                              ' Матриця парних
порівнянь
    MLPutVar "LV", LV                        ' Вектор власних значень
    MLPutVar "MaxL", MaxZ               ' Максимальне власне значення
' Виконання команд MATLAB
    MLEvalString "LV = eig(M)"          ' Обчислення власного вектора
матриці
    MLEvalString "MaxL = max(LV)"  ' Обчислення max власного значення
' Імпорт результатів в Excel
    MLGetVar "LV", LV
    MLGetVar "MaxL", MaxZ
…….
```

Звіт про оброблення результатів порівнянь виконується за формою, поданою на рис. 3.6. Він містить перелік вагових коефіцієнтів для вибраних об'єктів відносно визначеного критерію, а також індексів та коефіцієнтів узгодженості. Користувачеві надаються висновки про результати оброблення матриць.

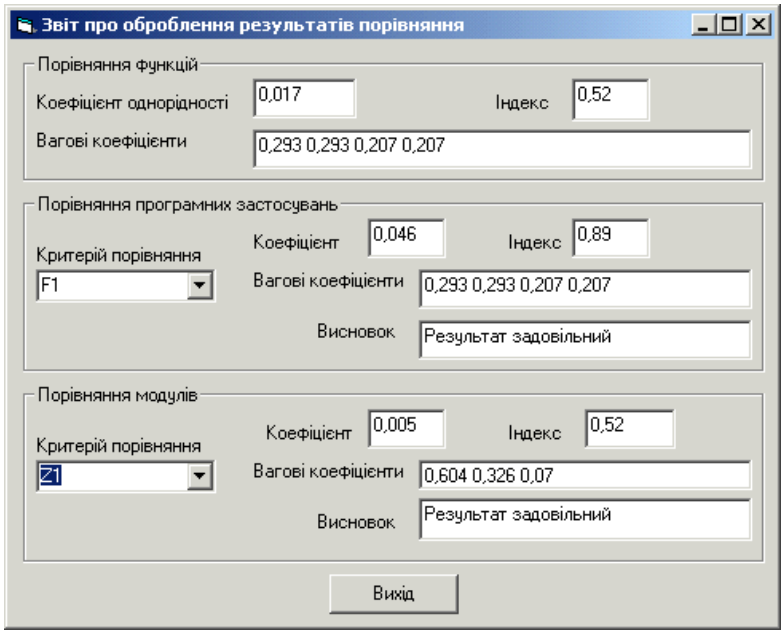


Рис. 3.6. Звіт про оброблення результатів порівняння

Після визначення локальних вагових коефіцієнтів виконується їх згортка (ієрархічний синтез) та отримання узагальнених вагових коефіцієнтів для ПрЗ та модулів, тобто формуються параметри моделі розподілу надійності. Ці результати відображаються у формі, поданій на рис. 3.7, де користувач може послідовно встановити обмеження (загальні та для кожного модуля), потрібні для розв’язку задачі оптимізації.

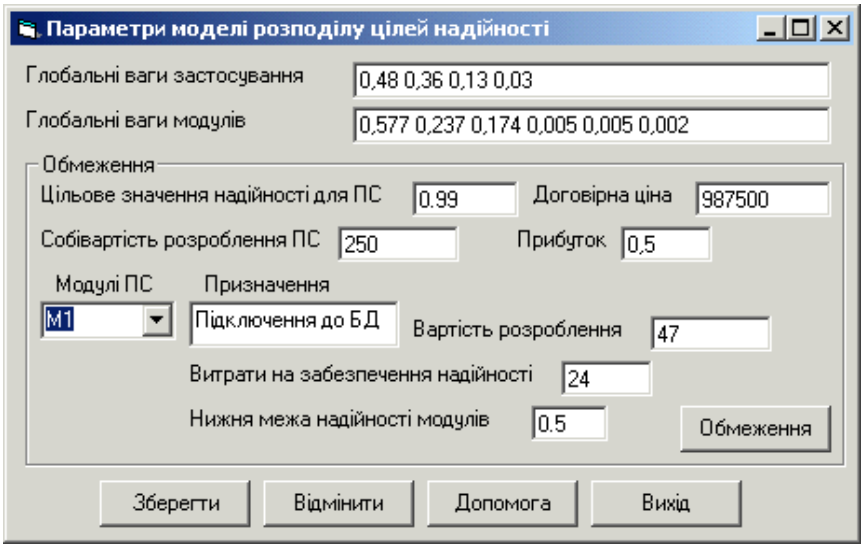


Рис. 3.7. Підготовка параметрів моделі розподілу надійності

Задача оптимізації розв’язується за допомогою пакету MATLAB 6.5 (компонент Optimization Toolbox, функція fmincon(-Qps,…). Її розв’язок виконується за таким алгоритмом:

– формування опису цільової функції у вигляді програмного коду у M-файлі MATLAB;

– формування матриці та вектору лінійних вартісних обмежень, векторів нижніх та верхніх меж надійності модулів, які використовуються як вхідні параметри для програми оптимізації (також M-файл MATLAB);

– автоматичне підключення до MATLAB та розв’язок задачі оптимізації;

– відображення користувачу отриманих результатів, їх збереження у змінних VB та закриття MATLAB;

– збереження отриманого розподілу надійності у БД.

За результатами розв’язку задачі оптимізації формується звіт про розподіл надійності по об’єктах ПС.

3.2. Програмно-технологічний модуль прогнозування надійності

Оцінка розміру коду виконується в умовних одиницях функціональності з урахування кількості та складності функцій, які реалізує ПрЗ, і кількості та складності даних, якими оперують ці функції. В даному технологічному модулі реалізовано базовий FP-метод оцінювання розміру, детально описаний у [2, 48]. Для ПрЗ, які розробляються засобами Oracle Designer’2000, оцінки розміру отримуються безпосередньо із звіту утиліти Oracle Repository Reports “Function point analysis”. Ця утиліта виконує оцінки на двох рівнях – за даними аналізу ієрархії функцій та діаграм потоків даних і за даними опису проекту програмного засобу у репозитарії Oracle, та застосовує метод “FPA Mk-II”. Звіт виконується у вигляді HTML-файлу, придатного для аналізу засобами технологічного модуля прогнозування надійності.

Отримана оцінка розміру в УОФ конвертується у KSLOC з урахуванням характеристики середовища розроблення ПрЗ (а саме, мови програмування).

Прогнозування щільності дефектів виконується у ТМ двома методами – за моделлю RLM та за розробленою графічною моделлю.

Прогнозування за RLM полягає в оцінюванні впливу на щільність дефектів 9 чинників. Отримані оцінки є модифікаторами базового значення щільності дефектів для визначеного класу ПрЗ.

Метод прогнозування щільності дефектів за розробленою графічною моделлю полягає у визначенні ймовірності значень змінних у вершинах Байєсівської мережі “Складність проблеми”, “Якість розроблення”, “Якість перевірки” та “Точність корекції” з огляду на поточний стан проекту, і їх наступного експорту у байєсівську мережу.

Значення всіх параметрів моделі прогнозування надійності зберігаються у БД.

3.2.1. Склад та функції технологічного модуля прогнозування надійності

Операції технологічного модуля виконуються менеджером проекту на початку кожного етапу розроблення програмного засобу. Склад технологічного модуля описано у табл. 3.7.

Таблиця 3.7

Склад технологічного модуля “Прогнозування надійності”

Складові ТМ	Опис складових ТМ прогнозування надійності
Базовий метод та модель	Розроблений метод прогнозування, модель Муси, модель RLM, метод FPA, розроблена графічна модель, метод аналізу альтернатив для прийняття рішень
Інформаційна підтримка	СКБД Oracle, утиліта Oracle Repository Reports, шаблон БМ у файлі “BM_Defects.net”, форми Excel для обміну даними з БМ.
Інструментальна підтримка	Середовище VB, VBA, API-функції Hugin Lite 6.5.
Вхід	Значення параметрів моделі RLM, залишок дефектів від попереднього етапу робіт з ПрЗ, значення параметрів моделі прогнозування надійності, цільове розподілене значення надійності для ПрЗ.

Продовження таблиці 3.7

Складові ТМ	Опис складових ТМ прогнозування надійності
Вихід	Прогнозоване значення надійності для ПрЗ у порівнянні з цільовим розподіленим значенням. Рекомендації (альтернативи) щодо подальших дій.
Операції	1. Коригування змінних у вершинах БМ для врахування поточного стану проекту ПрЗ. 2. Оцінка параметрів моделі прогнозування надійності. 3. Прогнозування щільності дефектів. 4. Визначення порогових значень (допусків) для оцінок результатів прогнозування та аналізу альтернатив. 5. Розрахунок прогнозного значення надійності для ПрЗ.

Головне вікно технологічного модуля “Прогнозування надійності” з розгорнутою формою для визначення параметрів моделі RLM зображене на рис. 3.8. Функції головного меню стисло описані в табл. 3.8.

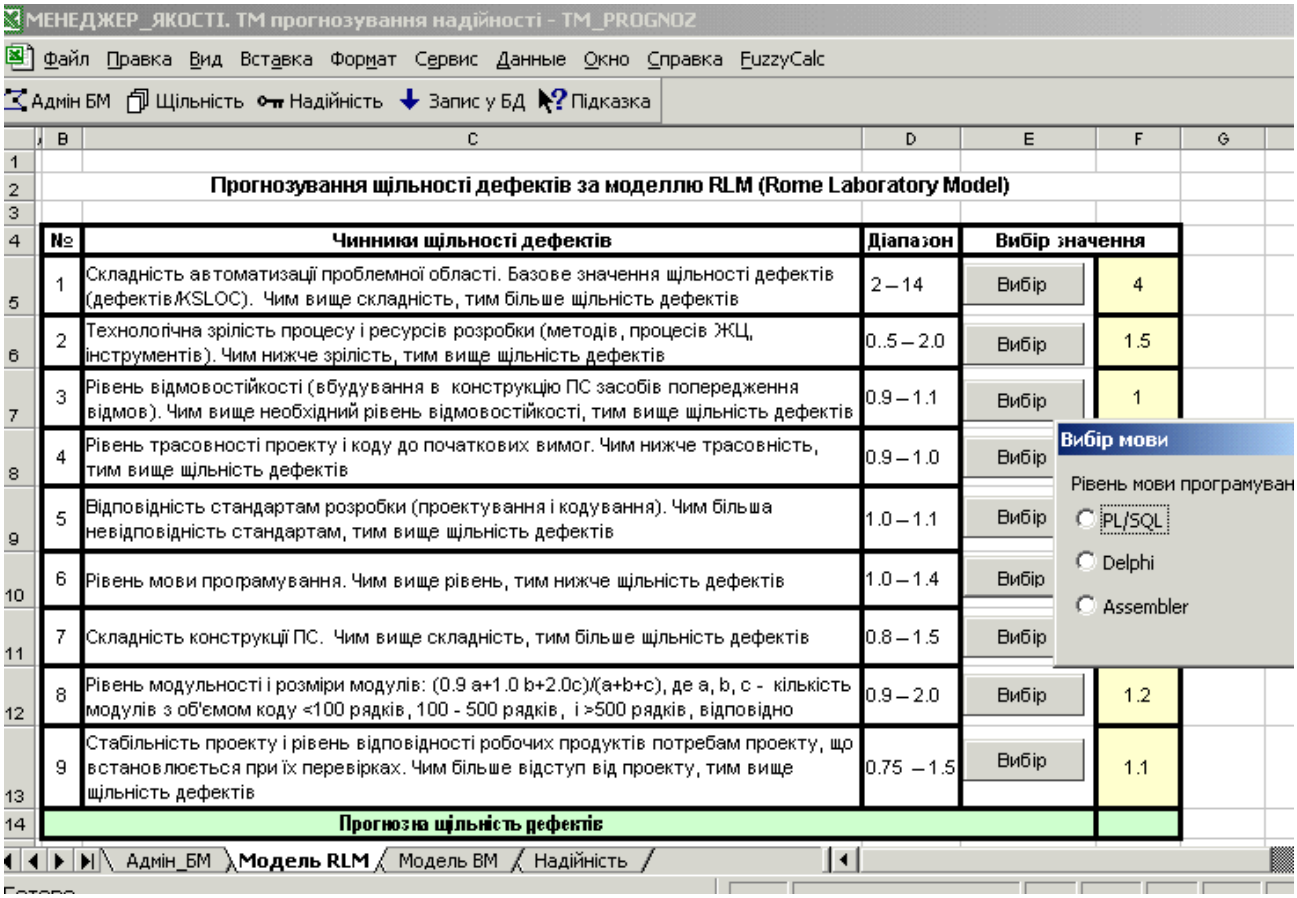


Рис. 3.8. Головне меню ТМ з формою для застосування моделі RLM

Таблиця 3.8

Функції головного меню ТМ “Прогнозування надійності”

Назва елемента меню	Опис функції	Область дії (лист книги)
Адмін БМ	Настройка БМ: імпорт значень ТІВ у Excel, їх коригування та експорт до ТІВ, компіляція БМ	“Адмін БМ”
Щільність	Прогнозування щільності дефектів: за моделлю RLM (на листі “Модель RLM”), за розробленою графічною моделлю (на листі “Модель ВМ”)	“Модель RLM” “Модель ВМ”
Надійність	Прогнозування надійності за розробленим методом	“Надійність”
Підказка	Контекстна довідка про кнопки, які діють на даному листі	Будь-який лист крім “Службовий”

3.2.2. Прогнозування щільності дефектів за графічною моделлю

Графічна модель щільності дефектів (байєсівська мережа) будується за розробленим шаблоном БМ окремо для кожного ПрЗ на етапі технологічної підготовки до його розроблення. Для цього шаблон БМ у файлі *BM_Defects.net* тиражується та перейменовується (наприклад, *BM_PKVVID. net*), а потім виконується “настроювання” БМ у відповідності до оцінок поточного стану виконання проекту ПрЗ. Оцінки отримуються через віконний інтерфейс, поданий у вигляді опитувальних листів (рис. 3.9).

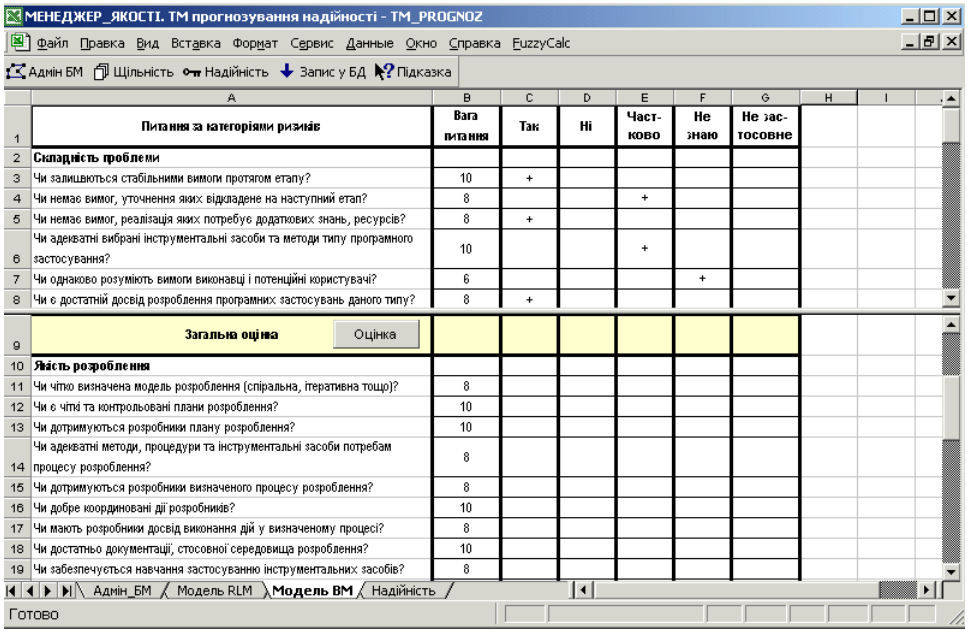


Рис. 3.9. Форма для отримання оцінок поточного стану ПрЗ

Питання формулюються таким чином, щоб відповіді на них однозначно характеризували стан відповідного артефакту проекту, описаного змінною у вершині БМ. Відповіді на питання визначають вибір того значення ВВ, яке буде розпізнаватися мережею як об’єктивний факт (подія). Для кожної категорії питань значення підсумовуються і використовуються для коригування розподілів значень у вершинах.

Форма для настроювання БМ подана на рис. 3.10, а призначення кнопок описано в табл. 3.9.

Таблиця 3.9

Призначення кнопок форми для роботи з БМ у середовищі Excel

Кнопка	Призначення кнопки
Імпорт БМ	Доступ до таблиць ймовірності вершин БМ та перенесення ймовірностей значень змінних у відповідні таблиці Excel.
Експорт БМ	Доступ до таблиці ймовірності вершини БМ, яка відповідає таблиці Excel, та перенесення ймовірностей значень із таблиці Excel у ТІВ.
Залишок	Перенесення ймовірностей значень у таблиці Excel “Залишок дефектів” до таблиці “Залишок старих дефектів”
Компіляція	Збереження БМ у файлі “*.net” та компіляція БМ, після чого БМ готова до прогнозування щільності дефектів
Запис у БД	Збереження розподілів ймовірностей значень змінних БМ у БД

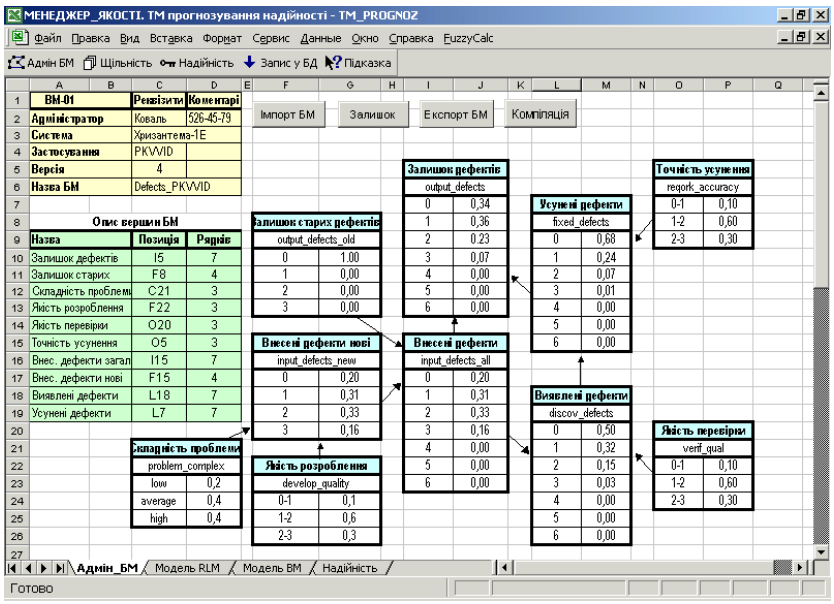


Рис. 3.10. Форма для обміну інформацією з ТМ байєсівської мережі

3.2.3. Оцінка надійності програмного застосування

Прогнозна оцінка надійності (безвідмовності) ПрЗ виконується у VBA на листі “Надійність” після отримання на екрані розрахованих значень параметрів моделі надійності, вибраних із БД за допомогою кнопки “Параметри” (рис. 3.11).

Прогнозне значення надійності ПрЗ обчислюється за моделлю прогнозування при натисненні кнопки “Надійність” на листі Excel. Результат прогнозування зберігається у БД.

Аналіз стану забезпечення якості виконується за кнопкою “Аналіз” на панелі інструментів ПК Менеджер_Якості. Відбувається пошук у БД і відображення на листі “Стан” переліку назв ПрЗ (рис. 3.12).

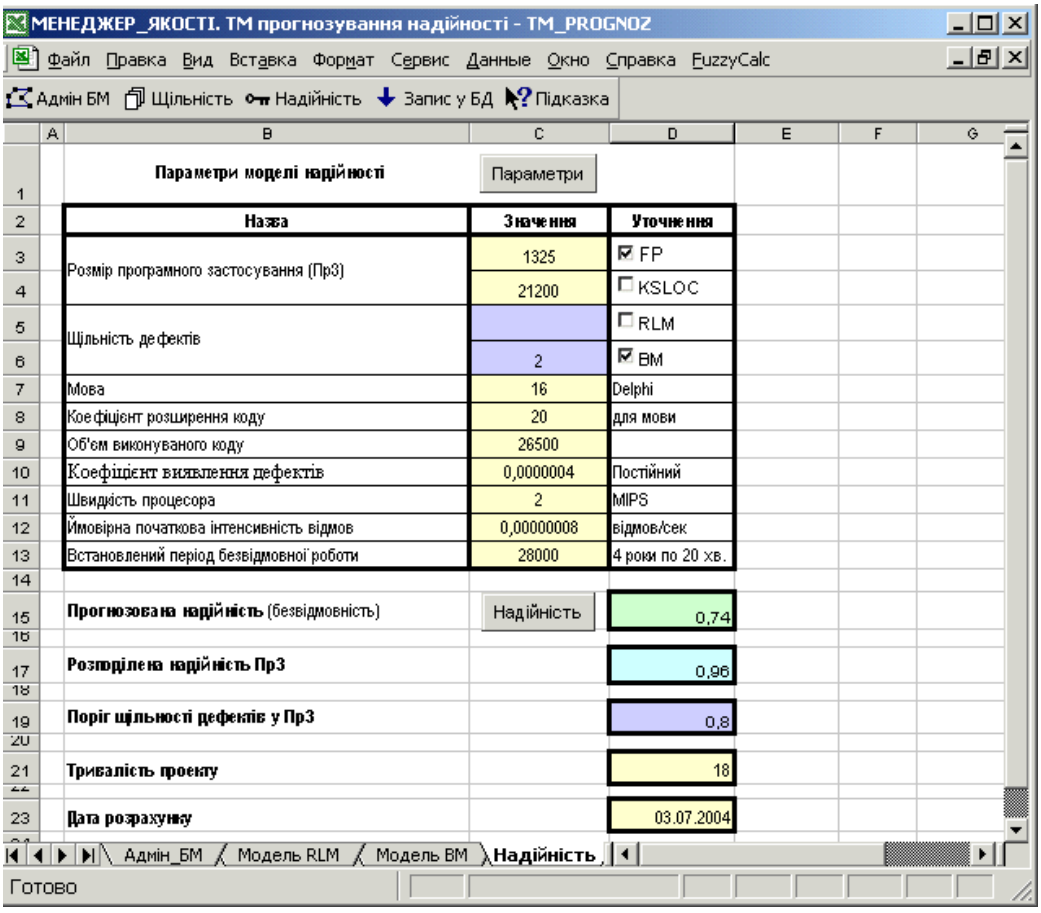


Рис. 3.11. Форма для визначення прогнозної оцінки надійності ПрЗ

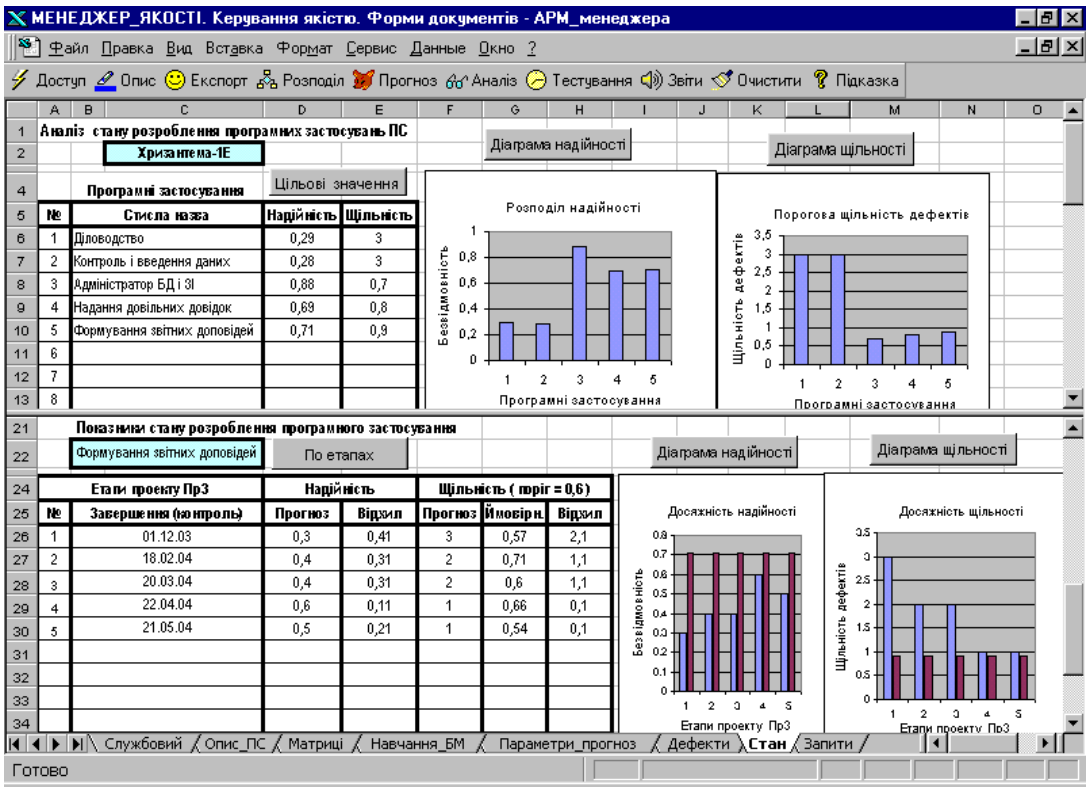


Рис. 3.12. Форма для аналізу стану забезпечення якості ПС

У верхній частині листа подається підсумкова інформація про розподіли цільових значень надійності ПрЗ та порогових значень щільності дефектів, за яких можливе досягнення цих значень.

У нижній частині листа (після вибору ПрЗ із переліку і натиснення кнопки “По етапах”) виконується пошук у БД та відображення переліку етапів та прогнозних характеристик надійності і щільності дефектів у ПрЗ.

За кнопками “Діаграма…” відображаються порівняльні гістограми, які візуалізують стан виконання проекту за критерієм якості.

3.3. Програмно-технологічний модуль підтримки верифікації ПС

3.3.1. Методи верифікації ПС під час розроблення

Для контролю якості програмних продуктів (включаючи всі робочі продукти) призначені процеси верифікації і валідації [2].

Мета верифікації - досліджуючи трансформації одних (вхідних) робочих продуктів в інші (вихідні) робочі продукти перевірити, чи правильно розробляється програмний продукт (наприклад, чи правильний код програмного модуля по відношенню до специфікації проекту цього модуля).

Мета валідації – досліджуючи сукупність робочих продуктів, отриманих на певному етапі процесу розроблення, пересвідчитися в тому, що вони розроблені правильно, тобто відповідають призначенню і вихідним вимогам до програмного продукту (наприклад, сукупність програмних модулів представляє необхідний програмний продукт).

Ці процеси тісно взаємопов'язані і часто визначаються одним терміном «верифікація і валідація», якому відповідає термін «Verification and Validation» (V&V), що використовується в зарубіжній літературі.

Рекомендовані для застосування методи V&V, зокрема методи наскрізного контролю, колегіальної перевірки та інспекції описані у [2]. Форми, призначені для використання при проведенні верифікації, подано у додатку Б.

Для програмних застосувань системи, які розробляються у середовищі Oracle Designer’2000, для верифікації додатково використовуються дані репозиторію Oracle. Доступ до них забезпечується за допомогою таких утиліт:

– матричного діаграмера (Matrix Diagrammer), який застосовується для перехресної перевірки, тобто відстеження зв’язків між будь-якими двома елементами репозитарія та їх відображення у зручній табличній формі,

– підсистеми побудови звітів по репозиторію (Repository Reports), який застосовується для отримання до 100 різноманітних звітів стосовно об’єктів репозиторію та їх зв’язків (рис. 3.13).

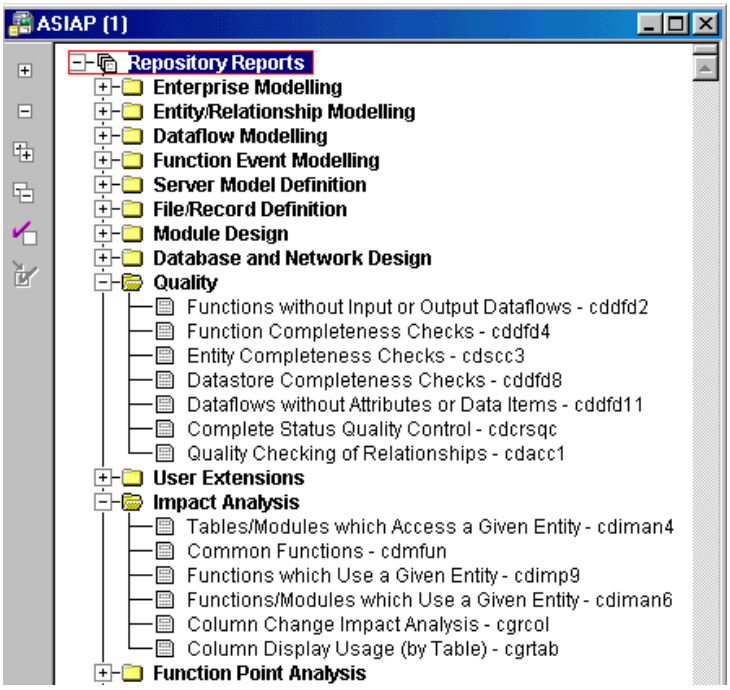


Рис.3.13. Групи звітів Repository Reports

Для класифікації дефектів запропоновано використовувати відому методологію ODC (Orthogonal Defects classification), згідно з якою для кожного виявленого дефекту визначаються [2, 38]:

– тип дефекту. Призначається виходячи з характеру дій, які повинні бути виконані для усунення дефекту (виправлення документації, інтерфейсів, функцій, алгоритмів, значень тощо). Служить мірою досконалості продукту;

– тригер дефекту (причина виявлення). Напрям перевірки (в рамках одного з трьох видів перевірки - огляди, тестування, випробування), при виконанні якої був виявлений дефект (перевірка повноти, узгодженості, інтерфейсів, використання пам’яті тощо). Служить мірою ефективності V&V;

– вплив дефекту. Критерії оцінки наслідків, які дефект матиме з позицій забезпечуваної та експлуатаційної якості ПС (тобто наслідки для функціональності, надійності, зручності застосування тощо). Служить мірою якості продукту при використанні.

3.3.2. Функції технологічного модуля підтримки верифікації

Технологічний модуль підтримки верифікації забезпечує збір та оброблення інформації про дефекти в ході перевірки ПрЗ верифікатором протягом всього періоду виконання проекту. Менеджер використовує ТМ для отримання підсумкової інформації стосовно виявлення та усунення дефектів та визначення ефективності процесу верифікації. Склад технологічного модуля описано в табл. 3.10.

Таблиця 3.10

Склад ТМ "Підтримка процесу верифікації"

Складові ТМ	Опис складових ТМ підтримки процесу верифікації
Базовий метод	Наскрізний контроль, колегіальна перевірка [2, глава 5]
Інформаційна підтримка	СКБД Oracle, шаблони для реєстрації дефектів, формування підсумкових звітів про дефекти, планування перевірок, збору даних про виконання перевірок .
Інструментальна підтримка	Середовище VBA, API-функції для роботи з репозитарієм Oracle Designer'2000, утиліта Oracle Repository Reports.
Вхід	Плани перевірок в період виконання поточного етапу проекту.
Вихід	Дані про виявлені дефекти. Дані про хід перевірок (виконання планів, витрачений час тощо)
Операції	1. Збір даних про дефекти. 2. Реєстрація дефектів за шаблонами форм та їх збереження у БД. 3. Отримання регламентованих звітів про дефекти. 4. Отримання регламентованих звітів про виконання перевірок.

Реєстрація дефектів виконується на листі "Дефекти" екземпляра програмного комплексу, встановленого на комп'ютері верифікатора (рис. 3.14).

Коди класифікації дефектів за методом ODC отримуються при натисненні кнопки "Класифікація ODC". Крім того, дефекти класифікуються:

– за рівнем серйозності (серйозні, помірні, не суттєві);

– джерелом помилок (замовник, користувач, аналітик, розробник);

– станом оброблення дефекту (переданий, усунений, відкритий).

Для кожного дефекту вказується дата виявлення та орієнтований час на усунення.

Описи дефектів експортуються до БД за допомогою розроблених універсальних алгоритмів завантаження даних з форм Excel у середовище БД Oracle з використанням механізмів ADO-інтерфейсу.

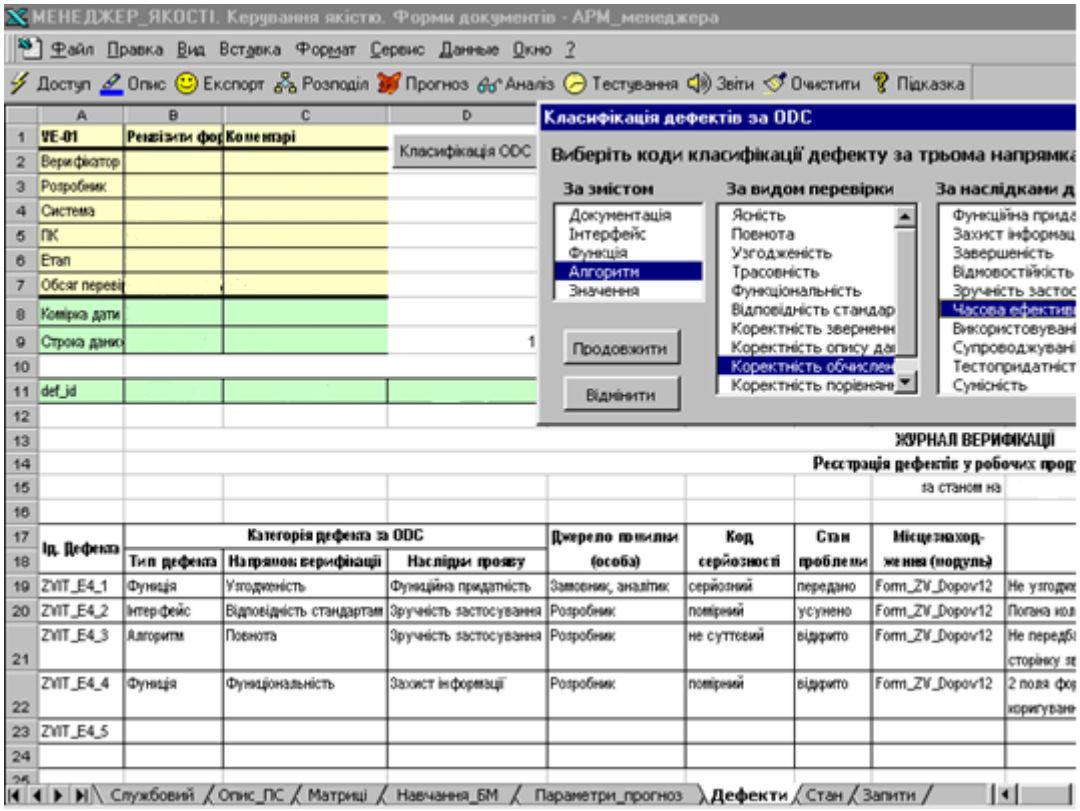


Рис. 3.14. Форма “Журнал верифікації. Реєстрація дефектів”

На листі “Запити” надаються такі види звітів:

– зведений перелік дефектів для кожного ПрЗ у розрізі:

– рівнів серйозності дефектів;

– витоків (джерел) вад та помилок;

– станів оброблення дефектів;

– типів модулів (форми, меню, звіти);

– кожному модулю ПрЗ;

– зведений перелік дефектів у ПС (кількість та середній час усунення дефектів різних категорій);

– зведений звіт про верифікацію кожного ПрЗ та ПС загалом із зазначенням метрик верифікації.

3.3.3. Метрики процесу верифікації

Верифікація ПС виконується призначеними верифікаторами за відповідними планами. Дані планування та ходу виконання верифікації зберігаються у БД. Для контролю виконання процесу верифікації та оцінки його ефективності пропонуються метрики, подані у табл. 3.11.

Таблиця 3.11

Метрики процесу верифікації

Метрика	Позначення	Опис
Кількість запланованих перевірок	$К_п$	Кількість перевірок, визначених у плані роботи верифікатора ПрЗ на поточний етап робіт
Кількість виконаних перевірок	$К_ф$	Кількість фактично проведених перевірок верифікатором ПрЗ на поточному етапі робіт
Рівень виконання	$К_{вип} = \frac{К_п}{К_ф}$	Відношення кількості запланованих перевірок до кількості фактично проведених
Середній час на підготовку до перевірки	$Т_п$	Середній час на ознайомлення з ПрЗ, опитування розробника, навчання (на етапі)
Середній час перевірки	$Т_в$	Середній час, потрібний верифікатору для виконання перевірки версії ПрЗ на етапі робіт
Сумарний час зайнятості верифікатора	$Т_{вик} = Т_п + Т_в$	Загальні витрати часу верифікатора на проведення перевірки версії ПрЗ протягом етапу
Інтенсивність перевірок ПрЗ верифікатором	$I_{вип} = \frac{V_{прз}}{T_{вик}}$	Відношення об’єму матеріала для перевірки (розміру ПрЗ) до сумарного часу зайнятості верифікатора
Загальна кількість дефектів, виявлених на етапі робіт	$Д_{ет}$	Загальна кількість дефектів, виявлених верифікатором ПрЗ у поточній версії на етапі робіт
Відносна ефективність верифікації	$B_{еф} = \frac{Д_{ет} / T_{вик}}{Д_{тест} / T_{тест}}$	Відношення кількості дефектів, виявлених в одиницю часу верифікації, до кількості дефектів ($Д_{тест}$), виявлених в одиницю часу тестування ($Т_{тест}$)

Значення характеристик процесу верифікації, розраховані за поданими у табл. 3.10 метриками, використовуються для побудови звітів про процес на листі “Запити” та прийняття рішень щодо доцільності його вдосконалення.

3.4. Експериментальні дослідження засобів автоматизації прогнозування якості програмних систем

Випробування розроблених моделей та методів виконувалися у системі інформаційно-аналітичної підтримки прийняття управлінських рішень, яка складалася з 13 програмних комплексів, об'єднаних бізнес процесом, та призначалася для функціонування у розподіленому середовищі Oracle RunTime, Oracle Express (OLAP) та MS Office (Word, Excel). Фрагмент архітектурно-функціональної схеми ПС подано на рис. 3.15.

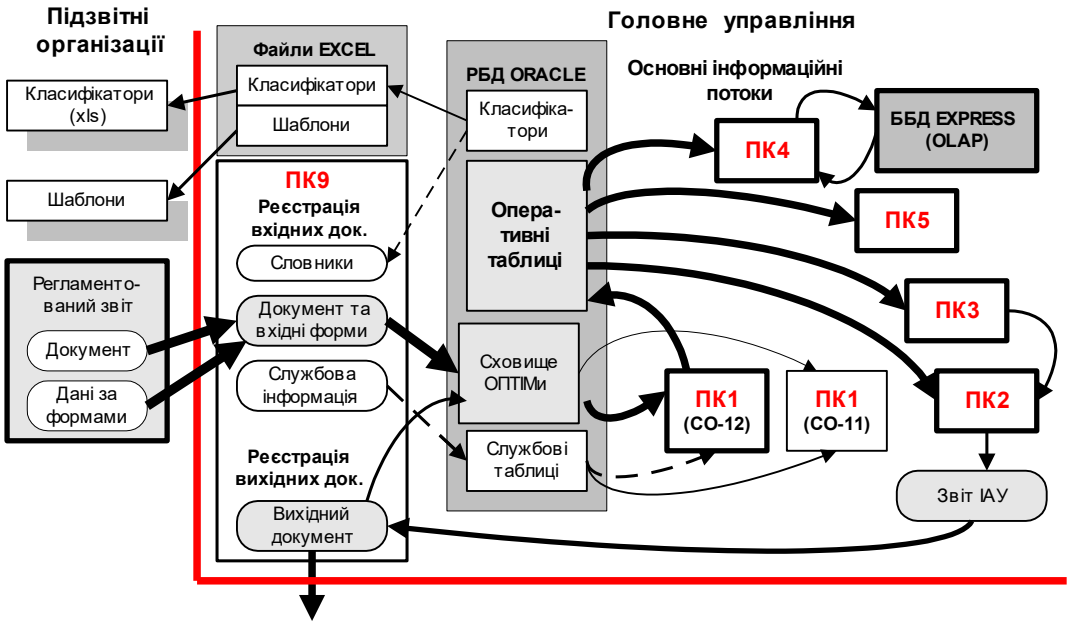


Рис. 4.1. Архітектурно-функціональна схема досліджуваної ПС

Для виконання проекту ПС була розроблена Програма забезпечення надійності (ПЗН), в рамках якої збиралися дані про середовище розроблення та експлуатації ПС, відмови та дефекти в ПС [26].

3.5. Випробування моделі розподілу надійності

Для випробовування моделі розподілу надійності з множини функцій системи були вибрані 4 функції.

Ієрархічна структура досліджуваної ПС подана на рис. 3.16, а опис її елементів – функцій, програмних застосувань та модулів - у табл. 3.12, табл. 3.13 та табл. 3.14, відповідно.

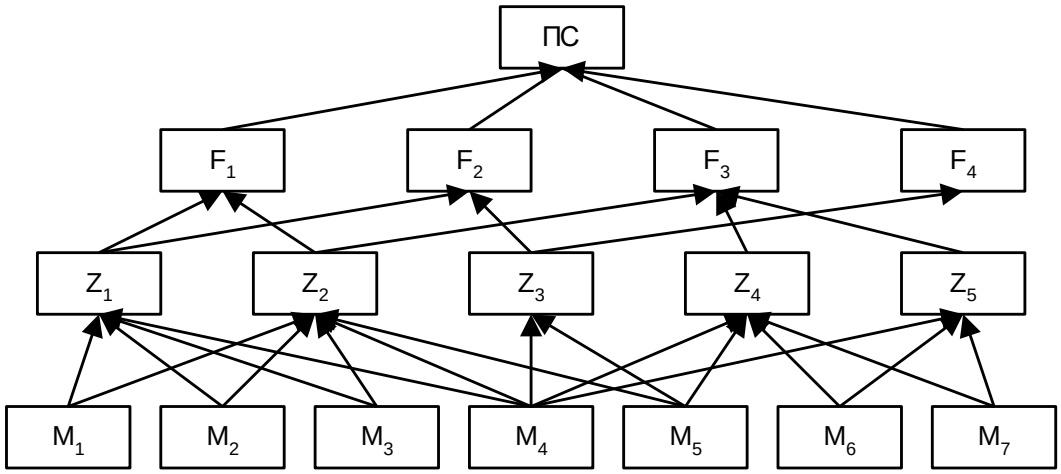


Рис. 3.16. Ієрархічна структура ПС

Таблиця 3.12

Функції ПС

Код функції	Опис функції
F1	Прийом інформації, що надходить до органу управління по різних каналах і її реєстрація
F2	Збереження інформації у БД (вхідної, отриманої та підготовленої звітної інформації)
F3	Аналітична обробка та узагальнення інформації, підготовка довідок і звітів
F4	Ведення та підтримка програмно-інформаційного середовища системи

Таблиця 3.13

Програмні застосування (програмні комплекси) ПС

Код ПрЗ	Призначення	Коди функцій
Z1	Ведення діловодства і контролю виконавської діяльності	F1, F2
Z2	Контроль і введення регламентованої інформації до БД	F1, F2, F3
Z3	Підтримка діяльності адміністраторів системи та захисту інформації	F2, F4
Z4	Надання довільних та регламентованих довідок	F3
Z5	Формування звітних тижневих, квартальних, піврічних та річних доповідей	F3

Таблиця 3.14

Модульний склад програмних застосувань ПС

Код модуля	Призначення	Коди ПЗ
М1	Реєстрація надходження інформації	Z1, Z2
М2	Контроль стану надходження та оброблення інформації	Z1, Z2
М3	Завантаження інформації до БД з форм Excel	Z1, Z2
М4	Підключення до БД	Z1, Z2, Z3, Z4, Z5
М5	Контроль та розподілення доступу до інформації	Z2, Z3, Z4
М6	Побудова (опис) критерію пошуку інформації у БД	Z4, Z5
М7	Вивантаження інформації із БД на екран та форми Word	Z4, Z5

Оскільки ПС розроблялася у середовищі Oracle, поняття незалежного модуля пов’язувалось з програмною конструкцією модуль-форма. Модуль-форма інкапсулює в собі сукупність програмних блоків, які зв’язані між собою по управлінню та даним. Приклад модуля-форми поданий на рис. 3.17.

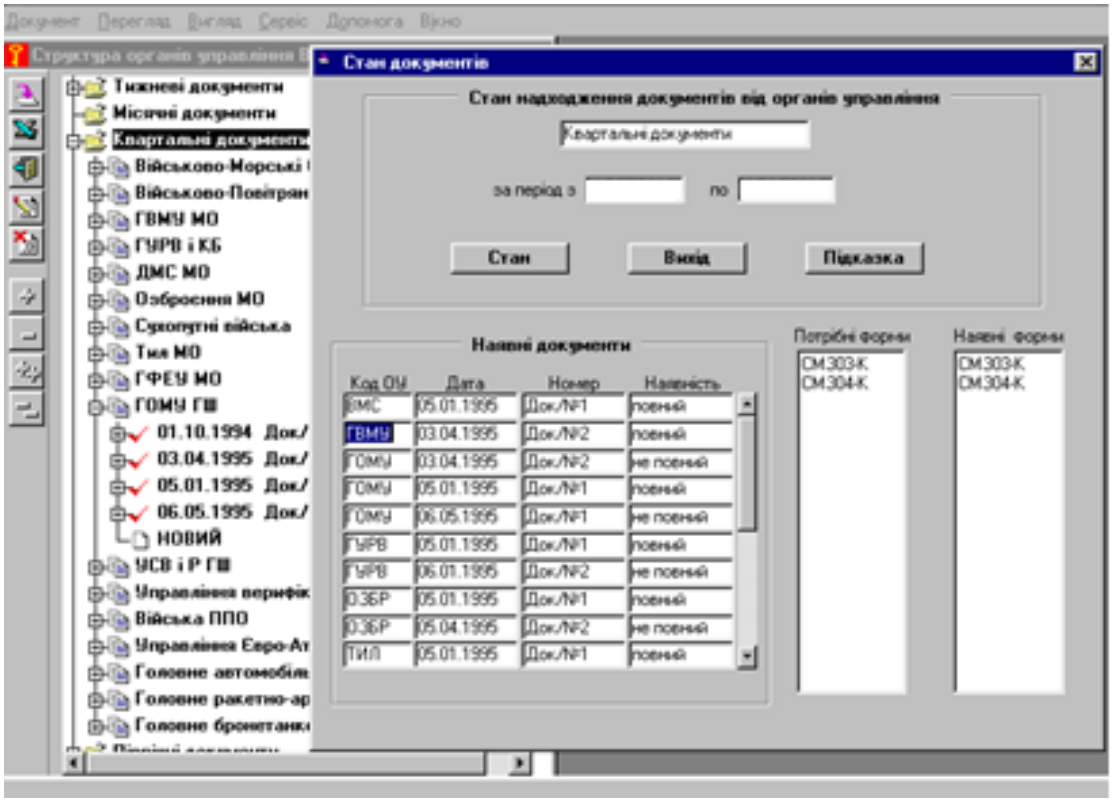


Рис. 3.17. Модуль контролю стану опрацювання документів

Встановлена ціна розроблення ПС G = 300 тис. грн, собівартість розроблення – C = 230 тис. грн, частка прибутку - δ = 0.3 (30%).

Результати порівняння структурних об’єктів ПС наведені на рис. 3.18.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Порівняння функцій відносно Qps						Порівняння Пр3									
2	по Qps	F1	F2	F3	F4		по F1	Z1	Z2			по F3	Z2	Z4	Z5	
3	F1	1,000	2,000	4,000	5,000		Z1	1,000	4,000			Z2	1,000	0,500	0,200	
4	F2	0,500	1,000	2,000	3,000		Z2	0,250	1,000			Z4	0,333	1,000	5,000	
5	F3	0,250	0,333	1,000	6,000							Z5	5,000	4,000	1,000	
6	F4	0,500	0,333	0,167	1,000		по F2	Z1	Z2	Z3						
7							Z1	1,000	2,000	4,000		по F4	Z3			
8							Z2	0,500	1,000	5,000		Z3	1,000			
9							Z3	0,250	0,200	1,000						
10																
11	Порівняння модулів															
12	по Z1	M1	M2	M3	M4		по Z2	M1	M2	M3	M4	M5		по Z3	M4	M5
13	M1	1,000	3,000	0,200	2,000		M1	1,000	2,000	1,000	3,000	0,500		M4	1,000	0,250
14	M2	0,333	1,000	0,333	3,000		M2	0,500	1,000	2,000	0,300	3,000		M5	4,000	1,000
15	M3	5,000	3,003	1,000	2,000		M3	1,000	0,500	1,000	2,000	4,000				
16	M4	0,500	0,333	0,500	1,000		M4	0,333	3,333	0,500	1,000	6,000				
17							M5	2,000	0,333	0,250	0,167	1,000				
18																
19	по Z4	M5	M6	M7		по Z5	M4	M6	M7							
20	M5	1,000	2,000	2,000		M4	1,000	2,000	0,333							
21	M6	0,500	1,000	5,000		M6	0,500	1,000	3,000							
22	M7	0,500	0,200	1,000		M7	3,000	0,333	1,000							
23																

Рис. 3.18. Матриці порівнянь об’єктів системи

Вектор відносної ваги порівнюваних функцій ПС $F_1, \ldots, F_4$:

$(u_1, u_2, \ldots, u_4)$ = (*0.425 0.230 0.274 0.071*).

Узагальнені вагові коефіцієнти, які відображають відносну важливість надійної роботи модулів $M_1, \ldots, M_7$ у функціонуванні ПС:

$(w_1^* \quad w_2^* \quad \ldots \quad w_7^*)$ = (*0.163 0.133 0.244 0.197 0.174 0.050 0.08*).

Узагальнені вагові коефіцієнти, які відбивають відносну важливість надійної роботи ПрЗ $Z_1, \ldots, Z_5$ у функціонуванні ПС:

$(v_1^* \quad v_2^* \quad \ldots \quad v_5^*)$ = (*0.364 0.349 0.161 0.029 0.097*).

Задача оптимізації – знайти такий модульний розподіл надійності $(r_1, \ldots r_7)$, який максимізує функцію корисності:

$$Q_{nc} = 0{,}364 \cdot (r_1 \cdot r_2 \cdot r_3 \cdot r_4) + 0{,}349 \cdot (r_1 \cdot r_2 \cdot r_3 \cdot r_4 \cdot r_5) + 0{,}161 \cdot (r_4 \cdot r_5) +$$

$$+ 0{,}029 \cdot (r_5 \cdot r_6 \cdot r_7) + 0{,}097 \cdot (r_4 \cdot r_6 \cdot r_7)$$

Задача вирішувалась за таких обмежень

$15+25r_1 \le 39.12$

$12+21r_2 \le 31.92$

$26+34r_3 \le 58.56$

$10+38r_4 \le 47.52$

$20+22r_5 \le 41.47$

$2+8r_6 \le 12$

$6+12r_7 \le 19.2$

$91 + 25r_1 + 21r_2 + 34r_3 + 38r_4 + 22r_5 + 8r_6 + 12r_7 \le 250$

$0.6 \le r_1 \le 0.99$; $0.6 \le r_2 \le 0.99$; $0.6 \le r_3 \le 0.99$; $0.6 \le r_4 \le 0.99$; $0.6 \le r_5 \le 0.99$;

$0.6 \le r_6 \le 0.99$; $0.6 \le r_6 \le 0.99$;

Розв'язок задачі виконувався у середовищі MATLAB із застосуванням функції *fmincon* пакету оптимізації:

[x,f,flag,output]=fmincon(@minfun1,x0,A,B,[],[],lb,ub)

де *minfun1* – ім'я функції, яка обчислює цільову функцію Q_{nc}, параметр *x0* – задає початкові наближення для r_i, A, B - матриця та вектор лінійних обмежень виду

```
A=[25  0   0  0  0   0   0
    0  21  0  0  0   0   0
    0  0  34  0  0   0   0
    0  0  0  38  0   0   0
    0  0  0   0  22  0   0
    0  0  0   0   0   8   0
    0  0  0   0   0   0  12
   25 21 34 38 22 8  12]
B=[24.12; 19.92; 32.56; 37.52; 21.76; 10; 13.2; 152]
```

Отримане значення $MaxQ_{nc} = 0.85$. У табл. 3.12 підсумовані параметри та розв'язок задачі.

Таблиця 3.12

Параметри та результати розв'язку задачі оптимізації

Модулі	w_i^*	c_i **тис. грн**	d_i **тис. грн**	α_i	β_i	r_i
M1	0.163	15	25	0.6	0.99	0.65
M2	0.133	12	21	0.6	0.99	0.49
M3	0.244	236	35	0.6	0.99	0.958
M4	0.197	10	38	0.6	0.99	0,987
M5	0.174	20	22	0.6	0.99	0.989
M6	0.050	2	8	0.6	0.99	0.965
M7	0.080	6	12	0.6	0.99	0.70

У графічній формі розподіл надійності по модулях подано на рис. 3.19.

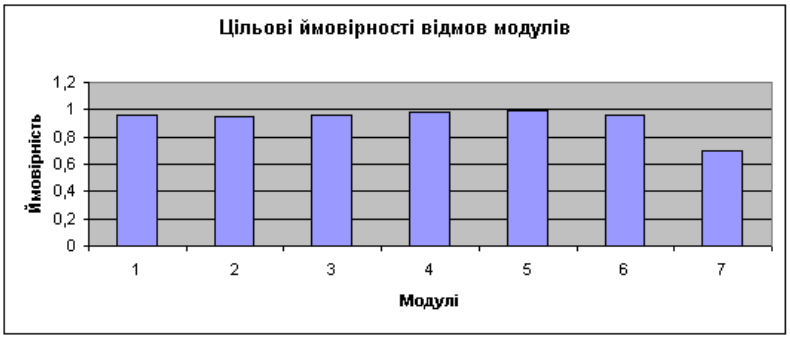


Рис. 3.19. Стовпчикова діаграма розподілу надійності по модулях ПС

Розраховані цільові надійності $q_1, \ldots, q_5$ програмних застосувань $Z_1, \ldots, Z_5$: *0.866, 0.856, 0.976, 0.665, 0.663*

У графічній формі розподіл цільових значень надійності по ПрЗ подано на рис. 3.20.

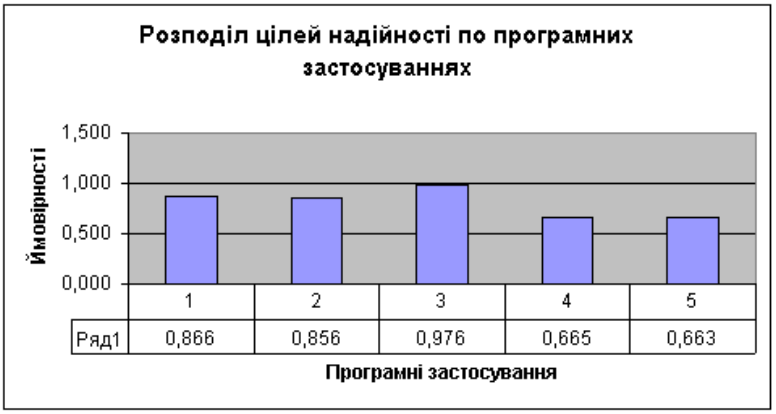


Рис. 3.20. Діаграма розподілу надійності по ПрЗ

Отриманий розподіл надійності був схвалений користувачами ПС та її менеджером, оскільки враховував передбачуваний профіль експлуатації ПС, а також ресурсні обмеження проекту. Розподілені значення надійності ПрЗ були зафіксовані у ПЗН як цільові вимоги до надійності компонентів ПС.

3.6. Висновки

У даному розділі одержано наступні практичні результати:

1. Охарактеризовано функції доповненого програмного комплексу, який підтримує виконання процесів керування якістю та верифікації ПС.
2. Подано стислий опис методів і алгоритмів реалізації запропонованих моделей інженерії якості, а також збору та класифікації даних про дефекти у ПрЗ та хід виконання процесів розроблення та верифікації ПрЗ.
3. Проведено експериментальні дослідження розробленого методу та моделей прогнозування та верифікації якості програмного комплексу управління організаційними процесами підприємства, що підтвердило їх ефективність.

РОЗДІЛ 4
ОБГРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ

Темою дипломної роботи магістра є дослідження моделей та методів підтримки прийняття рішень при прогнозуванні якості програмних систем на ранніх стадіях життєвого циклу. Важливим аспектом впровадження моделей та методів щодо прийняття рішень при прогнозуванні якості програмних систем є їх економічна доцільність.

4.1. Розрахунок норм часу на виконання науково-дослідної роботи

Основні етапи виконання НДР можна визначити наступним чином:

– визначення актуальності теми ДР магістра;

– розробка моделі та методу підтримки прийняття рішень щодо прогнозування якості програмних систем на ранніх стадіях життєвого циклу;

– проведення аналізу існуючих підходів підтримки прийняття рішень щодо якості програмних систем;

– проектування алгоритму роботи системи підтримки прийняття рішень;

– розробка архітектури системи підтримки прийняття рішень;

– створення інструкції з реалізації програмного комплексу;

– оформлення інструкцій.

Дані витрат часу на виконання окремих стадій (етапів) можна звести у табл. 4.1.

Таблиця 4.1

Основні етапи виконання НДР

№ та назва етапу	**Середній час виконання стадії інженером, год.**
1 Визначення актуальності теми ДР магістра	12

Продовження таблиці 4.1

№ та назва етапу	**Середній час виконання стадії інженером, год.**
2 Розробка моделі та методу підтримки прийняття рішень щодо прогнозування якості програмних систем на ранніх стадіях життєвого циклу	150
3 Проведення аналізу існуючих підходів підтримки прийняття рішень щодо якості програмних систем	40
4 Проектування алгоритму роботи системи підтримки прийняття рішень	30
5 Розробка архітектури системи підтримки прийняття рішень	64
6 Створення інструкції з реалізації програмного комплексу	30
7 Оформлення інструкцій	28
Разом	354

Витрати часу керівника на виконання окремих стадій (етапів) при недостатній кількості інформації доцільно приймати в межах 5% сумарних витрат часу інженерів на виконання цих стадій (етапів).

4.2. Розрахунок витрат на проведення НДР

Розрахунок поточних витрат на проведення НДР проводять в розрізі таких калькуляційних статей:

- основна заробітна плата (з/п);
- додаткова (з/п);
- нарахування на (з/п);

- косультаційні витрати;
- матеріали для виконання робіт по НДР ;
- експериментально-виробничі витрати;
- загальновиробничі витрати;
- адміністративні витрати;
- позавиробничі витрати.

При системному розв'язанні питання про облік праці і заробітної плати велике значення має умовно-постійна (нормативна, довідкова та інша) інформація, яка в даному разі характеризує переважно постійних виконавців (людей і механізми) та постійні процеси (технологічні операції). Тому у першу чергу зміст по обліку праці і заробітної плати неодмінно повинна входити інформація про виконавців (облік складу працівників).

Основна з/п складається із прямої з/п і доплати, яка при укрупнених розрахунках становить 25% - 35% від прямої з/п. При розрахунку з/п кількість робочих днів в місяці слід приймати — 25,4 дні/міс., що відповідає 203,2 год./міс. Приймемо розмір місячного окладу інженера 1800 грн..

Пряма з/п визначається:

$$ЗП = O_i * T_i / 203{,}2, \tag{4.1}$$

де O_i - розмір місячних окладів і -х категорій працівників;

T_i -трудомісткість робіт виконаних працівниками і-х категорій.

Для інженера: $ЗП = 1800 * 354 / 203{,}2 = 3135{,}83\ грн.$;

Величина доплат обраховується за формулою:

$$ЗП_1 = ЗП * K_i \tag{4.2}$$

де K_i - коефіцієнт доплат (0,25 - 0,35).

Вибираємо коефіцієнт 0,25:

Для інженера: $ЗП_1 = 3135,83 * 0,25 = 783,96\ грн.$;

Основна з/п обчислюється за формулою:

$$ЗП_О = ЗП + ЗП_1 \quad (4.3)$$

Для інженера:

$ЗП_О = 3135,83 + 783,96 = 3919,79\ грн.$.

Величина додаткової з/п обчислюється за формулою:

$$ЗП_Д = ЗП_О * К_Д, \quad (4.4)$$

де $К_Д$ - коефіцієнт додаткової з/п (0,05 - 0,1).

Нехай коефіцієнт додаткової $К_Д = 0,1$.

Для інженера додаткова плата становить:

$ЗП_Д = 3919,79 * 0,1 = 391,98\ грн.$,

Витрати, на проведення НДР, крім річного фонду заробітної плати, включають ще й соціальні нарахування. Всього норматив нарахувань на заробітну плату становить 37% .

Загальний норматив нарахувань на заробітну плату розраховується за формулою:

$$З_Н = (ЗП_О + ЗП_Д) * К_Н, \quad (4.5)$$

де $ЗП_О$ - величина основної заробітної плати;

$ЗП_Д$ - величина додаткової заробітної плати;

$К_Н$ - загальний відсоток нарахувань на заробітну плату.

Для інженера загальний норматив нарахувань становить:

$З_Н = (3919,79 + 391,98) * 0,37 = 1595,35\ грн.$

Таблиця 4.2

Зведена відомість витрат на заробітну плату, грн.

№ п/п	Категорія працівників	Основна заробітна плата			Додаткова заробітна	Нарахування на заробітну плату	Всього витрати на заробітну плату
		Пряма заробітна	Доплати	Всього:			
1	Інженер	3135,83	783,96	3919,79	391,98	1595,35	5907,12

Для розрахунку витрат на консультації, врахуємо, що консультації були надані в обсязі 4 год., вартість їх 360 грн.

Витрати на матеріали розраховуються на основі норм їх витрат і відповідних оптових цін:

$$M_З = \sum_{i=1}^{П} H_{мі} * Ц_{оі} \qquad (4.6)$$

де $M_З$ – затрати на матеріали;

$H_{мі}$ – норма затрат і-их матеріалів;

$Ц_{оі}$ – оптова ціна за одиницю витрат і-их матеріалів;

Таблиця 4.3

Визначення величини матеріальних витрат

Найменування матеріальних ресурсів	**Одиниця виміру**	**Норма витрат**	**Ціна за одиницю, грн**	**Затрати матеріалів, грн**	**Транспортно-заготівельні витрати, грн.**	**Загальна сума витрат на матеріали, грн.**
1 Основні матеріали						
Середовище MS Visio 2010	шт.	1	6000	6000	600	6600

Продовження таблиці 4.3

Найменування матеріальних ресурсів	**Одиниця виміру**	**Норма витрат**	**Ціна за одиницю, грн**	**Затрати матеріа-лів, грн**	**Транспортно-заготіве-льні витрати, грн.**	**Загальна сума витрат на матеріали, грн.**
2 Додаткові матеріали						
Ручка	шт.	2	10	20	2	22,00
Олівець	шт.	3	2	6	0,6	6,60
Гумка	шт.	1	2	2	0,2	2,20
Компакт диски	шт.	2	4	8	0,8	8,80
Разом						6639,60

Експериментально-виробничі витрати визначаються як витрати на машинний час, який є потрібним для виконання необхідного об'єму робіт виходячи з його вартості за одиницю часу, тобто:

$$З_{Е.В.} = В_Р * Т, \quad (4.7)$$

де $З_{Е.В.}$ - затрати експериментально-виробничі;

$В_Р$ - витрати на користування ПК та послуги інтернет;

$Т$ - час роботи ПК.

Вартість роботи на ПЕОМ і користування мережею Інтернет встановлюємо виходячи з реальних даних (5 грн./год.). Оскільки, інтернет та ПК використовувався на усіх стадіях, то експериментально-виробничі затрати становлять:

$З_{Е.В.} = 5 * 354 = 1770{,}00$ *грн.*

Загальновиробничі витрати при укрупнених розрахунках приймаємо на рівні 70% - 90% від суми основної і додаткової з/п інженерів, яка була нарахована за роботу при проведенні НДР, тобто:

$$З_{З.В.} = (ЗП_О + ЗП_Д) * К_З \quad (4.8)$$

де $З_{З.В.}$ - загально-виробничі затрати;

$ЗП_О$ - основна заробітна плата;

$ЗП_Д$ - додаткова заробітна плата;

$К_З$ - коефіцієнт загальновиробничих затрат.

В даному випадку приймемо коефіцієнт загально-виробничих затрат на рівні 80%, тоді сума затрат становитиме:

$З_{З.В.} = (3919,79 + 391,98) * 0,8 = 3449,42$ *грн.*

Аналогічно визначаються адміністративні витрати, які доцільно приймати на рівні 50% - 60% від суми основної і додаткової з/п інженерів.

$$З_{З.А.} = (ЗП_О + ЗП_Д) * К_А, \quad (4.9)$$

де $З_{З.А.}$ - адміністративні витрати;

$ЗП_О$ - величина основної заробітної плати;

$ЗП_Д$ - величина додаткової заробітної плати;

$К_А$ - коефіцієнт адміністративних витрат.

Нехай коефіцієнт адміністративних витрат становить 50%, то величина адміністративних витрат буде рівна:

$З_{З.А.} = (3919,79 + 391,98) * 0,5 = 2155,89$ *грн.*

Позавиробничі витрати слід приймати на рівні 3% - 7% від виробничої собівартості. Виробнича собівартість включає в себе основну заробітну плату,

додаткову заробітну плату, нарахування на заробітну плату, консультації, матеріали, експериментально-виробничі та загально-виробничі витрати, тобто:

$$B_{П.В.} = (ЗП_О + ЗП_Д + З_Н + М_З + З_{Е.В.} + З_{З.В.}) * К_{П.В.} \quad (4.10)$$

де $B_{П.В.}$ - позавиробничі витрати;

$К_{П.В.}$ - коефіцієнт позавиробничих витрат.

В даному випадку $К_{П.В.} = 5\%$, тоді позавиробничі витрати становитимуть:

$$B_{П.В.} = (3919,79 + 391,98 + 1595,35 + 6639,60 + 1770,00 + 3449,42 + 360) * 0,05 = 906,31 \text{ грн.}$$

Розрахунок поточних витрат зводиться в таблицю 4.4:

Таблиця 4.4

Калькуляція собівартості проведення НДР

Статті витрати, грн.	**Витрати, грн.**	**В % до загальної суми**
Основна заробітна плата	3919,79	18,50%
Додаткова заробітна плата	391,98	1,85%
Нарахування на заробітну плату	1595,35	7,53%
Консультації	360	1,70%
Матеріали	6639,60	31,34%
Експериментально-виробничі витрати	1770	8,35%
Загальновиробничі витрати	3449,42	16,28%
Разом виробнича собівартість	18126,14	85,55%
Адміністративні витрати	2155,89	10,17%

Продовження таблиці 4.4

Статті витрати, грн.	**Витрати, грн.**	**В % до загальної суми**
Позавиробничі витрати	906,31	4,28%
Повна собівартість	21188,34	100,00%

Заключною частиною роботи на цій ділянці є показники, що необхідні для встановлення собівартості, проведення комплексного економічного аналізу затрат праці і нарахованої заробітної плати.

4.3. Розрахунок ціни НДР і економічна ефективність від використання методу підтримки прийняття рішень при прогнозуванні якості програмних систем

Ціну НДР можна визначити [13]:

$$Ц = (C_{пр} / N + C_{коп}) + П, \qquad (4.11)$$

де $C_{пр}$ - собівартість НДР, грн.;

N - кількість замовлень, од.;

$C_{коп}$ - собівартість копіювання (ксерокопії, дискети, компакт-диски, поштові витрати, відрядження спеціалістів для запуску та наладки програмного забезпечення тощо), грн.;

$П$ - нормативна величина прибутку (15% - 30% від собівартості $C_{пр}$).

Кількість замовлень при проведенні НДР: $N = 1$, собівартість копіювання становить: $C_{коп} = 100\,грн.$ Оцінка економічної ефективності розробки НДР при створенні програмної системи рівна:

$Ц = (21188,34/1+100) + 21188,34 * 0,2 = 25526,01 грн.$

Економічна ефективність від використання НДР зумовлена:

– скороченням трудовитрат при виконанні певних завдань;

– скороченням машинного часу при виконанні певних завдань.

При визначенні економічної ефективності необхідно порівняти використовуваний (базовий) програмний продукт і пропонований. З допомогою відповідних розрахунків (в разі значної складності використання експертних оцінок) визначається скорочення трудовитрат і (або) машинного часу, і як наслідок – економія коштів при використанні нового програмного продукту.

Для визначення ефективності НДР розраховують чисту приведену цінність NPV і термін окупності T_{OK} [13]:

$$NPV = \sum((Д_t - B_t)/(1+i)^t), \quad (4.12)$$

де $Д_t$ - повний дохід за рік t при використанні системи підтримки прийняття рішень;

B_t - повні витрати за рік t при використанні системи підтримки прийняття рішень;

t - відповідний рік проекту;

i - дисконтна ставка (0,3).

Нехай повний дохід за рік при використанні методу підтримки прийняття рішень при прогнозуванні якості програмних систем – 20000 грн., а витрати на обслуговування – 5000 грн. тоді чиста приведена цінність:

$$NPV \sum_{t=1}^{2}((20000-5000)/(1+0,3)^t) = 20414,20 \ грн.$$

Термін окупності визначається за формулою [13]:

$$T_{OK} = Ц / \sum(Д_t/(1+i)^t), \quad (4.13)$$

Термін окупності методу підтримки прийняття рішень при прогрнозуваннія якості програмних систем становить:

$$T_{ОК} = 25526{,}01 / 20414{,}20 /(1+0{,}3)^{1}) + 25526{,}01 / 20414{,}20 /(1+0{,}3)^{2}) = 1{,}7 \; року.$$

Таблиця 4.5

Основні показники ефективності

№ п/п	Назва показника	Один, вимір.	Величина
1	Витрати часу на розробку моделі та методу підтримки прийняття рішень при прогнозуванні якості програмних систем на ранніх стадіях життєвого циклу	год.	354
2	Витрати на розробку моделі і методу підтримки прийняття рішень при прогнозуванні якості програмних систем на ранніх стадіях життєвого циклу	грн.	21188,34
3	Кількість покупців системи	од.	1,00
4	Ціна розробки моделі і методу підтримки прийняття рішень при прогнозуванні якості програмних систем на ранніх стадіях життєвого циклу	грн.	25526,01
5	Чиста приведена цінність моделі і методу підтримки прийняття рішень при прогнозуванні якості програмних систем на ранніх стадіях життєвого циклу	грн.	20414,01
6	Термін окупності витрат по НДР	рік	1,7

Отже, собівартість моделі, методу і системи підтримки прийняття рішень при прогнозуванні якості програмних систем на ранніх стадіях життєвого циклу складає 25526,01грн. Термін окупності НДР становить 1,7 року, що забезпечує економічну доцільність впровадження результатів науково-дослідної роботи.

РОЗДІЛ 5
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1. Охорона праці

Метою дипломної роботи магістра є розробка методу і засобу прогнозування якості програмних систем на ранніх стадіях життєвого циклу. Оскільки, проведення робіт з прогнозування якості програмного забезпечення передбачає використання комп’ютерної техніки, зокрема ПК та периферійних пристроїв, то обов’язком виконавця такого процесу є забезпечення оптимальних умов праці з охорони праці і техніки безпеки.

Оскільки, в якості виконавця зазвичай виступає колектив працівників деякої фірми, то її керівництво повинно забезпечити безпечні умови праці.

Роботодавець зобов'язаний згідно Закону України «Про охорону праці» стаття 13 «Управління охороною праці та обов’язки роботодавця» створити на робочому місці в кожному структурному підрозділі умови праці відповідно до нормативно-правових актів, а також забезпечити додержання вимог законодавства щодо прав працівників у галузі охорони праці.

Із цією метою роботодавець забезпечує функціонування системи управління охороною праці, а саме:

– створює відповідні служби і призначає посадових осіб, які забезпечують вирішення конкретних питань охорони праці, затверджує інструкції про їхні обов'язки, права та відповідальність за виконання покладених на них функцій, а також контролює їх додержання;

– розробляє за участю сторін колективного договору і реалізує комплексні заходи для досягнення встановлених нормативів та підвищення існуючого рівня охорони праці;

– забезпечує виконання необхідних профілактичних заходів відповідно до обставин, що змінюються;

– впроваджує прогресивні технології, досягнення науки і техніки, засоби механізації та автоматизації виробництва, вимоги ергономіки, позитивний досвід з охорони праці тощо;

– забезпечує належне утримання будівель та споруд, виробничого обладнання та устаткування, моніторинг за їх технічним станом;

– забезпечує усунення причин, що призводять до нещасних випадків, професійних захворювань, та здійснення профілактичних заходів, визначених комісіями за підсумками розслідування цих причин;

– організовує проведення аудиту охорони праці, лабораторних досліджень умов праці, оцінку технічного стану виробничого обладнання та устаткування, атестацій робочих місць на відповідність нормативно-правовим актам з охорони праці в порядку і строки, що визначаються законодавством, та за їх підсумками вживає заходів з усунення небезпечних і шкідливих для здоров'я виробничих факторів;

– розробляє і затверджує положення, інструкції, інші акти з охорони праці, що діють у межах підприємства та встановлюють правила виконання робіт і поведінки працівників на території підприємства, у виробничих приміщеннях, на будівельних майданчиках, робочих місцях відповідно до нормативно-правових актів з охорони праці, забезпечує безоплатно працівників нормативно-правовими актами підприємства з охорони праці;

– здійснює контроль за додержанням працівником технологічних процесів, правил поведінки з машинами, механізмами, устаткуванням та іншими засобами виробництва, використанням засобів колективного та індивідуального захисту, виконанням робіт відповідно до вимог з охорони праці;

– організовує пропаганду безпечних методів праці та співробітництво з працівниками у галузі охорони праці.

Роботодавець несе безпосередню відповідальність за порушення нормативно-правових актів з охорони праці.

Для забезпечення оптимальних умов праці працівників при проведенні прогнозування якості програмних систем, необхідно передбачити відповідність мікроклімату у приміщеннях згідно вимог ДСН 3.3.6.042-99.

Категорія робіт при експлуатації засобу прогнозування якості належить до легкої – Іб.

Для того щоб визначити, чи відповідає повітряне середовище певного приміщення встановленим нормам, необхідно кількісно оцінити кожний з його параметрів. Оптимальні показники мікроклімату, які необхідно забезпечити у приміщеннях, де експлуатуються ПК у теплу пору року повинні становити: температура – 22-24 °С, відносна вологість – 40-60%, швидкість руху повітря 0,1 м/с.

Окрім, забезпечення оптимальних показників мікроклімату, необхідно передбачити ще й оптимальні показники шуму та вібрації на робочих місцях.

Граничні величини шуму на робочих місцях регламентуються ГОСТ 12.1.003-86. В ньому закладено принцип встановлення певних параметрів шуму, виходячи з класифікації приміщень та їх використання для трудової діяльності.

Робоче місце працівників, які прогнозують якість програмного забезпечення, можна прирівняти до робочих місць у приміщеннях конструкторських бюро, програмістів обчислювальних машин, лабораторій для теоретичних робіт і опрацювання експериментальних даних, прийому хворих в медпунктах і відповідно необхідно передбачити відповідні рівні звукового тиску.

Окрім цього, на робочих місцях працівників необхідно забезпечити дотримання вимог НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин». Основними вимогами, визначеними у цьому нормативному документі є:

– площу та об'єм для одного робочого місця оператора визначають згідно з вимогами ДСанПіН 3.3.2-007-98. Площа має бути не менше 6,0 кв.м, об'єм - не менше 20,0 куб.м.

– заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця операторів (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), мають бути надійно захищені діелектричними щитками або сітками з метою недопущення потрапляння працівника під напругу.

– приміщення, де розміщені робочі місця операторів, крім приміщень, у яких розміщені робочі місця операторів великих ЕОМ загального призначення (сервер), повинні бути оснащені системою автоматичної пожежної сигналізації.

При експлуатації програмної системи прогнозування якості важливим, з точки зору охорони праці, є забезпечення достатньої величини природного та штучного освітлення. Нормованим параметром природного освітлення є коефіцієнт природного освітлення (КПО). КПО встановлюється в залежності від розряду виконуваних зорових робіт.

Робота працівників, що проводять прогнозування якості ПЗ, відноситься до робіт середньої точності, що передбачає IV розряд зорових робіт з мінімальним розміром об'єктів розрізнення 0,5 – 1,0 мм. При цьому в будівлях через віконні отвори передбачається мінімальне бокове освітлення з КПО=1,5 %.

Для штучного освітлення нормованим параметром виступає $Е_{мін}$ – мінімальний рівень освітленості, та $К_п$ – коефіцієнт пульсації світлового потоку. Для забезпечення оптимальних умов праці необхідно передбачити коефіцієнт пульсації світлового потоку на рівні не більшому, ніж 20% відповідно до ДБН В.2.5-28-2006.

Оскільки, робота щодо прогнозування якості програмних систем на ранніх етапах життєвого циклу відноситься до IV розряду зорових робіт, то мінімальний рівень штучного освітлення, який необхідно передбачити складає 300...500 Лк.

Сукупність заходів щодо забезпечення охорони праці, починаючи від виконання встановлених законами України норм та правил, а також виконання правил техніки безпеки, гарантує особам, які працюють над прогнозуванням якості програмних систем, безпеку праці та нівелювання негативних факторів впливу на їх здоров’я.

5.2. Захист населення у надзвичайних ситуаціях від впливу радіації

При виникненні надзвичайної ситуації, зокрема, при аварійному викиданні в атмосферу радіоактивних речовин можливі такі види радіоактивного впливу на населення:

– зовнішнє опромінення при проходженні радіоактивної хмари;

– внутрішнє опромінення при вдиханні радіоактивних аерозолів (інгаляційна небезпека);

– контактне опромінення внаслідок радіоактивного забруднення шкіри і одягу;

– зовнішнє опромінення, зумовлене радіоактивним забрудненням поверхні землі, будівель, споруд та ін.;

– внутрішнє опромінення при використанні забруднених продуктів харчування і води.

Розрахункові дані та результати прямих вимірювань рівня радіації і дози опромінення мають бути основою для вжиття заходів захисту населення від зовнішнього і внутрішнього опромінення, в тому числі й профілактичне застосування стабільного йоду.

Основою розробки заходів захисту населення в умовах радіоактивного забруднення при ядерній аварії є рекомендації Міжнародного агентства з атомної енергії (МАГАТЕ) 1988 р., а також норми радіаційної безпеки України (НРБУ—1997).

Враховуючи рівень радіації, а також прогноз можливих аварійних викидів радіоактивних речовин та метеорологічні дані, приймається рішення про проведення таких термінових і невідкладних заходів захисту в умовах ранньої фази радіаційної аварії:

– укриття населення;

– обмеження перебування населення на відкритій місцевості;

– евакуація у разі загрози здоров'ю;

– проведення йодової профілактики;

– тимчасова заборона вживання продуктів харчування і води із зони радіоактивного забруднення.

Крім цих заходів у період ранньої і пізньої фази проводяться довгострокові заходи:

– тимчасове відселення;

– евакуація — переселення на постійне місце проживання;

– обмеження вживання води і продуктів харчування забруднених радіоактивними речовинами;

– заходи захисту при виробництві продукції тваринництва, рослинництва і лісогосподарської діяльності;

– дезактивація території і будівель;

– інші заходи: гідрологічні, протиповіневі, обмеження лісокористування, полювання, рибної ловлі, перебування у полі при проведенні сільськогосподарських робіт.

Критерієм для прийняття рішення про заходи захисту населення на ранній і середніх фазах після аварії є дози зовнішнього і внутрішнього опромінення (табл. 5.1) з установленими двома рівнями радіаційного впливу — нижнім і верхнім — згідно з рекомендацією МАГАТЕ і НРБУ—1997.

При прогнозованому опроміненні, що не перевершує нижнього рівня, заходи, перелічені в табл. 5.1 не проводяться. Якщо прогнозоване опромінення перевищує нижній рівень, але не досягає верхнього рівня, то проведення вказаних заходів може бути відкладене.

Якщо прогнозоване опромінення досягає або перевищує верхній рівень, то обов'язково необхідно проводити заходи, наведені в табл. 5.1, навіть якщо вони пов'язані з порушенням нормальної життєдіяльності населення і об'єктів.

Таблиця 5.1

Критерії для прийняття рішень на ранній фазі розвитку аварії

Захисні заходи	**Дозові критерії (прогнозована доза за перші 10 діб), мЗв**			
	Все тіло		**Окремі органи (легені, щитовидна залоза, шкіра)**	
	Нижній рівень	**Верхній рівень**	**Нижній рівень**	**Верхній рівень**
Укриття, захист органів дихання і шкіри	5	50	50	500
Йодова профілактика:				
дорослі	—	—	50*	500*
діти, вагітні жінки	—	—	50*	250*
Евакуація:				
дорослі діти, вагітні жінки	50 10	500 50	500 200*	5000 500*

Радіаційний захист населення включає в себе:

– організацію безперервного контролю, виявлення та оцінку радіаційної та хімічної обстановки в районах розміщення радіаційно-небезпечних об'єктів;

– завчасне накопичення, підтримання в готовності і використання при необхідності засобів індивідуального захисту, приладів радіаційної розвідки і контролю;

– створення, виробництво та застосування уніфікованих засобів захисту, приладів і комплектів радіаційної розвідки і дозиметричного контролю;

– придбання населенням у встановленому порядку в особисте користування засобів індивідуального захисту та контролю за використанням їх за призначенням;

– своєчасне впровадження і застосування засобів і методів виявлення та оцінки масштабів і наслідків аварій на радіаційно-небезпечних об'єктах;

– створення і використання на радіаційно-небезпечних об'єктах систем (переважно автоматизованих) контролю обстановки і локальних систем оповіщення;

– розробку і застосування, за необхідності, режимів радіаційного захисту населення і функціонування об'єктів економіки та інфраструктури в умовах забрудненості (зараженості) місцевості;

– завчасне пристосування об'єктів комунально-побутового обслуговування і транспортних підприємств для проведення спеціальної обробки одягу, майна і транспорту, проведенням цієї обробки в умовах аварій;

– навчання населення використання засобів індивідуального захисту і правилам поведінки на забрудненій (зараженої) території.

До числа основних заходів щодо захисту населення від радіаційного впливу під час радіаційної аварії, належать:

– виявлення факту радіаційної аварії та оповіщення про неї;

– виявлення радіаційної обстановки в районі аварії;

– організація радіаційного контролю;

– встановлення та підтримання режиму радіаційної безпеки;

– проведення, при необхідності, на ранній стадії аварії йодної профілактики населення, персоналу аварійного об'єкта, учасників ліквідації наслідків аварії;

– забезпечення населення, персоналу аварійного об'єкта, учасників ліквідації наслідків аварії засобами індивідуального захисту та використання цих коштів;

– укриття населення, яке опинилося в зоні аварії, в притулках і укриттях, що забезпечують зниження рівня зовнішнього опромінення і захист органів дихання від проникнення в них радіонуклідів, які опинилися в атмосферному повітрі;

– санітарна обробка населення, персоналу аварійного об'єкта, учасників ліквідації наслідків аварії;

– дезактивація аварійного об'єкта, об'єктів виробничого, соціального, житлового призначення, території, сільськогосподарських угідь, транспорту, інших технічних засобів, засобів захисту, одягу, майна, продовольства і води;

– евакуація або відселення громадян із зон, в яких рівень забруднення перевищує допустимий для проживання населення.

Висновки

Радіація на сьогодні є чи не найнебезпечнішим фактором впливу не тільки на людину, але й на усі живі організми на планеті. Неконтрольовані ядерні реакції, ядерні війни та ряд інших можливих результатів людської діяльності, пов’язаних із радіацією негативно впливають на здоров’я людини та навколишнє середовище. Підтвердженням цього є аварія на Чорнобильській АЕС, наслідки якої відчувають до сьогодні як в Україні, так і за її межами. Дотримання рекомендацій щодо захисту населення від впливу радіації, які проаналізовано вище, дає змогу мінімізувати ризики, пов’язані із загибеллю великої кількості людей, а також зберегти їхнє здоров’я.

РОЗДІЛ 6 ЕКОЛОГІЯ

6.1. Методи узагальнення екологічної інформації

Статистичні таблиці є формою раціонального та систематизованого викладення цифрової інформації. Основною перевагою цифрової інформації, зведеної в таблиці, є компактність, наочність, виразність. Інформація стає легкодоступною і рельєфною, компактною і раціональною.

За допомогою таблиць можна досягти систематизації цифрової інформації, полегшити і прискорити ефект її сприйняття, інтенсифікувати пізнавальний процес явища, зекономити місце при викладенні інформації.

Статистичними таблицями вважають тільки ті, що містять наслідки статистичного аналізу еколого-економічних явищ і процесів [15].

Таблиця за своїм логічним змістом розглядається як «статистичне речення», що має свій підмет і присудок [15]. Підмет таблиці характеризує об'єкт дослідження, а присудок - це система показників, що відображає підмет як об'єкт.

Статистична таблиця має ряд горизонтальних рядків і вертикальних граф. Перетин рядків і граф утворює клітини таблиці. Ліві бічні і верхні клітини призначені для словесних заголовків, а решта - для числових.

Кожна таблиця має три заголовки:

– загальний – відображає зміст таблиці (його місце над таблицею);

– ліві (бічні) – найменування рядків, розкривають зміст підмета;

– верхні – найменування граф, розкривають зміст присудка.

Оптимальний за розміром об'єм таблиці складає 25-30 графо-клітин, який є добутком рядків і граф.

Статистичний графік є рисунком, який описує статистичні сукупності умовною мовою геометричних знаків тієї чи іншої форми: крапок, ліній, площин, фігур та різних їх комбінацій.

«Статистичні графіки – це спосіб умовного зображення цифрової інформації у вигляді крапок, ліній, стовпчиків, кругів або фігур [15].

Графічний метод використовують для популяризація цифрової інформації, забезпечення доступності її сприйняття та узагальнення цифрових даних. У більшості випадків статистичних графіків використовують не об'ємне зображення, складне за побудовою, а площинне. Графіки за призначенням поділяють на аналітичні, які дають можливість порівняти графічні образи, ілюстративні – допомагають при порівняні геометричних фігур та показують зміну розмірів явищ, а також інформаційні, які містять інформацію лише про об'єкт вивчення. Графічні образи на графіках поділяють на крапкові, лінійні; площинні (квадратні, кругові), просторові (об'ємні), зображувальні (фігурні).

Застосування табличних і графічних методів узагальнення екологічної інформації дають змогу наочно представити результати екологічних досліджень, порівняти цільові показники в часі, а також задовольнити вимоги статистичного опрацювання вхідних (досліджуваних) об’єктів чи явищ.

6.2. Формування бази статистичних даних в екології

Основою будь-якої статистичної обробки та оцінки екологічної інформації є збір інформації, що здійснюється методом статистичного спостереження. Статичне спостереження – одна з найважливіших стадій статистичного дослідження. Воно вважається фундаментом статистичного дослідження, адже в процесі його здійснення формується первинна статистична інформація, яка на наступних етапах дослідження підлягає обробленню і аналізу.

Статичне спостереження в екології – це планомірний, науково-організований збір масових даних про екологічні явища і процеси. Здійснюється шляхом реєстрації за заздалегідь розробленою програмою спостереження.

Об'єктом спостереження є стан забруднення навколишнього середовища (природних об’єктів) атмосферного повітря, природних водних об’єктів, земель та ґрунтів. Збір даних проводиться не стихійно, а регулярно, що дає змогу

вивчити тенденції, напрями, закономірності розвитку екологічних явищ і процесів. План статистичного спостереження передбачає широке коло питань методики та організації збору статистичної інформації, контролю її якості та вірогідності. Для об'єкта статистичного спостереження характерне те, що його не можна вивчати безпосередньо в цілому, для цього потрібно виділити в його складі окремі одиниці.

Одиниця статистичного спостереження – це складовий елемент об'єкта дослідження, який є основою рахунку і носієм істотних ознак та властивостей, які підлягають реєстрації. Це первинний елемент об'єкта дослідження. Одиницю спостереження встановлюють, виходячи із завдань спостереження і складності об'єкта дослідження.

Правильне визначення одиниці спостереження має істотне значення для організації і проведення статистичного дослідження. Цим значною мірою зумовлюється об'єктивність одержаних результатів.

Основне завдання статистичного спостереження – отримання вірогідних статистичних даних, які об’єктивно характеризують явища і процеси суспільного життя. Інформація статистичного спостереження повинна бути об'єктивною і якісною, а отже, забезпечуватись правильною науковою організацією її одержання, належним виконанням самого спостереження. Завдання статистичного спостереження зумовлюється завданнями, які ставляться перед дослідженням певних екологічних процесів і явищ і випливають з потреб управління ними.

Наукова організація статистичного спостереження зумовлює дотримання певних вимог щодо його здійснення:

– статистичне спостереження повинно здійснюватися на науковій основі заздалегідь розробленою програмою, яка забезпечувала б науковий підхід до вирішення методологічних та організаційних питань;

– статистичне спостереження повинне забезпечувати збір масових даних, у яких відбивається вся сукупність фактів. Неповнота зведень про досліджувані процеси призведе до помилкових висновків з результатів аналізу;

– орієнтація статистичного спостереження на збирання не тільки інформації, яка безпосередньо характеризує досліджуваний об'єкт, а й такої, що сприяє зміні його стану;

– інформація, одержана за результатами статистичного спостереження, повинна бути вірогідною;

– дані статистичного спостереження повинні бути порівнювані. Лише в такому разі забезпечується їх узагальнення і зіставлення у просторі й часі.

Програма статистичного спостереження представляє собою перелік питань, на які треба одержати відповіді в процесі збирання статистичних зведень щодо кожної досліджуваної одиниці.

Один і той самий об'єкт може бути обстежений з різних боків. Тому склад і зміст питань програми спостереження залежить від завдань дослідження і особливостей об'єкта. Вона повинна охоплювати широке і повне коло відомостей. Чим ширша програма, тим повніше висвітлюється досліджуване явище. Проте в неї не слід включати зайвих питань, які могли б ускладнити і розтягнути термін розроблення даних. У той же час не слід складати програму надто вузько, адже в дослідження можуть не потрапити важливі питання.

У статистичний практиці застосовують дві організаційні форми спостереження: звітність і спеціально організовані статистичні спостереження.

Звітність – це форма статистичного спостереження, при якій статистичні дані надходять у статистичні органи від підприємств і установ у вигляді обов'язкових і таких, що мають юридичну силу звітів про їх роботу.

Звітність підприємств, установ та організацій є поки що основним джерелом статистичної інформації. У ній передбачається система строго регламентованих показників, які характеризують діяльність підприємств, установ та організацій. Зміст звіту, форма і термін подання також встановлюється вищим статистичним органом. Звітність складають на основі документів первинного оперативно- технічного і бухгалтерського обліку. Вірогідність гарантується також юридичною відповідальністю керівників підзвітних підприємств та організацій.

Перелік усіх форм із зазначенням їх реквізитів називають табелем звітності.

За різними ознаками статистичну звітність поділяють на окремі види. Насамперед розрізняють типову і спеціалізовану звітність:

– типова звітність має єдину форму і зміст для всіх підприємств окремої галузі або всього народного господарства.

– спеціалізована звітність властива тим підприємствам чи окремим виробництвам, що мають свої специфічні особливості.

За періодичністю подання звітність буває тижнева, двотижнева, місячна, квартальна, різна; за способом подання - термінова (телеграфна) і поштова. Вид звітності впливає на техніку збору і зведення статистичної інформації. Удосконалення статистичної звітності на сучасному етапі відбувається у напрямі скасування термінової звітності та скорочення кількості поштових звітів.

ВИСНОВКИ

1. Проаналізовано сучасний стан досліджень у сфері забезпечення якості програмних систем та сформульовано вимоги до моделі розподілу завершеності ПС, що належать до класу систем оброблення даних, визначено, що основною прогнозованою мірою надійності ПС на ранніх стадіях має бути кількість (щільність) дефектів, визначено необхідність впровадження процесів вимірювання та оцінювання у проекти ПС; побудову моделі раннього прогнозування з урахуванням потреб динамічного ЖЦ (без “прив’язки” до стадій каскадної моделі ЖЦ); необхідність вдосконалення процесів ЖЦ (зокрема верифікації та керування проектом) шляхом надання методичної та інструментальної підтримки.

2. Розроблено метод прогнозування надійності програмних застосувань - найбільш важливої характеристики якості для ПС оброблення даних. В основі методу – прогнозування щільності дефектів у програмних засобах на початку кожного етапу проекту.

3. Обґрунтовано застосування графічної моделі прогнозування щільності дефектів із застосуванням байєсівської мережі, що забезпечує врахування причинно-наслідкових зв’язків чинників та ймовірнісні судження менеджерів про поточну ситуацію в проекті.

4. Розроблено програмний модуль прогнозування якості програмних систем на ранніх стадіях життєвого циклу та доповнено програмний комплекс управління якістю програмних систем, що дало змогу автоматизувати процес прогнозування і провести експериментальні дослідження розробленого методу та моделей.

5. Проведено техніко-економічні розрахунки щодо доцільності проведення досліджень дипломної роботи, які підтвердили економічну ефективність розробленого методу і засобів.

6. Проаналізовано питання охорони праці і безпеки в надзвичайних ситуаціях

7. Проаналізовано питання методів узагальнення екологічної інформації та формування бази статистичних показників в екології.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сергієнко І.В. Становлення і розвиток досліджень з інформатики/ І.В. Сергієнко// - К.: Наукова думка. – 1998. - 205 с.

2. Основы инженерии качества программных систем / Ф.И.Андон, Г.И.Коваль, Т.М. Коротун, В.Ю. Суслов / Под ред. И.В. Сергиенко. – К.: Академпериодика. - 2002. - 504 с.

3. Лаврищева Е.М. Основы технологической подготовки разработки прикладных программ систем обработки данных/ Лаврищева Е.М. // Препринт / АН УССР, Ин-т кибернетики им. В.М. Глушкова; 87-5. – Киев.- 1987. – 30 с.

4. ДСТУ 3918-99 Інформаційні технології. Процеси життєвого циклу програмного забезпечення.– Київ.– Держстандарт України.– 2000. – 49 с.

5. ДСТУ 2844-94 Програмні засоби ЕОМ. Забезпечення якості. Терміни та визначення. – Київ. – Держстандарт України. – 1994. – 18 с.

6. IEEE 982.1 Standard Dictionary of Measures to produce Reliable Software. – IEEE. - 1988.

7. ISO 9000-3 Quality management and quality assurance standards – Part 3: Guidelines for the application of ISO 9001:1994 to design, development, supply, installation and maintenance of computer software.- 1997.

8. Highsmith J. Retiring Lifecycle Dinosaurs // Software Testing & Quality Engineering. - May/June 2000.- http: //www.stqemagazine.com/

9. Андон Ф.И. Методы инженерии распределенных компьютерных систем/ Ф.И. Андон, Е.М. Лаврищева //. – К.: Наукова думка.- 1997.- 228 с.

10. Лаврищева Е.М. Сборочное программирование/ Е.М. Лаврищева, В.Н. Грищенко // – К.: Наукова думка. - 1991. – 213 с.

11. Лаврищева Е.М. Сборочное программирование. Некоторые итоги и перспективы/ Е.М. Лаврищева // Проблемы программирования.- 1999.- №2.- С. 20 – 31.

12. Лаврищева Е.М. Парадигма интеграции в программной инженерии. / Е.М. Лаврищева//Проблемы программирования.– 2000.– № 1-2.- С. 351 – 360.

13. Бабенко Л.П. От программирования к программной инженерии / Л.П. Бабенко, Е.М. Лаврищева // Проблемы программирования.- 2000.– № 3-4.– С.18–21.

14. Бабенко Л.П. Основи програмної інженерії/ Л.П.Бабенко, К.М. Лаврищева // – К.: Знання. - 2001. – 269 с.

15. Котляров В.П. Вхідна мова опису тестів на основі розширених MSC діаграм / В.П. Котляров, В.О. Сухомлинов //Кибернетика и системный анализ. – 2004. -№1. –С. 52 - 62.

16. Липаев В.В. Тестирование программ/ В.В. Липаев // -М.: Радио и связь. - 1986. -296 с.

17. Лаврищева Е.М. Построение процесса тестирования программных систем/ Е.М. Лаврищева, Т.М. Коротун // Проблемы программирования.-2002. - № 1-2. - С. 272-281.

18. Липаев В.В. Надежность программных систем.- М.: СИНТЕГ.-1998.– 321с.

19. Липаев В.В. Выбор и оценивание характеристик качества программных систем. Методы и стандарты. – М.: СИНТЕГ.- 2001. – 224 с.

20. Липаев В.В. Методы обеспечения качества крупномасштабных программных систем. – М.: СИНТЕГ.- 2003.–510 с.

21. ISO/IEC 9126-2001 Software engineering – Product quality (Part 1 – 4).

22. Липаев В.В. Качество программных средств. Методические рекомендации / Под ред. А.А. Полякова.- М: Янус-К.- 2002.-400 с.

23. Кулаков А.Ф. Обеспечение качества программных средств ЭВМ. Состояние, проблемы и пути их решение/ А.Ф. Кулаков, Е.В.Шмигидина // Проблемы программирования. –1997.- Вып.1.- С. 107 – 114.

24. Коваль Г.И. Проблемы анализа причинно-следственных связей отказов и ошибок в ПО/ Г.И. Коваль, Т.М. Коротун // Сб. Программная инженерия.- Киев: Ин-т кибернетики им. В.М. Глушкова НАНУ, 1994. - С. 83 - 93.

25. Мороз Г.Б. Определение целей и задач инженерии надежности программного обеспечения/ Г.Б. Мороз, Г.И. Коваль, Т.М. Коротун // Проблемы программирования.- 1997.- Вып. 1.- С. 98 – 106.

26. Планирование обеспечения надежности информационных систем / Г.И. Коваль, Т.М. Коротун, Т.Л. Яблокова, Л.И. Куцаченко // Проблемы программирования.- 2001.- №3-4. - С. 40 - 47.

27. Модель оценки технологической зрелости организаций-разработчиков ПО / Ф.И. Андон, В.Ю. Суслов, Т.М. Коротун, Г.И. Коваль, О.А. Слабоспицкая // Проблемы программирования.- 1998. - №4. - С. 46 -57.

28. Abd-Allah A. Extending reliability block diagrams to software architectures / Center for Software Engineering. Computer Science Department. University of Southern California. Los Angeles. Technical Report: USC-CSE-97-501.- http:// sunset.usc.edu/publications/ TECHRPTS/1997/usccse97-501/usccse97-501.ps

29. Lakey P.B. System and software reliability assurance notebook / P.B. Lakey, A.M. Neufelder // Rome Laboratory Report, Griffiss Aif Force Base, Rome NY. – 1997. – 186 с.

30. Мороз Г.Б. Концепция профилей в инженерии надежности программных систем / Г.Б. Мороз, Г.И. Коваль, Т.М. Коротун // Математичні машини і системи. – 2004.- №1. – С. 166 – 184.

31. Cheung R. A User-oriented Software Reliability Model // IEEE Trans. Soft. Eng., -1980. - SE-6, N. 2. - P. 11- 125

32. Musa J.D. Operational Profiles in Software Reliability Engineering // IEEE Software.-V.10.- N.2.- 1993.- P. 14 - 32.

33. Munson J. Software reliability as a function of user execution patterns/ J. Munson, S. Elbaum // Proc. of the 32nd Hawaii International Conference on System Sciences. - 1999. - P.1-12.

34. Rajgopal J. V. Optimum Combined Test Plans for Systems and Components/ J. Rajgopal, M. Mazumdar, S. Majety // IIE Transactions. – 1999.- v. 31.- P. 481-490.

35. Whittaker J. Markov chain model for statistical Software testing // IEEE Trans. Soft. Eng. – 1994. - SE-20, N.10.- P. 812 - 824.

36. Runesson P. Usage Modelling: The Basis for Statistical Quality Control // Proceedings 10th Annual Software Reliability Symposium, Denver, Colorado. – 1992. - P. 77 - 84.

37. Hecht H. An Alternative Software Reliability Assessment // Proceedings 14h International Symposium on Software Reliability Engineering (ISSRE 2003), Denver, Colorado.- 2003. - P. 293 - 295.

38. Ohlsson N. Quality Improvement by Identification of Fault-Prone Modules using Software Design Metrics/ N. Ohlsson, M. Helander, C. Wohlin // Proceedings Sixth International Conference on Software Quality. – 1996. - P. 1 – 13.

39. Lyu M.R. Optimization Of Reliability Allocation And Testing Schedule For Software Systems // Proceedings Eighth International Symposium on Software Reliability Engineering (ISSRE '97).- 1997. - P. 336 - 438.

40. Helander M.E. Planning Models for Software Reliability and Cost/ M.E. Helander, M. Zhao, N. Ohlsson //IEEE Trans. Softw. Eng. – 1998. – V. 24. - N. 6.- P. 420 – 434.

41. Коваль Г.И. Прогнозирование надежности ПО компьютерных систем. // Сб. материалов конференции. УкрПРОГ’98, 2-4 сентября 1998 г., Киев.-1998. - С. 358 - 361.

42. Парадигма качества программного обеспечения / Андон Ф.И, Суслов В.Ю., Коротун Т.М., Коваль Г.И. // Проблемы программирования. –1999. - №2. – С. 51 - 62.

43. Мороз Г.Б., Лаврищева Е.М. Модели роста надежности ПО/ Г.Б. Мороз, Е.М. Лаврищева // Киев. - 1992. – 25 с. - (Препр. / АН Украины. Ин-т кибернетики им. В.М. Глушкова; 92 - 38).

44. Управление риском проектов ПО / Ф.И. Андон, В.Ю. Суслов, Т.М. Коротун, Г.И. Коваль, О.А. Слабоспицкая // Проблемы программирования.- 1999.- №1. - С. 53 – 62.

45. Fenton N.E. A critique of software defect prediction models/ N.E Fenton // IEEE Trans. On Soft. Eng. – 1999. – V. 25. – N.5. – P. 675 – 689.

46. Fenton N.F. Quantitative analysis of faults and failures in a complex software system/ N.F. Fenton // IEEE Trans. On Soft. Eng. – 2000. – V. 26. – N. 8. – P. 797 – 814.

47. Коваль Г.И. Подход к прогнозированию надежности ПО при управлении проектом/ Г.И. Коваль // Проблемы программирования. –2002. - № 1 – 2. – С. 282 – 290.

48. Коваль Г.И. Методы определения размера ПО/ Г.И. Коваль // Проблемы программирования. - 1999. - №1.- С. 63 - 71.

49. Malaiya Y.K. What do the Software Reliability Growth Model Parameters Represent/ Y.K. Malaiya, J. Denton // Proc. IEEE-CS Int. Symp. on Software Reliability Engineering ISSRE. - Nov. 1997. - P. 124 - 135.

50. Chulani S. Constructive quality modeling for defect density prediction: COQUALMO/ S. Chulani // International Symposium on Software Reliability Engineering (ISSRE'99), Boca Raton, November 1-4. - 1999.

51. Бойчик І. М. Економіка підприємства: Навч. посібник./ І. М. Бойчик // — К.: Атіка, 2004.—480 с.

52. Жидецький В.Ц. Охорона праці користувачів комп’ютерів / В.Ц. Жидецький// – Львів: Афіша, 2000. – 176с.

53. Желібо Є. Безпека життєдіяльності /Є. Желібо, Н. Заверуха, В. Зацарний// – К.: 2001 – 483 с.

54. Джигирей В.С. Екологія та охорона навколишнього природного середовища. Навчальний посібник / В.С. Джигирей //– К.: Знання. – 2000. – 356 с.

Додаток А

Текст публікації

Матеріали IV Міжнародної науково-технічної конференції молодих учених та студентів
Актуальні задачі сучасних технологій – Тернопіль 25-26 листопада 2015

УДК 004.4
[1]В.В. Яцишин канд. техн. наук, доц., [2]Р.Б. Ладика канд. фіз-мат. наук, [1]А.В. Любий
[1]Тернопільський національний технічний університет імені Івана Пулюя ,Україна
[2]Тернопільський державний медичний університет ім. І. Я. Горбачевського, Україна

ПРОГНОЗУВАННЯ ЯКОСТІ ПРОГРАМНИХ СИСТЕМ НА СТАДІЯХ ЖИТТЄВОГО ЦИКЛУ

V.V. Yatsyshyn Ph.D., Assoc. Prof., R.B. Ladyka, Ph.D., Assoc. Prof., A.V.Lubiy
SOFTWARE QUALITY PREDICTION AT THE LIFE CYCLE STAGES

Якість програмного забезпечення невід'ємно асоціюється з характеристикою надійності. Надійність є визначальною в контексті задоволення працездатності та безвідмовності програмного продукту і , як правило, її оцінюють за моделями зростання надійності на завершальних етапах проекту. Однак, важливим аспектом розробки програмних систем є прогнозування якості як на ранніх етапах життєвого циклу, так і впродовж усього часу роботи над проектом.

Відомо, що раннє прогнозування надійності ПС полягає в побудові проекції значень вимірів надійності, отриманих за внутрішніми метриками на певній стадії ЖЦ, на значення вимірів надійності, обчислюваних за зовнішніми метриками на будь-якій наступній стадії та в кінці розроблення ПС [1].

При ранньому прогнозуванні використовують дані з робочих продуктів, процесів та ресурсів проекту, які збираються та опрацьовуються в ході кількісного оцінювання. На основі зібраних даних можна будувати моделі прогнозування дефектів, які використовуються протягом усього життєвого циклу програмних систем.

До результатів раннього прогнозування якості програмних систем входять значення інтенсивності відмов програмних систем на початку тестування системи. Це мінімальна надійність, яка згодом може тільки підвищуватися завдяки усуненню дефектів у програмних системах. Мета, яку досягають за рахунок раннього прогнозування надійності програмних систем, полягає у визначенні, удосконаленні методів і процесів розроблення. При цьому можна забезпечити мінімальну щільність дефектів до моменту початку системного тестування.

Пізнє прогнозування надійності ПС пов'язується із застосуванням аналітичних моделей зростання надійності на стадії системного тестування після збору певної кількості даних про відмови ПС. Всі відомі моделі зростання надійності ПС фактично охоплюють період після раннього прогнозування, коли надійність підвищується внаслідок тестування і виправлення дефектів [2]. Якщо результати прогнозування за цими моделями показують, що надійність ПС низька, єдина можливість її підвищення - продовження тестування і усунення дефектів протягом розрахованого періоду часу [1]. Таке прогнозування надійності звичайно може зменшити ризик відмови ПС в експлуатації, але не зменшить ризику проекту.

Література

1. Основы инженерии качества программных систем / Ф.И. Андон, Г.И. Коваль, Т.М. Коротун, Е.М. Лаврищева, В.Ю. Суслов; —К.: Академперіодика, 2007. — 670 с.

2. Коваль Г. Байєсівські мережі як засіб оцінювання та прогнозування якості програмного забезпечення / Г. Коваль – Проблемы программирования. – 2005 – № 2. – С. 15-24.

Додаток Б

Форми для процесів вимірювання та верифікації

1. Основні форми документів для процесу вимірювання

Форма контрольного листка для обліку дефектів у процесі вимірювання

Стан проблем	**Включати**	**Не включати**	**Кількість**
Відкриті	**X**		**X**
Визнані			
Оцінені			
Вирішені			
Закриті	**X**		**X**
Тип проблем	**Включати**	**Не включати**	**Кількість**
Дефекти у ПЗ			
Дефекти вимог	**X**		**X**
Дефекти проекту	**X**		**X**
Дефекти коду	**X**		**X**
Дефекти документів	**X**		**X**
Дефекти тестів		**X**	
Інші дефекти		**X**	
Інші проблеми			
Проблеми апаратури		**X**	
Проблеми ОС		**X**	
Помилки користувача		**X**	
Помилки в операціях		**X**	
Нові вимоги		**X**	
Унікальність	**Включати**	**Не включати**	**Кількість**
Вперше	**X**		**X**
Повторно		**X**	
Не відомо		**X**	
Критичність	**Включати**	**Не включати**	**Кількість**
1 рівень (вищий)	**X**		**X**
2 рівень	**X**		**X**
3 рівень	**X**		**X**
4 рівень	**X**		**X**
Не відомо	**X**		
…			

Форма опису структурованих цілей

Об'єкт інтересу: _____
_____ (процес, продукт, ресурс, задача, дія, [сутність] та ін.)

Намір: _____ (охарактеризувати, проаналізувати, оцінити та ін.)
(що зробити)
_____ ([сутність], [аспект], [атрибути] та ін.)
(з чим)
для того щоб _____ (зрозуміти, обгрунтувати, передбачити, спланувати, порівняти та ін.)

Позиція:
Перевірити _____ (поведінка, стабільність, якість, прогрес, [атрибути] та ін.)
с позицій _____ (розробника, менеджера, керівника, замовника та ін.)

Умови: _____(чинники середовища або інші параметри для визначення контексту)

Форма контрольного листка технологічної підготовки

№ п/п	Задачі	Відмітка про виконання (%)			
		Елемент даних			
		1	2	.	n
1	Визначити елементи даних				
2	Визначити частоту збору даних і контрольні точки в процесі, в яких будуть виконуватися вимірювання				
3	Визначити витрати часу, необхідні для перенесення результатів, зібраних в кожній контрольній точці, в базу даних або їх доставку споживачам				
4	Розробити форми для збору і фіксації даних				
5	Розробити процедури для збору і фіксації даних				
6	Визначити, як дані зберігаються, і як до них буде забезпечуватися доступ. Встановити, хто несе відповідальність за проектування БД і за введення і ведення даних				
7	Визначити, хто буде збирати дані і мати до них доступ. Призначити відповідальних за ці дії				
8	Визначити, як дані будуть аналізуватися				
9	Визначити, як буде складатися звіт про зібрані дані				
10	Ідентифікувати службові інструменти, які потрібно розробити або придбати для того, щоб допомогти автоматизувати процес				
11	Підготувати методики по виконанню збору даних				

2. Основні форми для процесу верифікації (інспекції)

Сповіщення про проблему

Повідомлення про проблему в робочому продукті
Найменування проекту:_____ **Найменування робочого продукту (частини)** _____
Ідентифікатор проблеми:_____ **Детальний опис проблеми:**_____ _____
Серйозність проблеми: *дуже важлива* (можливий суттєвий дефект, повторити процедуру перевірки) *помірно важлива* (можливий помірний дефект, усунення проконтролювати) *не важлива* (неістотний технічний дефект, контроль необов'язковий) **Критичність проблеми для проекту:** **критична (дефект на верхньому рівні, глобальні ресурсоємні переробки (критичний ресурс – час))** *усунена* (локальні ресурсоємні переробки (час)) *легко усунена* (не вимагає додаткових витрат ресурсів) **Потенційний збиток: оцінка збитку на кожній з подальших стадій ЖЦ у разі невирішення проблеми** **Пріоритет усунення:** *негайно, до певного терміну, в наступній версії*
Стан проблеми: *передбачувана, відкрита, закрита, закрита із зауваженнями повторно виникла* **Хронологія вирішення проблеми:** *4 важливі дати, пов'язані із змінами станів проблеми*
Дата виявлення:_____ **Хто знайшов проблему:**_____
Кому направлена: _____ **Відповідальний за усунення:**_____
Укладач (підпис):_____ **Дата складання:**_____

Структура контрольного опитувального листа

Позначення виду продукту	Найменування виду (типу) робочих продуктів		
	Номер категорії (підкатегорії)	Найменування категорії (підкатегорії) дефектів	
		Номер питання	Формулювання питання для виявлення дефекту певної категорії

Структура описа дефекту:

- *порядковий номер дефекту в списку* (а можливо, і складовий номер з посиланням на код інспектора, що знайшов проблему, і номер проблеми, визнаної дефектом)
- *категорія (підкатегорія) дефекту* і *номер питання* (по запитальнику), пошук відповіді на який послужив поштовхом до виявлення проблеми;
- *характер дефекту* (упущення автора, помилкове рішення, помилка верхнього рівня);
- *рівень серйозності дефекту* (істотний, помірний, неістотний)
- *місцезнаходження* в продукті (номер сторінки, розділу, параграфа, абзацу, пропозиції) або посилання на опис відповідної проблеми, визнаної дефектом;
- *орієнтовний час* на усунення дефекту;
- *ідентифікація особи* (групи), якій (яким) продукт прямує на переробку;
- *тип повторної перевірки продукту (З – нарада автора з координатором, R – нарада автора з координатором і інспекторами);*
- відмітка про усунення дефекту. Дата підтвердження усунення дефекту. Проставляється тільки після повторної перевірки переробок продукту (на етапі 7).

Структура звіту про перевірку

- *титульна частина.* Ідентифікація проекту і продукту, вид інспекції, тип проведеної наради;
- *стан продукту.* Один з можливих варіантів тексту:
- «*Читання завершено. Продукту призначена категорія* (A/C/R)»
  - «Читання не завершено. Буде продовжено на нараді < дата> з <точна вказівка місця в робочому продукті, з якого повинно бути продовжено читання>. Категорія не призначена»;
- *дата перевірки* виправлень і *тип повторної перевірки* (якщо продукту призначена категорія З або R);
- *список дефектів (набір описів дефектів);*
- підсумкова інформація:
- число розглянутих проблем
  - число зареєстрованих дефектів
  - число відкритих проблем, перенесених на «третю годину»
  - число закритих проблем <число проблем, закритих на «третій годині»>, з них встановлено дефектів <число> (заповнюється після закінчення «третьої години»);
- *коментарі*. Інформація про необхідність розгляду нових документів для вирішення відкритих проблем (наприклад, документів попереднього рівня розробки проекту), необхідність залучення нових осіб для участі в обговоренні (експертів, розробників продуктів попередніх рівнів) і ін;
- *підписи членів групи інспекції*. Вказується ПІБ кожного учасника і підпис.

Форма зведеного звіту про дефекти

Номер дефекту в списку дефектів	Категорія дефекту	Рівень серйозності дефектів								
		Істотних			Помірних			Не істотних		
		упущень	помилок	помилок верхнього рівня	упущень	помилок	помилок верхнього рівня	упущень	помилок	помилок верхнього рівня
1										
2										
.										
Разом										

Структура звіту про динаміку виявлення дефектів

Звіт про динаміку виявлення дефектів

Дата:_____ Тип звіту: тижневий

Ідентифікатор звіту з проблеми	Дата виявлення	Опис	Серйозність

Всього виявлено дефектів:_____
З них:
Критичних:_____
Серйозних:_____
Незначних:_____
Тривалість перевірки:_____

Структура звіту про динаміку усунення дефектів

Звіт про динаміку усунення дефектів

Дата:_____ Тип звіту: тижневий

Ідентифікатор звіту з проблеми	Дата відкриття	Опис	Пріоритет усунення	Дата закиття

Всього виявлено: _____
Всього усунено:_____
Відкрито: _____

Структура звіту про виконання циклу перевірки ПрЗ

Дата складення звіту:_____
Проект/Програма:_____Версія:_____ Цикл:_____
Дата початку перевірки: _____ Рівень/Задача перевірки:_____

Серйозність дефектів	Усунено	Відкладено	Відхилено
Критичні			
Серйозні			
Середні			
Інші			
Всього			

Тривалість перевірки:_____
Трудомісткість перевірки:_____
Звіт склав:_____