

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**  
Кафедра електроніки і комп'ютерної техніки

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА  
НА ТЕМУ:

**ТЕСТУВАННЯ ЯКОСТІ ТА ВЕРИФІКАЦІЯ  
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ**

Завідувач кафедрою електроніки  
і комп'ютерної техніки

\_\_\_\_\_ А.С. Опанасюк

Керівник роботи

\_\_\_\_\_ І.А. Кулик

Виконав студент гр. ТК-81

\_\_\_\_\_ Р.В. Пушкар

Сумський Державний Університет

Факультет Електроніки та ІТ

Кафедра Електроніки

та комп'ютерної техніки

Спеціальність Телекомунікації та радіотехніка

ЗАТВЕРДЖУЮ:

Зав. кафедри Опанасюк А.С.

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г.

### Завдання

на дипломний проект студенту

Пушкар Роман Віталійович

(прізвище, ім'я, по батькові)

#### 1. Тема проекту

«Тестування якості та верифікація програмного забезпечення телекомунікаційних систем»

затверджено наказом по інституту від « 12 » квітня 2022 р. № 0241-VI

#### 2. Термін здачі студентом закінченого проекту \_\_\_\_\_

3. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають перегляду) Введення, Життєвий цикл послуг, Тестування вимог, Архітектура інтелектуальної мережі, Тестування за знанням системи, Регресійні тести, Тестування на відмову та відновлення, Функціональне і нефункціональне тестування, Тестування ефективності, Тестування за використання сценарію, Управління помилками TMN, Висновки, Література.

#### **4. Перелік графічного матеріалу**

Малюнок 2.1 - Тестування в життєвому циклі послуг

Малюнок 3.1. - Архітектура інтелектуальної мережі

Малюнок 6.1 - Підхід до тестування нової служби.

Малюнок 6.2. - Підхід до наскрізного тестування, що відстежується

Малюнок 6.3 - Підхід тестування відповідності послуг

Студент-дипломник Пушкар Р.В.

Керівник проекту Кулик І.А.

## РЕФЕРАТ

В кваліфікаційній роботі бакалавра визначено структуру процесу тестування, пов'язаного з експлуатацією нових послуг. Спочатку представлений огляд понять та концепцій життєвого циклу послуги. Потім визначається процедура, що охоплює всі види тестування, які мають бути розглянуті до та під час експлуатації послуги.

В кваліфікаційній роботі бакалавра проведений детальний огляд основ для проведення випробувань та перевірки нових телекомунікаційних послуг. Замість того, щоб пропонувати нові та амбітні методи тестування, ми досліджуватимемо роль тестування у життєвому циклі послуги, пояснимо різні терміни та поняття, пов'язані з життєвим циклом послуги, запропонуємо процедуру для всіх видів тестування, які мають бути розглянуті до та під час надання послуги.

Для виконання випускної роботи використано 9 літературних джерел. Область застосування даного пристрою - схема значною мірою заснована на архітектурі інтелектуальної мережі. Однак вона придатна для застосування та в інших архітектурах.

Кваліфікаційна робота бакалавра містить 35 сторінок тексту і 5 рисунків.

Ключові слова: життєвий цикл послуг, архітектура інтелектуальної мережі, код Файра, тестування

## ЗМІСТ

Анотація .....	7
Введення.....	8
1. Життєвий цикл послуг.....	8
1.1.Що таке послуга? .....	8
1.2.Етапи життєвого циклу тестування послуги.....	9
1.3.Тестування в життєвому циклі послуг.....	10
2. Тестування вимог .....	11
2.1.Методи тестування вимог.....	11
2.2.Критерії оцінки якості вимог.....	12
3. Архітектура інтелектуальної мережі.....	13
4. Тестування послуг.....	14
4.1.Тестування платформи.....	15
4.2.Тестування нової послуги.....	15
4.3.	
5. Тестування за знанням системи.....	16
5.1.Чорна коробка.....	16
5.2.Біла коробка.....	17
5.3.Сіра коробка.....	17
6. Рівні тестування за ступенем ізольованості компонентів.....	18
6.1.Модульне тестування.....	18
6.2.Інтеграційне тестування.....	18
6.3.Наскрізне тестування служби.....	19
6.3.1. Контрольований підхід до наскрізного тестування служби.....	21
6.4.Тестування відповідності послуг.....	24
6.5. Визначення пріоритету у виборі підходу тестування.....	25
7. Регресійні тести.....	26
8. Тести на взаємодію.....	27
9. Тести управління сервісом.....	28
10.Функціональне і нефункціональне тестуванням.....	29
10.1. Що таке функціональне тестування? .....	30
10.2. Що таке нефункціональне тестування? .....	31

					<b>ЕлІТ 6.172.00.02.464 ПЗ</b>			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>	Тестування та веревікація програмного забезпечення телекомунікаційних систем	<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
Разраб.	Пушкар Р.В.						5	35
Провер.	Кулик І.А.							
Реценз.								
Н. контр.	Кулик І.А.					<b>СумДУ ТК-81</b>		
Утверд.	Опанасюк А.С.							

## ЗМІСТ

11.Тестування ефективності.....	30
12.Тестування на відмову та відновлення.....	31
13.Тестування за використання сценарію.....	30
13.1. Дослідницьке тестування.....	31
13.2. Вільне тестування.....	32
14.Управління помилками TMN.....	33
15.Висновки.....	34
16.Література.....	35

					ЕЛІТ 6.172.00.02. 464 ПЗ	Лист
Изм.	Лист	№ докум.	Підпись	Дата		6

## Анотація

Тестування послуг - це новий напрямок розвитку, викликане швидким зростанням кількості послуг, запропонованих операторами зв'язку. У цій статті визначено структуру процесу тестування, пов'язаного з експлуатацією нових послуг. Спочатку представлений огляд понять та концепцій життєвого циклу послуги. Потім визначається процедура, що охоплює всі види тестування, які мають бути розглянуті до та під час експлуатації послуги.

## Введення

У неспокійному світі телекомунікацій швидке впровадження нових телекомунікаційних послуг може забезпечити постачальникам послуг конкурентну перевагу. Впровадження архітектури інтелектуальної мережі дозволить операторам скоротити терміни впровадження нових послуг з багатьох років до кількох місяців або навіть кілька тижнів. Крім того, у операторів з'явиться можливість створювати нові послуги у власному середовищі створення послуг.

У телекомунікаціях, з іншого боку, постачальники послуг навряд чи можуть дозволити собі збої у наданні послуг. Збої в мережі можуть мати катастрофічний ефект на імідж оператора як надійного постачальника послуг. Ця загроза обумовлює необхідність "тестування послуг". Перш ніж телекомунікаційна послуга буде розгорнута в мережі, необхідно переконатися, що послуга функціонально коректна, і що вона не завдає шкоди постійній базі.

Мета цього документа – надати основу для проведення випробувань та перевірки нових телекомунікаційних послуг. Замість того, щоб пропонувати нові та амбітні методи тестування, ми досліджуватимемо роль тестування у життєвому циклі послуги. Ця схема значною мірою заснована на архітектурі інтелектуальної мережі. Однак вона придатна для застосування та в інших архітектурах.

У другому розділі будуть пояснені різні терміни та поняття, пов'язані з життєвим циклом послуги. У третьому розділі буде запропоновано процедуру для всіх видів тестування, які мають бути розглянуті до та під час надання послуги. Четвертий розділ є завершальним.

					ЕлІТ 6.172.00.02. 464 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

# 1 ЖИТТЄВИЙ ЦИКЛ ПОСЛУГ

У цьому розділі наведено деякі терміни та поняття, що використовуються у світі надання послуг. Розуміння цих понять є важливим для того, щоб побачити їх зв'язок із завданнями тестування, представленими в наступному розділі. Читачі, які добре знайомі з проектуванням послуг, можуть відразу перейти до наступного розділу.

## 1.1. Що таке послуга?

Дуже важливим аспектом є те, що таке послуга взагалі. Виявляється, що у різних документах використовується безліч різних визначень. Це пов'язано з тим, що послуги можуть розглядатися з комерційної, технічної та користувальницької точок зору. Оскільки тестування зосереджено лише з технічної перспективи, технічне визначення, мабуть, буде найкориснішим.

З цієї причини ми вводимо таке визначення сервісу:

Телекомунікаційна послуга - послуга, що надається провайдером телекомунікацій, або певний набір можливостей передачі інформації, що надається групі користувачів телекомунікаційною системою.

Ми будемо використовувати це визначення послуги для тестування послуг, запропонованої в даній статті.

Приклад може прояснити значення цього визначення. Розглянемо послугу, яка комерційно визначається як: "Послуга переадресації дзвінків, що залежить від часу, яка переводить всі дзвінки на номер А з 17.00 до 9.00 на інший номер В". Програмне забезпечення, що забезпечує цю послугу, включає код, що перевіряє номер А і приймає рішення про його переадресацію. Необхідне обладнання може складатися з годинникового механізму, який забезпечує відповідні часові сигнали для цієї послуги.

					ЕЛІТ 6.172.00.02. 464 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8



## 1.2. Етапи життєвого циклу тестування послуги

Щоб зрозуміти різні аспекти, які сприяють реалізації послуги, корисно простежити етапи, пройдені під час розробки послуги.

У цій роботі ми вважаємо за краще розташовувати більшість заходів з тестування послуг після завершення завдань із впровадження послуг. Завдання тестування послуг виконуються до встановлення послуги в мережу. Тому результатом тестування сервісу має бути сервіс, який демонструє правильну поведінку і готовий до встановлення робочої мережі.

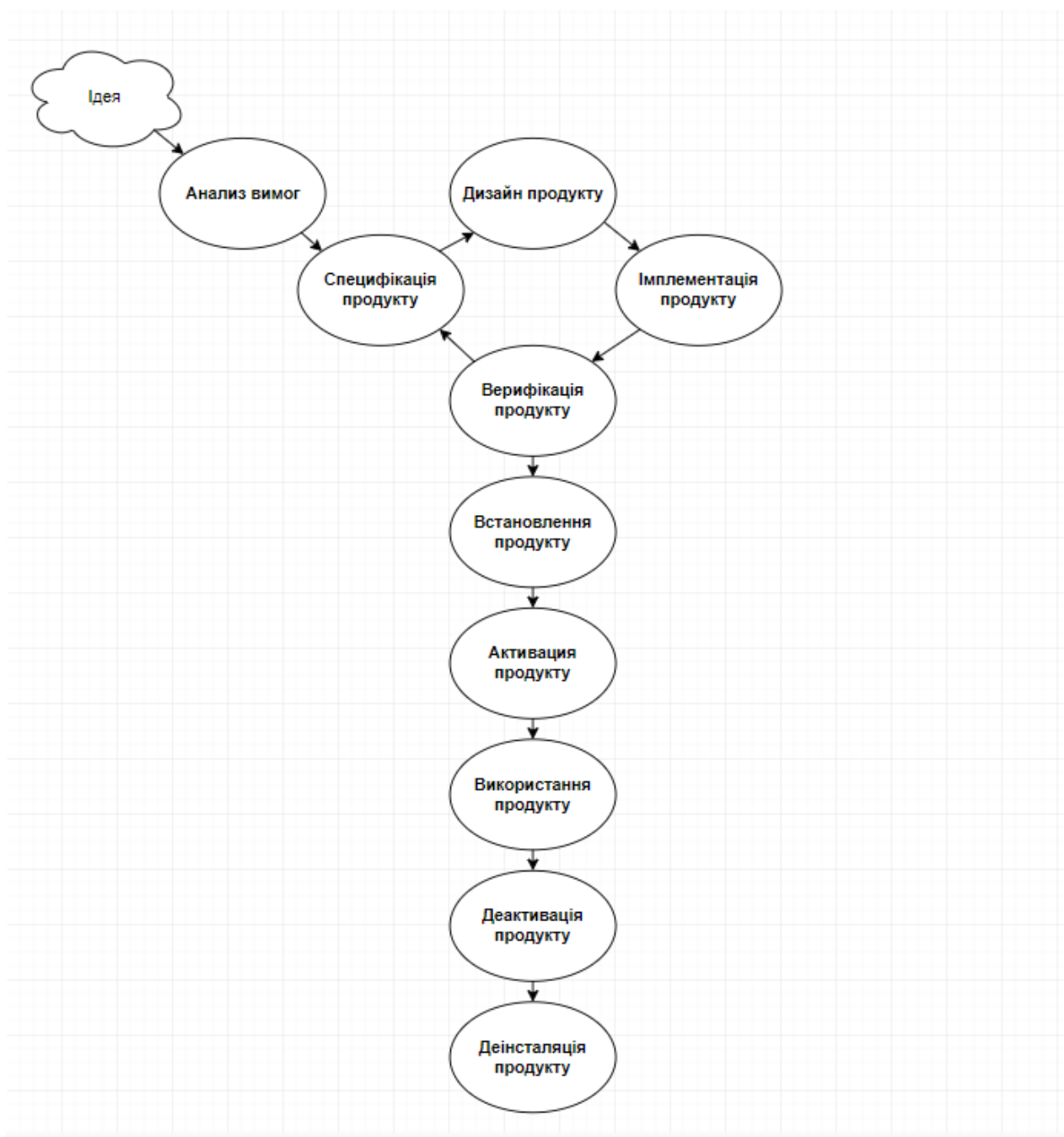
Однак незначна частина заходів з тестування послуг буде проводитися під час експлуатації послуги, яка включає фази після встановлення послуги і перед її деінсталяцією. Ми повернемося до цієї аномалії у третьому розділі.

Послуги можна надавати за допомогою архітектури «Інтелектуальна мережа».

На малюнку 2 зображена можлива конфігурація інтелектуальної мережі. У цій архітектурі виклик ІМ-(сервіс) виявляється в пункті перемикавання послуг (ППП). ППП звернеться до пункту управління сервісом (ПУС) для отримання інструкцій.

ПУС містить логіку обслуговування та сервісні дані, які використовуються для надання послуг ІМ. ПУС може отримати доступ до даних у сервісний пункт даних (СПД) безпосередньо або через мережу сигналізації.

					ЕліТ 6.172.00.02. 464 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9



Малюнок 2.1 - Тестування в життєвому циклі послуг

## 2 ТЕСТУВАННЯ ВИМОГ

### 2.1.Методи тестування вимог

Тестування вимог дозволяє зменшити кількість допрацювань і змін. Тобто ми заздалегідь придивились до того, що нам потрібно зробити, вивчили деякі речі, які в майбутньому підвищують ризик краху проєктів та унеможливають зміни. Це хороший процес, що дозволяє цілій команді ознайомитися і зрозуміти, що ж ми будемо будувати. Давайте поглянемо, які існують методи тестування вимог. Їх є три.

#### *Перевірка документації*

Цей метод дозволяє перевірити вимоги на правильність, повноту, відслідковуваність, важливість, зрозумілість, однозначність та вимірюваність. Виконується перевірка документації в достатньо простий спосіб. Тобто ми дивимося, читаємо і перед тим, як знайомитися з вимогами, задаємо собі правильні питання, звертаємо увагу на певні речі, про які я скажу пізніше, — і тим самим перевіряємо вимоги. Документація приймається замовниками, аналітиками, проєктними менеджерами, тестувальниками.

#### *Аналіз поведінки системи*

Це метод, коли ми формалізуємо події, тобто формуємо вимоги у форматі «вхід — вихід», «подія — наслідок», «умова — відповідь». Щось подається на вхід до системи, з системою щось відбувається, і вона має дати щось на виході. Найпоширеніший спосіб, у який це відбувається, — це або тест-кейси, або юз-кейси. Тест-кейси готують тестувальники, юз-кейси — зрозуміло, готують аналітики. Як і в будь-якого методу, у нього є і плюси, і мінуси.

#### *Прототипування*

Це ще один інструмент (і метод) тестування вимог.

Під прототипом ми розуміємо створення майбутньої моделі системи. Прототипування застосовується для перевірки вимог на повноту, правильність, реалізованість і на зручність використання користувацького інтерфейсу (якщо ми говоримо про розробку інформаційної системи).

Фактично, це чи не єдиний спосіб перевірити вимоги на реалізованість і на зручність користування інтерфейсом. Використовується аналітиками — найчастіше, а також архітекторами.

					ЕЛІТ 6.172.00.02. 464 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

## 2.2.Критерії оцінки якості вимог

### *Правильність*

Перший критерій — це правильність. Мається на увазі, що кожна вимога має бути чітко описана і точно розроблена. Тобто кожна вимога має правильно й точно описувати те, що має бути розроблено.

Недотримання цього критерію призводить до розробки неправильної функціональності.

### *Повнота вимог*

По-перше, всі вимоги мають бути задокументовані. І кожна вимога — містити всю необхідну інформацію для проектування, розробки і тестування рішень.

### *Зрозумілість*

Тобто вимоги, які інтерпретуються по-різному, мають бути або перероблені, або видалені.

### *Несуперечливість*

Тобто важливо впевнитись, що реалізація однієї вимоги не перешкоджає реалізації іншої вимоги.

### *Вимога не повинна визначати технічне рішення*

Вимога не повинна визначати технічне рішення. Тобто важливо розділяти нав'язування технічного рішення і відправні обмеження, які є в компанії.

### *Вимірюваність*

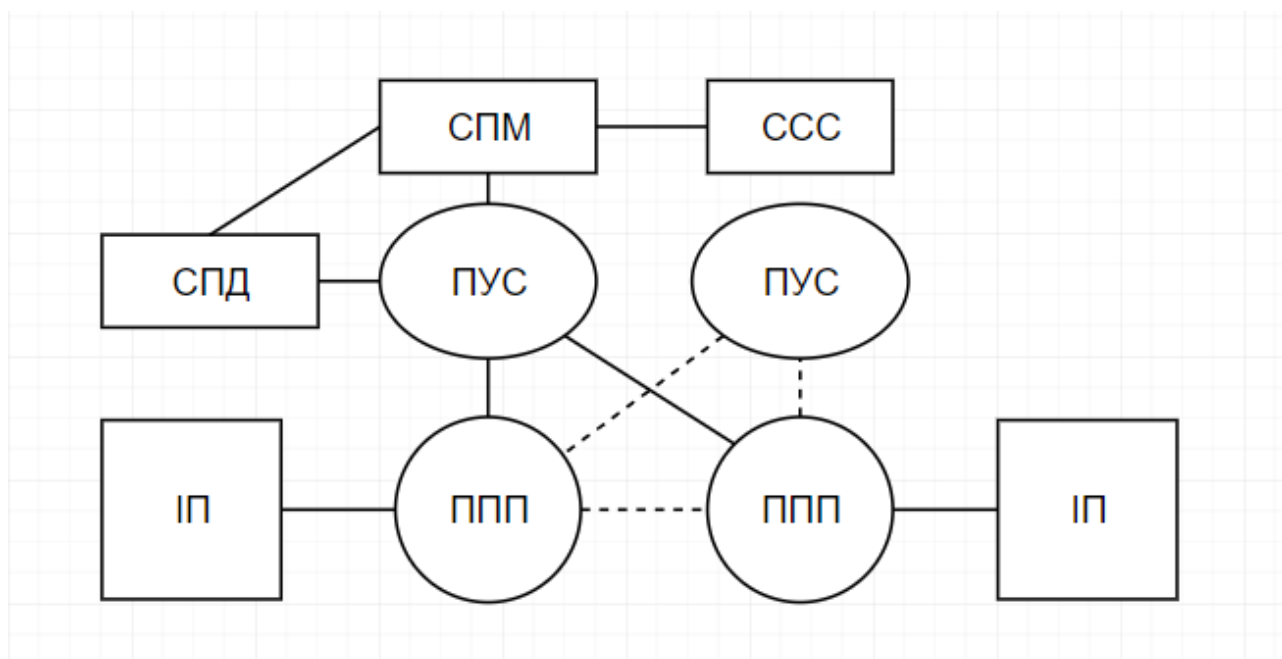
Вимога має бути сформульована таким чином, щоб можна було довести відповідність системи пред'явленій вимозі. Вимога не повинна містити невимірюваних характеристик або таких, які не піддаються тестуванню.

### *Досяжність*

Треба розуміти і планувати результат реалізації кожної вимоги. Мається на увазі, що якщо вимога технічно реалізовувана — то її реалізація вписується в часові й фінансові обмеження проекту.

					ЕЛІТ 6.172.00.02. 464 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

### 3 АРХІТЕКТУРА ІНТЕЛЕКТУАЛЬНОЇ МЕРЕЖІ



Малюнок 3.1. - Архітектура інтелектуальної мережі

**ССС** = Середовище створення сервісу

**ПУС/СПУ** = Сервісний пункт управління (містить службову логіку та службові дані використовується для надання послуг)

**ППП** = Сервісний перепискаючий пункт

**СПД** = Сервісний пункт даних

**СПМ** = Сервісний пункт менеджменту (здійснює контроль керування послугами, контроль надання послуг і контроль розгортання послуг)

**ІП** = Інтелектуальна периферія

Изм.	Лист	№ докум.	Подпись	Дата

Інтелектуальний периферійний (ІП) надає спеціалізовані ресурси для налаштування сервісу і підтримує гнучку взаємодію між користувачем і мережею. ІП можна використовувати, наприклад, для відтворення оголошень на кшталт «введіть свій PIN-код» у службі перевірки кредитної картки.

Сервісний пункт менеджменту здійснює контроль керування послугами, контроль надання послуг та контроль розгортання послуг. Середовище створення сервісу (ССС) використовується для визначення, розробки та тестування служби ІМ та введення її в СПУ.

#### 4 ТЕСТУВАННЯ ПОСЛУГ

На перший погляд, заходи щодо тестування, які необхідно проводити при впровадженні нової послуги, мають бути досить обмеженими. Можна припустити, що це проста перевірка доступності функціональності. Однак існує ряд причин для того, щоб застосовувати ретельну процедуру встановлення

- Реалізація нової послуги може вимагати внесення змін до платформи сервісу, або це може ненавмисно вплинути на функціональність платформи;

- Реалізація нової послуги може вимагати модифікації існуючих сервісів або це може ненавмисно змінити поведінку існуючих сервісів;

- Реалізація нової послуги може бути успішною, але збій може проявитися (через довгий час) після установки, через складні зв'язки різних елементів мережі

Для забезпечення правильної роботи сервісу необхідно виконати наступні 5 завдань:

1. Протестуйте сервісну платформу
2. Протестуйте "нову" послугу
3. Виконайте регресійні тести
4. Виконайте тести взаємодії із іншими послугами
5. Виконати сервісний менеджмент

Перш ніж встановити послугу в структуровану мережу ІМ, необхідно протестувати мережу; Це перше завдання у цій роботі називається тестуванням платформи.

					ЕЛІТ 6.172.00.02. 464 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

Під час другого етапу розглядається нова послуга. Необхідно перевірити, чи відповідає функціональність нової служби її специфікації.

Після встановлення та активації нової служби її потрібно перевірити (завдання 3), чи працюють уже встановлені служби. Потім, перш ніж запустити службу, завдання 4 перевіряє наявність небажаних взаємодій із службою.

Коли послугу введено в експлуатацію, все ще необхідно проводити тести під час керування послугою (завдання 5). Ці тести розглядають такі аспекти, як "чи працює ще послуга?", "Чи задовільна продуктивність? Як і раніше, задовільна" і т.д. У наступних підрозділах етапи тестування будуть розглянуті докладніше.

#### 4.1 Тестування платформи

Перш ніж встановлювати нові сервіси на сервісну платформу, сама платформа повинна бути протестована. У стандартах сервіс складається з так званих сервісних незалежних будівельних блоків (СНББ).

Можна сказати, що ці будівельні блоки є частиною платформи і повинні бути предметом тестування платформи. Можна перевірити СНББ у якомусь штучному середовищі (наприклад, симуляція). Проте перевірити один СНБ, реалізований в СПУ, буде важко. Для цієї мети може бути корисним «тестовий сервіс». Така послуга не обов'язково має функціональні можливості, які можуть використовувати кінцеві користувачі, але підтримує тестування відповідної частини логіки сервісу.

Зв'язок між мережевими елементами ППП і ПУС (а в деяких випадках і СПД) буде використовуватися за стандартизованим протоколом core-INAP. Підтверджуючи цей стек протоколів, стандарти перевірки відповідності розроблені (попередньо) в ETSI. Перш ніж розгортати ці елементи мережі в мережі, ці елементи повинні пройти ці тести на відповідність.

Крім функціональної поведінки, слід також перевірити нефункціональні аспекти, такі як продуктивність. Слід переконатися, що платформа демонструє достатню продуктивність для запуску потрібних служб. Це можна зробити за допомогою обчислень продуктивності (включаючи алгоритми передбачення), або за допомогою тестів продуктивності.

#### 4.2. Тестування нової послуги

Після створення послуги новий сервіс буде перевірено в тестовому середовищі.

					ЕлІТ 6.172.00.02. 464ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		15

Природно, це середовище має бути репрезентативним для робочого середовища. Звичайно, слід вжити заходів для того, щоб помилки, які виникають, не впливали на робоче середовище.

Можна застосувати кілька підходів, щоб перевірити, чи відповідає послуга специфікації, тобто потребам абонента послуги.

Ці підходи відрізняються вартістю, простотою використання та можливостями виявлення несправностей.

Проста методика застосовує просте наскрізне тестування поведінки служби. Більш просунутий підхід забезпечує моніторинг мережевого або протокольного трафіку в ряді стратегічних місць. Найбільш складне і вичерпне тестування може бути виконано шляхом автономного тестування окремого елемента мережі. Нижче ці підходи детально розглянуті.

## **5 ТЕСТУВАННЯ ЗА ЗНАННЯМ СИСТЕМИ**

### **5.1.Чорна коробка**

Тестування методом «чорної скриньки», також відоме як тестування, засноване на специфікації або тестування поведінки – техніка тестування, заснована на роботі виключно із зовнішніми інтерфейсами системи, що тестується.

Відповідно до ISTQB:

тестування чорної скриньки – це:

- Тестування, як функціональне, так і нефункціональне, що не передбачає знання внутрішнього пристрою компонента або системи.
- тест-дизайн, заснований на техніці чорної скриньки – процедура написання або вибору тест-кейсів на основі аналізу функціональної чи нефункціональної специфікації компонента чи системи без знання внутрішнього пристрою.

Чому саме «чорна скринька»? Тестована служба/сервіс/програма для тестувальника – як чорна непрозора скринька, змісту якої вона не бачить.

Метою цієї техніки є пошук помилок у таких категоріях:

- неправильно реалізовані або відсутні функції;

					ЕЛІТ 6.172.00.02. 464ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		16



- Помилки інтерфейсу;
- помилки у структурах даних чи організації доступу до зовнішніх баз даних;
- Помилки поведінки або недостатня продуктивність системи;

Таким чином, ми не маємо уявлення про структуру та внутрішній устрій системи. Потрібно концентруватись на тому, що програма робить, а не на тому, як вона це робить.

## 5.2. Біла коробка

Тестування методом білої скриньки (також: прозорої, відкритої, скляної скриньки; засноване на кодї або структурне тестування) – метод тестування програмного забезпечення, який передбачає, що внутрішня структура/пристрій/реалізація системи відома тестувальнику. Ми вибираємо вхідні значення, ґрунтуючись на знанні коду, який їх оброблятиме. Так само ми знаємо, яким має бути результат цієї обробки. Знання всіх особливостей програми та її реалізації – обов'язкові для цієї техніки. Тестування білого ящика – поглиблення у внутрішнє влаштування системи, межі її зовнішніх інтерфейсів.

Відповідно до ISTQB:

тестування білої скриньки – це:

- Тестування, засноване на аналізі внутрішньої структури компонента або системи.
- тест-дизайн, що базується на техніці білої скриньки – процедура написання або вибору тест-кейсів на основі аналізу внутрішнього пристрою системи або компонента.

Чому «біла скринька»? Тестована програма для тестувальника – прозора скринька, вміст якої він чудово бачить.

## 5.3. Сіра коробка

Тестування методом сірої скриньки – метод тестування програмного забезпечення, який передбачає комбінацію White Box і Black Box підходів. Тобто внутрішній пристрій програми нам відомий лише частково.

Передбачається, наприклад, доступ до внутрішньої структури та алгоритмів роботи програмного забезпечення для написання максимально ефективних

					Лист
					17
ЕліТ 6.172.00.02. 464 ПЗ	Изм.	Лист	№ докум.	Подпись	Дата

тест-кейсів, але саме тестування проводиться за допомогою техніки чорної скриньки, тобто з позиції користувача.

Цю техніку тестування називають методом напівпрозорого ящика: щось ми бачимо, а щось – ні.

## **6 РІВНІ ТЕСТУВАННЯ ЗА СТУПЕНЕМ ІЗОЛЬОВАНОСТІ КОМПОНЕНТІВ**

### **6.1. Модульне тестування**

Відноситься до тестів, які перевіряють функціональність певного розділу коду, зазвичай на функціональному рівні. В об'єктно-орієнтованому середовищі, це, як правило, тестування на рівні класу, а мінімальні модульні тести містять у собі конструктори та деструктори

Такі типи тестів зазвичай пишуться персоналом який розробляє мережу (стиль «білої скриньки»), щоб впевнитись, що дана функція працює так, як очікувалося. Одна функція може мати кілька тестів, щоб переглянути всі випадки використання коду. Модульне тестування саме по собі не може перевірити функціонування частини мережі, а використовується щоб гарантувати, що основні блоки мережі працюють незалежно один від одного.

Залежно від очікуваної організації розробки мережі, модульне тестування може включати статичний аналіз мережі, аналіз потоку даних аналізу метрик, експертні оцінки мережі, аналізу покриття коду та інші методи перевірки мережі.

### **6.2. Інтеграційне тестування**

Виходячи з відмінностей між модульним тестуванням та системним тестуванням, інтеграційне тестування є перехідним етапом між представленням мережі у вигляді окремих модулів до вигляду повністю функціональної системи.

Інтеграційне тестування – вид тестування, у якому відповідність вимог перевіряється інтеграція модулів, їх взаємодія між собою, і навіть інтеграція підсистем у одну загальну систему.

Для інтеграційного тестування використовуються компоненти, що вже перевірені за допомогою модульного тестування, які групуються в безлічі.

					ЕліТ 6.172.00.02. 464 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		18

Дані множини перевіряються відповідно до плану тестування, складеного для них, а об'єднуються вони через свої інтерфейси.

Оскільки модулі з'єднуються між собою за допомогою передбачених реалізацією інтерфейсів і в процесі тестування ми не маємо потреби розглядати внутрішню структуру компонентів, можна стверджувати, що інтеграційне тестування виконується методом «чорної скриньки».

Існує кілька підходів до інтеграційного тестування:

- Знизу вгору. Спочатку збираються та тестуються модулі найнижчих рівнів, а потім за зростанням до вершини ієрархії. Цей підхід вимагає готовності всіх модулів, що збираються, на всіх рівнях системи.
- Зверху вниз. Цей підхід передбачає рух із високорівневих модулів, а потім прямує вниз. При цьому використовуються заглушки для тих модулів, які знаходяться нижче за рівнем, але включення яких до тесту ще не відбулося.
- Великий вибух. Всі модулі всіх рівнів збираються до купи, а потім тестується. Даний метод заощаджує час, але вимагає ретельного опрацювання тест кейсів.

### 6.3. Наскрізне тестування служби

У той час як модульні тести дають певний ступінь упевненості, модульні та компонентні тести обмежені у своїх можливостях щодо забезпечення цілісного охоплення сервісу при розгортанні у виробничому середовищі. В результаті наскрізні (E2E) тести забезпечують покриття того, що, можливо, є найважливішим аспектом сервісу: що відбувається, коли користувачі фактично використовують сервіс.

Іншими словами, **наскрізні** тести перевіряють усі рівні у вашому додатку. Це не тільки код вашого зовнішнього інтерфейсу, але і всі пов'язані з ним серверні служби та інфраструктура, які більш репрезентативні для середовища, в якому будуть ваші користувачі. Тестуючи, як дії користувача впливають на вашу програму, тести E2E часто є ключем до більшої впевненості в тому, додаток працює правильно чи ні.

					ЕліТ 6.172.00.02. 464 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		19



Малюнок 6.1 - Підхід до тестування нової служби.

Изм.	Лист	№ докум.	Подпись	Дата

У цьому підході сервіс завантажується на (тестову) платформу. Тести контролюються на стороні користувача платформи, тобто шляхом моделювання поведінки користувача служби. Тести також спостерігаються в цьому інтерфейсі. Цей тип тестування можна назвати тестуванням «чорного ящика».

Наприклад, тестування служби переадресації викликів можна зробити за допомогою трьох телефонів. Припустимо, що телефон А переадресовує дзвінки на телефон Б. Згодом телефон С набирає номер А. Телефон Б дзвонить, і коли його підхоплюють, між С та Б встановлюється мовний канал. Перевагою цього підходу буде низька вартість та короткий час, необхідний для налаштування тестової конфігурації. Опис тестів можна навіть зробити оригінальною мовою. У цьому випадку технічні характеристики було б легко написати.

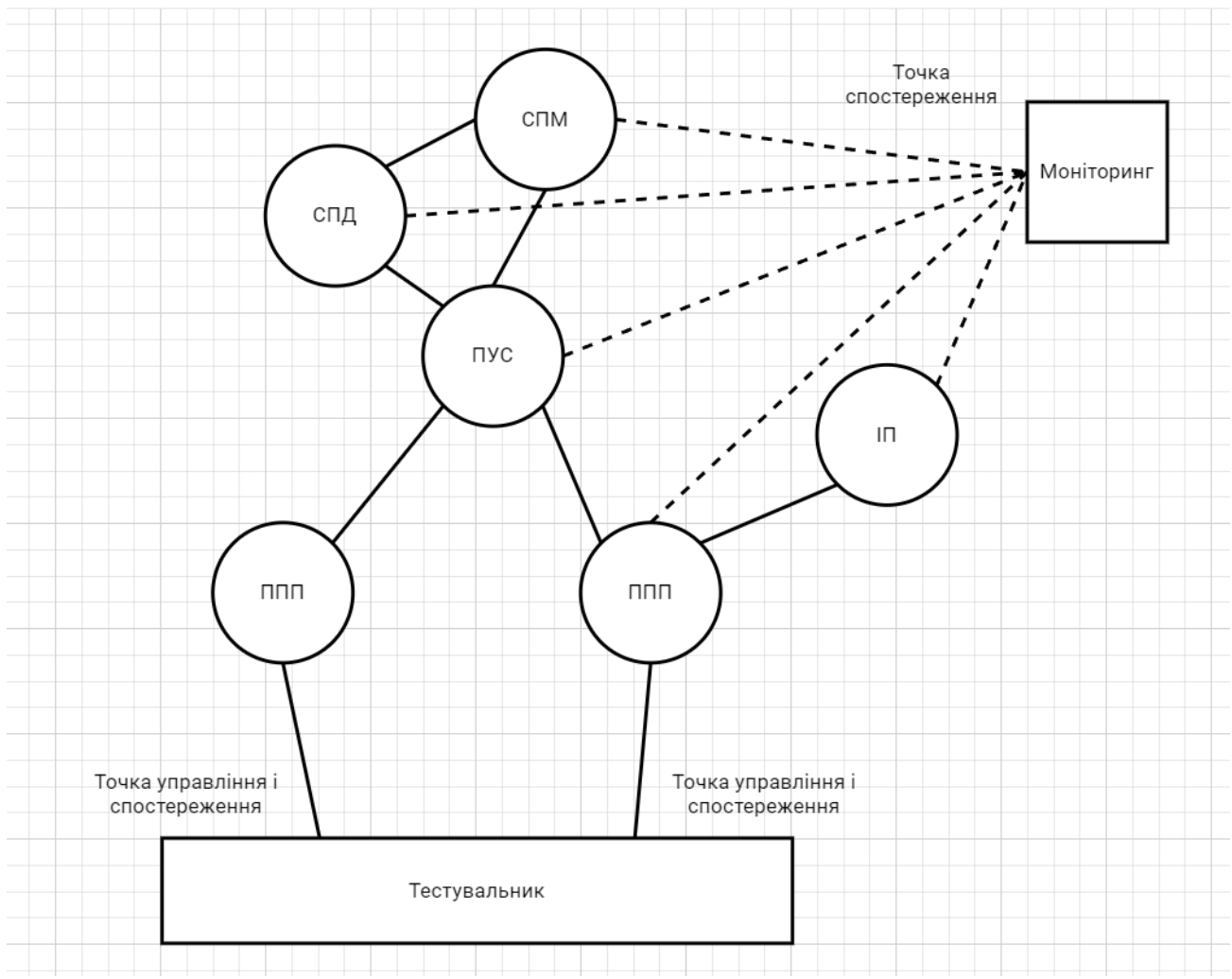
Основним недоліком такого підходу є обмежене охоплення тестів. Застосування термінального доступу означає, що частину логіки служби, яка має справу з винятковими ситуаціями, неможливо легко перевірити. Це викликано тим, що термінальний доступ не може або навряд чи може забезпечити збої в мережі або обмеження ресурсів. Ще одним недоліком цього підходу є відсутність локалізації несправності. Якщо виникають помилки, буде дуже важко визначити, яка частина мережі спричинила помилку.

### **6.3.1. Контрольований підхід до наскрізного тестування служби**

Одним із недоліків описаного вище підходу є можливість локалізації несправності. Більш досконалим підходом є так зване «наскрізне контрольне тестування» .

У цьому підході тести знову контролюються з точки зору користувача. Для спостереження за «внутрішньою поведінкою» мережі монітори можна встановити в кількох місцях. Очевидними точками спостереження для моніторного обладнання є лінії зв'язку.

Ці посилання часто базуються на міжнародних стандартах (core-INAP). Тому ці монітори є (або будуть) комерційно доступними. У деяких випадках мережеві елементи також забезпечать можливість моніторингу внутрішньої поведінки елементів мережі.



Малюнок 6.2. - Підхід до наскрізного тестування, що відстежується

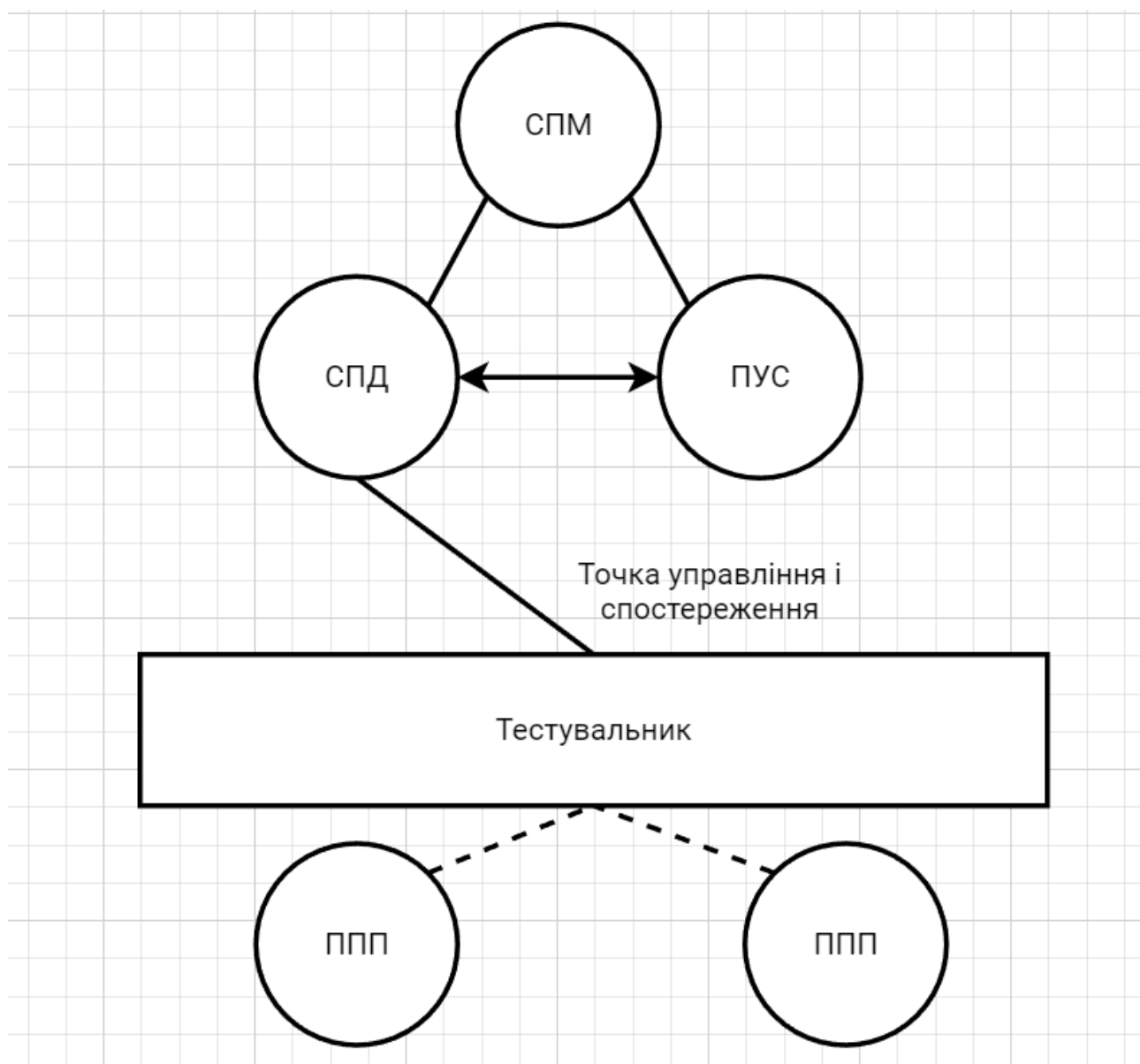
Для опису тесту можна використовувати два підходи. Один з підходів полягає у використанні того ж типу специфікації тесту, що й у тестовому підході 'end-to-end'. У цьому випадку монітори використовуються лише тоді, коли виникають помилки, щоб знайти джерела помилок.

Інший підхід - всі очікувані спостереження, видимі на моніторах, повністю описані. У специфікації тесту буде описано не тільки зовнішню поведінку (як у підході тестування end-to-end), опис внутрішніх подій у ППП і ПУС, повідомлення, якими обмінюється інтерфейс, і запити в також слід уточнити СПД. З одного боку, це дає детальну інформацію про те, що перевіряється. З іншого боку, специфікація тесту, як правило, стає дуже великою та важкою для читання.

Розглядаючи переваги/недоліки, використовуючи підхід контрольованого наскрізного тестування, отримуємо інформацію про внутрішні події ПП. Це дає можливість знайти джерело помилки. Порт тестування (термінальний доступ) залишається таким же, як описано для попереднього підходу. Тому потужність тестування все одно буде обмежена можливостями інтерфейсу користувача. Ми будемо називати цей тип тестування тестуванням «сірої коробки».

					ЕЛІТ 6.172.00.02. 464 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		23

#### 6.4.Тестування відповідності послуг



Малюнок 6.3 - Підхід тестування відповідності послуг

Изм.	Лист	№ докум.	Подпись	Дата

ЕЛІТ 6.172.00.02. 464 ПЗ

Лист

24



Тестування на відповідність - елемент оцінки відповідності, також відомий як тестування на відповідність, або типове тестування - це тестування або інші дії, які визначають, чи відповідає процес, продукт або послуга вимогам специфікації, технічний стандарт, контракт або правило. Тестування часто є або логічним тестуванням, або фізичним тестуванням. Крім простої відповідності, можуть застосовуватися інші вимоги до ефективності, сумісності або відповідності. Тестування на відповідність може проводитися виробником продукту або послуги, що оцінюється, користувачем або незалежною організацією, яка іноді може бути автором використововуваного стандарту. Якщо тестування супроводжується сертифікацією, продукти або послуги можуть рекламуватися як сертифіковані відповідно до згаданого технічного стандарту. Виробники та постачальники продуктів і послуг покладаються на таку сертифікацію, включаючи лістинг на веб-сайті органу сертифікації, щоб гарантувати якість для кінцевого користувача і те, що конкуруючі постачальники знаходяться на одному рівні.

Використовуючи цей метод, тести в основному зосереджені на одному елементі мережі, у цій конфігурації ПУС. Спеціальне програмне забезпечення для служби значною мірою знаходиться в ПУС. Частина служби в ППП в більшості випадків вплине лише на деякі налаштування параметрів (наприклад, оновлення точок виявлення тригерів)

На малюнку 5 ПКС розташований на лінії зв'язку, яка демонструє стандартизований протокол. Цей доступ контролює тести. Використання інтерфейсу надає широкі можливості для тестування, які виходять за межі рівня дійсного тестування поведінки. Інтерфейс також надає можливості для виконання тестів на неправильну поведінку та кодування даних.

## **6.5. Визначення пріоритету у виборі підходу тестування**

Вибір того, який підхід слід використовувати, залежить від ряду факторів. З одного боку, ми маємо мати справу з такими факторами, як «необхідна якість обслуговування»/ «необхідний рівень впевненості».

З іншого боку, ми обмежені наявними ресурсами та часом. На даний момент є мало обладнання для тестування відповідності, і немає повних тестових специфікацій. Крім того, відсутність досвіду в тестуванні відповідності послуг може призвести до неприйнятних затримок у жорсткому графіку тестування. Тому здається розумним почати набирати досвід у сфері тестування сервісів, застосовуючи наскрізні тести вручну.

					ЕЛІТ 6.172.00.02. 464 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		25

На практиці процес можна покращити за рахунок автоматизації та використання більш потужних методів. Генерацію та виконання тестів можна було б автоматизувати, а методи тестування могли б перейти від наскрізного тестування до тестування відповідності послуг.

## 7 РЕГРЕСІЙНІ ТЕСИ

**Регресійне тестування** – це вид тестування спрямований на перевірку змін, зроблених в додатку або середовищі (лагодження дефекту, злиття коду, міграція на іншу операційну систему, базу даних, веб сервер або сервер додатки), для підтвердження того факту, що існуюча раніше функціональність працює як і раніше.

Регресійними можуть бути як функціональні, так і нефункціональні тести. Як правило, для регресійного тестування використовуються тест кейси, написані на ранніх стадіях розробки і тестування. Це дає гарантію того, що зміни в новій версії програми не пошкодили вже існуючу функціональність.

Рекомендується робити автоматизацію регресійних тестів, для прискорення подальшого процесу тестування і виявлення дефектів на ранніх стадіях розробки програмного забезпечення.

Навіть якщо щойно встановлена служба ще не активована, вона може заважати функціональності інших служб; додавання нової послуги до встановленої бази могло спричинити зміни в існуючих сервісах. Тому необхідно проводити тестування і на існуючих сервісах. Такі типи тестів називаються регресійними.

На малюнку 6 зображено сценарій для тестів регресії служби. Перед початком регресійних тестів новий сервіс слід повністю протестувати за допомогою одного з підходів, розглянутих у попередніх параграфах. Щоб переконатися, що інші служби все ще працюють правильно, потрібно виконати регресійні тести, виконавши підмножину функціональних тестів для кожної служби.

Після виконання всіх регресійних тестів стандартний набір регресійних тестів може бути розширений підмножиною функціональних тестів для нової служби.

З малюнка 6 видно, що регресійне тестування може включати серйозну проблему управління. Як показано на малюнку, набір стандартних регресійних тестів буде збільшуватися з впровадженням кожної нової служби.

Для зменшення набору регресійного тесту можна використовувати два альтернативні підходи.

Замість того, щоб використовувати підмножину функціональних тестів, можна припустити, що можна було б використовувати якийсь «тестовий сервіс» для тесту регресії. Цей сервіс повинен містити функції всіх уже встановлених служб. Чи можливий такий підхід, сказати важко. Іншою можливістю може бути використання категорій і пріоритетів для вибору набору регресії. Цей принцип вимагає, щоб усі функціональні тести були класифіковані та визначені пріоритети (наприклад, відповідно до впливу збою). Хоча повний набір доступних тестів збільшується з додаванням нових сервісів, можна було б вибрати постійну кількість регресійних тестів на основі категорій і пріоритетів.

## 8 ТЕСТИ НА ВЗАЄМОДІЮ

Тестування взаємодії - вид тестування, націлений на оцінку якості взаємодії компонентів програмної системи або всієї програми з іншими компонентами або програмним забезпеченням.

Даний вид тестування набув серйозної актуальності зі зростанням популярності мережевих додатків, основною запорукою успіху яких є швидка, надійна і безпечна взаємодія між іншим ПЗ в мережі.

Особливе значення тестування взаємодії має таких галузей мережевих технологій, як:

- Потоки даних (різні мультимедіа, мова, інформація).
- Вузькосмугові та широкосмугові мережі.
- Мобільні та фіксовані мережі.
- Мережі з тимчасовим розділенням каналів та IP-мережі.

Суть даного виду тестування дозволяє включати тестування сумісності та інтеграційне тестування, які також відповідальні за якість як взаємодії деякої компоненти в рамках системи, так і взаємодія розроблюваного ПЗ з іншими додатками.

У попередніх параграфах було зазначено, що після тестування сервісу на функціональність необхідно виконати регресійні тести, щоб переконатися, що вже встановлені сервіси все ще працюють правильно. Тому виконується підмножина функціональних тестів кожної встановленої служби. Однак цього недостатньо для розгортання служби в мережі.

									Лист
									27
Изм.	Лист	№ докум.	Подпись	Дата					

Коли кожна окрема служба працює правильно, це ще не дає гарантії, що служби, що працюють одночасно, будуть правильно працювати разом. Служби можуть конфліктувати один з одним і навіть з (іншими екземплярами) самими собою.

Можна виділити два типи сервісної взаємодії. Перший тип притаманний логіці служби і може (повинен) бути виявлений у специфікації. Розумні алгоритми можуть виявляти ці взаємодії служби, а зміна служби може запобігти взаємодії. Однак другий тип залежить від значень динамічних даних служби, і його важко розпізнати під час специфікації.

На практиці, однак, поки що неможливо виявити небажані взаємодії сервісів у специфікаціях. Тому має бути виконано якийсь тест «сервісної взаємодії», щоб запобігти взаємодії з мережею. Ці тести можна виконувати або в середовищі моделювання, або в реальному тестовому середовищі. Одночасне виконання функціональних тестів може виявити другий вид взаємодій. Коли всі функціональні тести, виконані одночасно, проходять, можна стверджувати, що мережа вільна від взаємодій сервісів.

## **9 ТЕСТИ УПРАВЛІННЯ СЕРВІСОМ**

Після того, як служба запрацює, тестові дії все ще повинні виконуватися. Ці тестові дії належать до категорії керування помилками. У мережі керування телекомунікаціями визначено управління помилками. Управління несправністю (або обслуговуванням) складається з набору функцій, які дають змогу виявляти, ізолювати та виправляти ненормальну роботу телекомунікаційної мережі та її середовища.

## **10 ФУНКЦІОНАЛЬНЕ І НЕФУНКЦІОНАЛЬНЕ ТЕСТУВАННЯМ**

### **10.1.Що таке функціональне тестування?**

Функціональне тестування — це тип тестування, яке перевіряє, що кожна функція програмного додатка працює відповідно до специфікації вимог. Це тестування в основному включає тестування чорного ящика, і воно не стосується вихідного коду програми.

Кожна функціональність системи перевіряється шляхом надання відповідних даних, перевірки результатів і порівняння фактичних результатів з очікуваними. Це тестування включає перевірку користувальницького інтерфейсу, API, бази даних, безпеки, програм клієнта/сервера та функціональності послуги, що тестується.

									Лист
Изм.	Лист	№ докум.	Подпись	Дата					28

Тестування можна проводити як вручну, так і за допомогою автоматизації

Види не функціонального тестування:

- Юнітне тестування
- Тестування на дим
- Прийняття користувача
- Інтеграційне тестування
- Регресійне тестування
- Локалізація
- Глобалізація
- Сумісність

### 10.2.Що таке нефункціональне тестування?

Нефункціональне тестування — це тип тестування для перевірки нефункціональних аспектів (продуктивність, зручність використання, надійність тощо) послуги. Воно спеціально розроблено для перевірки готовності системи за нефункціональними параметрами, які ніколи не розглядаються функціональним тестуванням.

Хорошим прикладом нефункціонального тесту може бути перевірка кількості людей, які можуть одночасно увійти мережу.

Нефункціональне тестування так само важливе, як і функціональне, і впливає на задоволеність клієнта.

Види не функціонального тестування:

- Тестування навантаження
- Тестування відмовостійкості
- Тестування відповідності
- Юзабіліті-тестування
- Стрес-тестування
- Тестування ремонтпридатності
- Тестування масштабованості

## 11 ТЕСТУВАННЯ ЕФЕКТИВНОСТІ

Тестування ефективності – це комплекс типів тестування, метою якого є визначення працездатності, стабільності, потреби ресурсів та інших атрибутів якості додатків в умовах різних сценаріїв використання та навантажень.

Тестування ефективності дозволяє знаходити можливі вразливості та недоліки в системі з метою запобігання їх негативного впливу на роботу послуги в умовах використання. Необхідні параметри роботи системи в певному середовищі можна перевірити за допомогою:

- Визначення кількості користувачів послуги.
- Вимірювання часу виконання різних операцій системи.
- Визначення ефективності послуги при різних ступенях навантажень
- Визначення допустимих границь програми програми при різних рівнях навантажень.

Залежно від параметрів, які нам необхідно протестувати, тестування продуктивності ділиться на типи:

- Навантажувальне тестування– тестування часу відгуку програми на запити різних типів, з метою переконатися, що програма працює відповідно до вимог при звичайному навантаженні користувача.
- Стрес-тестування– тестування працездатності програми при навантаженнях, що перевищують користувацькі у кілька разів. При стрес-тестуванні (найчастіше, тільки за нього) ми можемо отримати реальні дані меж продуктивності програми, досліджувати здатність програми обробляти винятки, її стабільність і стійкість. Саме у значно збільшеному навантаженні на додаток і полягає різниця між тестуванням продуктивності та стрес тестуванням.
- Тестування стабільності або напрацювання на відмову досліджує працездатність програми за тривалої роботи в часі, за нормальної програми навантаження.
- Об'ємне тестування – тестування проводиться зі збільшенням не навантаження та часу роботи, а кількості даних, які використовуються і використовуються в додатку.

									Лист
									30
Изм.	Лист	№ докум.	Подпись	Дата					

Дуже часто при визначенні тестування продуктивності та його типах приходять до помилкового розуміння та плутанини даних термінів. Щоб уникнути цього і закріпити отримані знання, підіб'ємо підсумок. Отже, тестування продуктивності – це перевірка таких нефункціональних вимог, як продуктивність та працездатність програми за різних навантажень та умов. Залежно від досліджуваної характеристики програми ми можемо виділити такі типи тестування як:

- Навантажувальне тестування (продуктивність за нормальних умов).
- Стрес-тестування (працездатність, продуктивність та характеристики програми при екстремальних навантаженнях).
- Тестування стабільності (за тривалої роботи).

## 12 ТЕСТУВАННЯ НА ВІДМОВУ ТА ВІДНОВЛЕННЯ

Тестування на відмову та відновлення – вид тестування, основною місією якого є визначення здатності мережі до опору та відновлення після збоїв у роботі, що виникли як усередині послуги, так і від інших програмно-незалежних факторів (апаратура, мережа та і т.д.).

Даний вид тестування має дуже специфічний (порівняно з іншими видами) підхід до виконання тестів, оскільки об'єктами дослідження є:

- Поведінка ПЗ при перериванні обробки даних.
- У разі втрати мережі.
- У разі вимкнення електрики (на стороні клієнта або сервера).
- Втрата підключення носіїв даних.

Отже, і різні сценарії тестування розробляються спираючись на вищезгадані фактори впливу на здатність ПЗ до відновлення після збою.

Тестування на відмову та відновлення особливо актуальне при розробці систем, які мають працювати протягом тривалого часу, аж до 24\*7. Від можливості такого ПЗ відновлювати працездатність після непередбачуваної ситуації, а також мінімізувати втрати даних залежатиме не тільки репутація компанії-розробника, а часом і щось більше, ніж гроші.

									Лист
									31
Изм.	Лист	№ докум.	Подпись	Дата	ЕЛІТ 6.172.00.02. 464 ПЗ				

При моделюванні ситуації збою оцінюється як ступінь втрати даних (чи знаходиться вона в межах допустимого), так і здатність системи журналювати всі транзакції та їх статус виконання.

### **13 ТЕСТУВАННЯ ЗА ВИКОРИСТАННЯ СЦЕНАРІЮ**

На етапі контролю якості реалізованої функціональності використовується тестова документація, в якій записані стандартні та альтернативні сценарії роботи з програмою, що використовуються під час тестування чергової версії програми.

Тестова документація складається з тестових сценаріїв, тобто спеціальним чином розробленого опису послідовності дій у системі та очікуваної поведінки. Тестові сценарії використовуються для різних видів ручного тестування:

#### **13.1.Вільне тестування**

Вільне тестування (ad-hoc testing) – це вид тестування, який виконується без підготовки до тестування продукту, без визначення очікуваних результатів, проектування тестових сценаріїв. Це неформальне, імпровізаційне тестування. Воно не вимагає ніякої документації, планування, процесів, яких слід дотримуватися при тестуванні. Такий спосіб тестування в більшості випадків дає більшу кількість заведених звітів про помилку. Це обумовлено тим, що тестувальник на перших кроках приступає до тестування основної функціональної частини продукту і виконує як позитивні, так і негативні варіанти можливих сценаріїв.

Найчастіше таке тестування виконується, коли власник товару не володіє конкретними цілями, проектною документацією та раніше поставленими завданнями. При цьому тестувальник покладається на своє загальне уявлення про продукт, порівняння зі схожими продуктами, власний досвід. Однак при тестуванні ad-hoc, важливо володіти загальною інформацією про продукт, особливо якщо проект дуже складний і великий. Тому потрібно гарне уявлення про цілі проекту, його призначення та основні функції і можливості.

#### **13.2.Дослідницьке тестування**

Дослідницьке тестування – це одночасне вивчення програмного продукту, проектування тестів і їх виконання. Це неформальний метод проектування тестів, при якому тестувальник активно контролює проектування тестів в той

					ЕЛІТ 6.172.00.02. 464 ПЗ	Лист 32
Изм.	Лист	№ докум.	Подпись	Дата		



час, як ці тести виконуються, і використовує отриману під час тестування інформацію для проєктування нових тестів.

Якщо кожен наступний тест, який виконує тестувальник, вибирається за результатами попереднього тесту, це означає, що ми використовуємо дослідницьке тестування.

Головне, що потрібно пам'ятати про дослідницьке тестування, це те, що саме по собі воно не є методикою тестування. Це, скоріше, підхід, який можна застосувати до будь-якого виду тестування. Ще один важливий момент полягає в тому, що дослідницьке тестування – це не тільки виконання тестів.

Тестувальники можуть застосовувати дослідницький підхід і при розробці нових тестів на початку ітерації, і при аналізі вже завершених тестів. Також, дослідницьке тестування не повинно виконуватися недбало, в поспіху і без підготовки. Дослідницький підхід може вимагати дуже ретельної і тривалої підготовки до певних тестів, а накопичені за багато років знання і вміння тестувальника, що застосовує даний підхід, – це часто невидима, але важлива форма підготовки. Дослідницьке тестування може проводитися вручну, а може здійснюватися з широким застосуванням засобів автоматизації, тобто будь-яких допоміжних інструментів тестування.

## 14 УПРАВЛІННЯ ПОМИЛКАМИ TMN

У стандарті TMN управління несправністю поділяється на наступні чотири типи дій:

- Спостереження тривоги (деякий елемент мережі не працює)
- Локалізація несправності (де сталася несправність?)
- Виправлення несправності (усунення несправності)
- Тестування (чи служба все ще працює)

Оскільки ця стаття спрямована на тестування сервісу, перші три типи не будуть обговорюватися. Ці типи дій враховують помилки, які призводять до сповіщень. Цю помилку згодом локалізують і виправляють. У цьому звіті ми зосереджуємось на типі тестування керування несправностями. Визначенно два типи тестування. Або TMN запитує мережевий елемент для тестування служби та надсилає звіт, або TMN запитує доступ до елемента мережі та виконує деяке тестування самостійно.

									Лист
									33
Изм.	Лист	№ докум.	Подпись	Дата					

«Ці типи тестування можуть бути використані або для профілактики або для визначення несправності Хоча управління послугами дуже важливе для підтримки служби, на жаль, ця область ще досі не вивчена.

## 15 ВИСНОВКИ

У цьому документі представлена основа для тестування сервісів. Перш ніж тестувати саму нову послугу, слід переконатися, що платформа працює правильно. Новий сервіс потрібно протестувати в три етапи. У цьому фреймі доступно кілька можливостей, щоб переконатися, що як нова служба, так і встановлена база послуг, працюють правильно. Нарешті, коли служба працює, можна виконати тести, щоб переконатися, що служби все ще працюють, і знайти джерела помилок у разі збоїв.

Хоча область тестування служби є відносно новою, ряд уже доступних методів тестування можна повторно використовувати для тестування служби. Вже для тестування базових послуг телефонії використовуються генератори викликів і звичайне телефонне обладнання. Це обладнання також можна використовувати повторно для тестування.

Також для тестування послуг можна застосувати концепції та методологію зі світу тестування відповідності. Справді новим аспектом тестування служби буде час, доступний для тестування. Для досягнення цілей ринку зростатиме тиск на швидке впровадження послуги. Тому час, що залишився для тестування служби, буде все більше і більше під тиском. Цей фактор часу буде ставати все більш важливим у виборі методології тестування. Тому для дослідницької спільноти тестування є завданням знайти ефективні рішення

					ЕліТ 6.172.00.02. 464 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		34

## 16 ЛІТЕРАТУРА

1. Eurescom-P103 (1992) 'Evolution of the Intelligent Network, deliverable No.2'.
2. H. Velthuisen et al. (1993), ' A feature Interaction Benchmark in IN and beyond', Bellcore
3. Memorandum TM-SV-021982  
CCITT, Q1201 (1992), 'Principles, of Intelligent Network Architecture'.
4. ISO/IEC 9646: April 1994, Information Technology, OSI Conformance Testing  
a. Methodology and Framework
5. S.P. van de Burgt, IKroon, E.Kwast, H.I.Wilts (1990), The RNL Conformance Kit, Protocol Test Systems, IFIP.
6. P.Razol et al (1994), 'Service Interaction and Test Generation', proceedings ICIN  
a. conference.
7. R.I. Heijink (1994), 'FAITH, a general purpose protocol test system for ISDN',  
Computer Networks and ISDN Systems (1994),1581-1593.
8. L. Gabrielli, P. Marchese, (1992) 'Intelligent Network Service Testing', ICCS,  
Conference on IN, Tampa, USA.
9. E. Anders, I. Ellsberger, A. Wiles (1994), Experiences with Computer Aided Test  
Generation, proceedings IWP1S conference 1994.

					ЕЛІТ 6.172.00.02. 464 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		35