

МАГІСТЕРСЬКА РОБОТА

МР. ШМ - 10.00.00.000 ПЗ

Група ШМ-23-1

Грицак Богдан

2024

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Грицак Богдан Іванович

(прізвище, ім'я, по батькові)

УДК 004.94
(індекс)

МАГІСТЕРСЬКА РОБОТА

Методика системного підходу до проектування на основі інженерії

ВИМОГ

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Грицак Б.І.

(підпис, ініціали та прізвище здобувача освітнього ступеня)

Науковий керівник Юрчишин Володимир Миколайович, д.т.н., професор

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Допущено до захисту

Завідувач кафедри

доц. Бандура В.В.

(посада) (підпис) (дата) (ініціали та прізвище)

Нормоконтроль

доц. Вовк Р.Б.

(посада) (підпис) (дата) (ініціали та прізвище)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Івано-Франківськ – 2024

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень магістр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедрою

ІІЗ

доц.

В.В. Бандура

“ 04 ” вересня 2024 р.

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Грицаку Богдану Івановичу

(прізвище, ім'я, по-батькові)

1. Тема магістерської роботи “Методика системного підходу до проектування на основі інженерії вимог”

керівник проекту (роботи) Юрчишин Володимир Миколайович, д.т.н., професор

затверджені наказом закладу вищої освіти від “ 22 ” листопада 2024 р. № 781/7

2. Строк подання студентом проекту (роботи) 15 грудня 2024 р.

3. Вихідні дані до проекту (роботи) Теоретичні концепції та формальні моделі побудови та функціонування інформаційних технологій інженерії вимог

4. Зміст розрахунково - пояснювальної записки(перелік питань, які потрібно розробити)

1. Дослідження предметної області проектування систем на основі інженерії вимог

2. Моделі, методи та підходи до проектування на основі інженерії вимог

3. Представлення інженерної моделі вимог на основі системного підходу

4. Реалізація методики системного підходу до проектування на основі інженерії вимог

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1. Проблеми в інженерних проектах (рис. 1.1)

2. Представлення фаз проекту (рис. 1.2)

3. Фрагмент ініціації проекту та етапу розробки концепції (рис. 1.3)

4. Приклад створення вимог (рис. 1.4)

5. Приклад помилкового розуміння інженерії вимог (рис. 1.5)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата
Перевірка на плагіат	доц., к.т.н. Вовк Р.Б.	

7. Дата видачі завдання 04 вересня 2024 р.

Керівник _____

(підпис)

Завдання прийняв до виконання _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір і вивчення літератури по темі магістерської роботи	15.09.2024	виконано
2	Аналіз концепцій та алгоритмів предметної області	29.09.2024	виконано
3	Дослідження предметної області проектування систем на основі інженерії вимог	15.10.2024	виконано
4	Моделі, методи та підходи до проектування на основі інженерії вимог	08.11.2024	виконано
5	Представлення інженерної моделі вимог на основі системного підходу	20.11.2024	виконано
6	Реалізація методики системного підходу до проектування на основі інженерії вимог	01.12.2024	виконано
7	Затвердження пояснювальної записки роботи завідувачем кафедри	15.12.2024	виконано

Студент – магістр _____

(підпис)

Керівник роботи _____

(підпис)

АНОТАЦІЯ

Магістерська робота: 79 с., 28 рис., 4 табл., 51 джерело.

Тема: Методика системного підходу до проектування на основі інженерії вимог

Об'єкт дослідження: процес проектування складних програмних та інформаційних систем.

Мета роботи: розробка та обґрунтування методики системного підходу до проектування складних систем на основі інженерії вимог, яка забезпечує підвищення якості проектних рішень та відповідності вимогам замовників.

Предмет дослідження: методи, методика та методологія системного підходу до проектування на основі інженерії вимог.

Результати дослідження

В роботі використано методи системного аналізу для дослідження предметної області; порівняльний аналіз для оцінки існуючих моделей інженерії вимог; методи моделювання для формалізації вимог; методи концептуального проектування для розробки пропонованої методики; експериментальні методи для оцінки ефективності розробленої методики

Висновок

Запропоновано новий підхід до проектування складних систем на основі інженерії вимог, що включає інтеграцію системного підходу та сучасних методів формалізації вимог і розроблено удосконалену модель формалізації вимог, що забезпечує їхню узгодженість та однозначність.

СИСТЕМНИЙ ПІДХІД, ІНЖЕНЕРІЯ ВИМОГ, ПРОЄКТУВАННЯ СИСТЕМ, ФОРМАЛІЗАЦІЯ ВИМОГ, МЕТОДОЛОГІЯ ПРОЄКТУВАННЯ, ФРЕЙМВОРК ZACHMAN, МОДЕЛЮВАННЯ ВИМОГ

ABSTRACT

Master Thesis: 79 pp., 28 fig., 4 tab., 51 sources.

Thesis Subject: Methodology of a system approach to design based on requirements engineering

Object of research: the process of designing complex software and information systems.

The purpose of the work: development and substantiation of the methodology of the system approach to the design of complex systems based on requirements engineering, which ensures the improvement of the quality of project solutions and compliance with customer requirements.

Research subject: methods, methodology and methodology of a system approach to design based on requirements engineering.

Research results

The work uses the methods of system analysis to study the subject area; benchmarking to evaluate existing requirements engineering models; modeling methods to formalize requirements; methods of conceptual design for the development of the proposed methodology; experimental methods for evaluating the effectiveness of the developed methodology

Conclusion

A new approach to the design of complex systems based on requirements engineering is proposed, which includes the integration of a system approach and modern requirements formalization methods, and an improved model of requirements formalization, which ensures their consistency and ambiguity, is developed.

SYSTEMS APPROACH, REQUIREMENTS ENGINEERING, SYSTEMS DESIGN, REQUIREMENTS FORMALIZATION, DESIGN METHODOLOGY, ZACHMAN FRAMEWORK, REQUIREMENTS MODELING

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	9
ВСТУП.....	10
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ПРОЕКТУВАННЯ СИСТЕМ НА ОСНОВІ ІНЖЕНЕРІЇ ВИМОГ	12
1.1. Аналіз предметної області дослідження	12
1.2. Опис особливостей фаз проекту.....	14
1.3. Аналіз проблеми щодо розробки вимог.....	17
1.4. Огляд сутності інженерії вимог	19
1.4.1. Проблеми в розробці вимог.....	23
1.5. Інструменти для управління інженерією вимог	24
Висновки до розділу	30
РОЗДІЛ 2. МОДЕЛІ, МЕТОДИ ТА ПІДХОДИ ДО ПРОЕКТУВАННЯ НА ОСНОВІ ІНЖЕНЕРІЇ ВИМОГ	32
2.1. Порівняльний аналіз моделей інженерії вимог	32
2.1.1. Лінійна модель Маколея.....	32
2.1.2. Спіральна модель процесу розробки вимог	33
2.1.3. Загальна модель Pandey & Suman	35
2.1.4. Модель вимог Sommerville.....	36
2.2. Представлення інженерної моделі вимог на основі системного підходу.....	37
2.2.1. Пропонована інженерна модель вимог.....	38
2.2.2. Основи системного підходу	43
2.2.3. Модельно керований системний підхід.....	44
2.2.4. Аналіз інженерних методів вимог	45
2.3. Дослідження та опис сутності методології.....	48
2.3.1. Опис фреймворку Zachman	48

2.3.2. Адаптований фреймворк Zachman.....	49
2.3.3. Фреймворк Zachman як продукт	51
2.4. Формалізація вимог.....	52
2.4.1. Логічний підхід до формалізації вимог	54
2.5. Розширена методологія моделювання систем.....	55
Висновки до розділу	57
РОЗДІЛ 3. РЕАЛІЗАЦІЯ МЕТОДИКИ СИСТЕМНОГО ПІДХОДУ ДО ПРОЕКТУВАННЯ НА ОСНОВІ ІНЖЕНЕРІЇ ВИМОГ	59
3.1. Особливості пропонованої методології	59
3.2. Представлення та опис етапів методології	61
3.3. Структуризація та формалізація вимог	63
3.4. Розробка концепції на основі інженерії вимог	67
Висновки до розділу	71
ВИСНОВКИ	73
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	75

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

RE - Requirement Engineering
SYSML - System Engineering Modeling Language
SYSMOD - System Modelling
V-MODEL Verication and Validation model
MBSE - Model Based System Engineering
UC - Use Case
BDD - Block Denition Diagram
FL - Firing Line
DJF - Double Joint Factory
QHSE - Quality Health Safety and Environmental
PSV - Pipe Supply Vessels
PS - Pioneering Spirit
ARF - Action Request Form
R - Requirements
C - Requirements Checklist Database
DS - Design Specication
NR - Narrative Requirement
RC - Requirement Checklist
MS – Microsoft VB Visual Basic
DevOps - Development and Operations

ВСТУП

Актуальність теми.

Розробка складних програмних та інформаційних систем потребує чіткого та системного підходу до формування вимог, що визначають функціональність, якість та зручність кінцевого продукту. Помилки на етапі визначення вимог є однією з найпоширеніших причин невідповідності проєктних рішень потребам користувачів та замовників. Інженерія вимог, яка включає методи та підходи до збору, аналізу, формалізації та управління вимогами, є важливим елементом для забезпечення успішної реалізації проєктів. Застосування системного підходу до проєктування на основі інженерії вимог дозволяє підвищити ефективність процесу розробки, знизити ризики та забезпечити високу якість кінцевого продукту, що визначає актуальність цієї теми.

Мета дослідження - розробка та обґрунтування методики системного підходу до проєктування складних систем на основі інженерії вимог, яка забезпечує підвищення якості проєктних рішень та відповідності вимогам замовників.

Об'єкт дослідження - процес проєктування складних програмних та інформаційних систем.

Предмет дослідження – методи, методика та методологія системного підходу до проєктування на основі інженерії вимог.

Відповідно до мети роботи було сформовано наступні **задачі**:

- Провести аналіз предметної області проєктування систем з точки зору інженерії вимог;
- Дослідити існуючі моделі, методи та підходи до проєктування на основі інженерії вимог;
- Розробити пропоновану методику системного підходу до проєктування на основі інженерії вимог;

- Формалізувати вимоги та представити їх у вигляді інженерної моделі;
- Оцінити ефективність запропонованої методики для зниження ризиків та підвищення якості розроблюваних систем.

Методи дослідження.

В роботі використано методи системного аналізу для дослідження предметної області; порівняльний аналіз для оцінки існуючих моделей інженерії вимог; методи моделювання для формалізації вимог; методи концептуального проектування для розробки запропонованої методики; експериментальні методи для оцінки ефективності розробленої методики.

Наукова новизна отриманих результатів.

Запропоновано новий підхід до проектування складних систем на основі інженерії вимог, що включає інтеграцію системного підходу та сучасних методів формалізації вимог і розроблено удосконалену модель формалізації вимог, що забезпечує їхню узгодженість та однозначність.

Практичне значення магістерської роботи полягає в

Розроблена методика може бути застосована у процесі розробки складних програмних та інформаційних систем, що дозволяє підвищити точність визначення вимог, зменшити кількість помилок на етапі проектування та знизити ризики невідповідності кінцевого продукту очікуванням замовників.

Структура магістерської роботи. Робота складається зі вступу, трьох розділів та висновків. Загальний обсяг роботи становить 79 сторінок, і містить 28 рисунків, 4 таблиці, список використаних джерел із 51 найменування.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ПРОЕКТУВАННЯ СИСТЕМ НА ОСНОВІ ІНЖЕНЕРІЇ ВИМОГ

1.1. Аналіз предметної області дослідження

Основною метою цього проекту є розробка систематичної методології розробки вимог до проектування для покращення поточного процесу розробки. Методологія зосереджена на виявленні вимог, зменшенні помилок під час проектування, покращенні поточної фази розробки концепції та сприянні процесу прийняття рішень.

Поточні способи виявлення вимог, які застосовуються це неформальні співбесіди та неструктуровані зустрічі. Тому основний акцент у цьому проекті було зосереджено на структуруванні діяльності з виявлення та аналізу вимог. Підхід, який успішно використовується в методології системної інженерії, такий як «анкета - запитання з високим коефіцієнтом посилення» або «чек-лист». Це допомагає інженерам стимулювати правильне формулювання цілей проекту та структуроване мислення. У довгостроковій перспективі додавання нових запитань до анкети з часом допомогло б інженерам удосконалити методологію. Інформація щодо доступних ресурсів збирається для покращення економіки проекту. Під час аналізу вимог виявляються неправильні та непослідовні вимоги, щоб зменшити помилки під час проектування машини. Методологія розроблена з використанням модельного підходу для зменшення складності конструкції. Крім того, надано детальні та структуровані рекомендації для вдосконалення етапу розробки концепції. У методиці закладено основи системного підходу. Крім того, створюється база даних вимог, яка діє як координаційний центр для майбутніх етапів проекту та центральна мета для обліку вимог. Інструкції щодо діаграм вимог і класифікації вимог надані для ілюстрації зв'язків між вимогами та класифікації вимог. Методологія адаптована до вимог щодо робочих процедур.

Неефективна практика вимог є загальнопромисловою проблемою. Дослідження показують, що основна причина 56% усіх помилок [1], виявлених у продуктах та інженерних проектах, вводиться на етапі вимог. Розробка програмних засобів визначається правильними вимогами до продукту. Через збільшення складності продукту виникає незрозуміння. Крім того, погана комунікація вимог між усіма інженерними дисциплінами та відповідними зацікавленими сторонами забирає багато часу. На рисунку 1.1 зображено проблеми в інженерних проектах, які призводять до невдачі проекту.

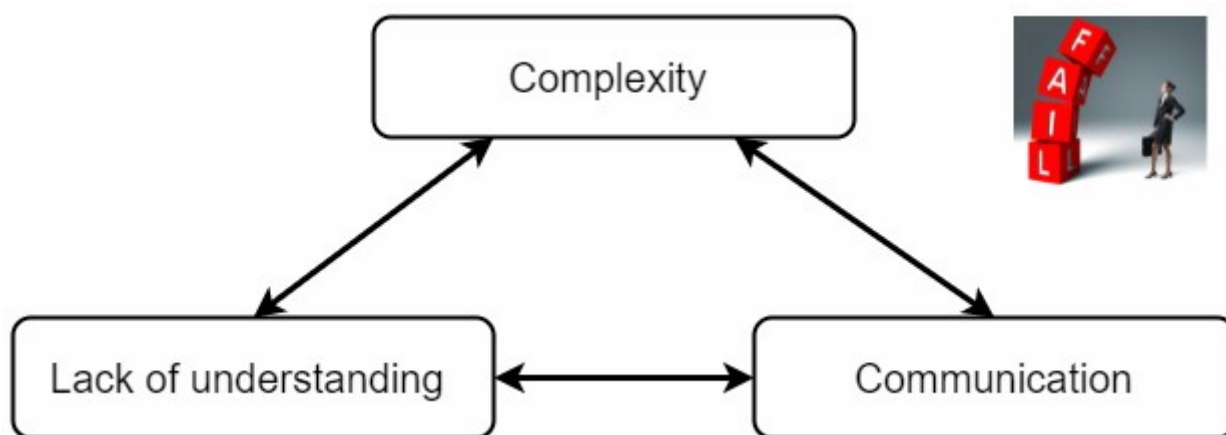


Рис. 1.1. Проблеми в інженерних проектах

Причиною цих труднощів є відсутність спеціальних системних інженерів для визначення, аналізу та відстеження вимог. Наразі, відповідно до підходу до проектування, який використовується, не вистачає чіткої основи чи методології, яка могла б виконувати належне виявлення та аналіз вимог на початковому етапі проектування проекту. Це призводить до затримок і додаткових витрат, оскільки помилки на етапі проектування та концепції виявляються лише на етапі тестування. Таким чином, щоб подолати всі вищезазначені проблеми та розробити робоче рішення, яке задовольняє очікування зацікавлених сторін, потрібен більш дисциплінований підхід до розробки вимог, щоб підвищити показники успіху проекту.

Метою дослідження є: розробити систематичну методологію розробки вимог, яка могла б покращити поточний процес розробки.

Питання дослідження:

- Як методологія вимог сприяє виявленню вимог ?
- Як методологія вимог допомагає знаходити та зменшувати помилки у вимогах під час розробки ?
- Як покращити процес розробки концепції за допомогою методології вимог ?
- Який вплив методології вимог на прийняття рішень у процесі розробки концепції ?

1.2. Опис особливостей фаз проекту

Фази інженерного проекту – це послідовні етапи, протягом яких деталізується концепція проекту від загальної ідеї до повністю розробленого документа, готового для реалізації. Кожна фаза має свої специфічні завдання та результати, які є основою для наступних етапів.

Основні фази інженерного проекту:

1. Передпроектна фаза:

- Формування концепції: Визначення мети проекту, аналіз вимог замовника, дослідження ринку та конкурентного середовища.
- Створення технічного завдання: Детальний опис функціональних можливостей майбутнього продукту або системи.
- Розробка концептуальних рішень: Створення початкових ескізів, схем, моделей, які відображають загальну ідею проекту.

2. Технічне проектування:

- Розробка технічної документації: Створення детальних креслень, схем, розрахунків, які описують конструкцію, матеріали, технологічні процеси.

- Вибір обладнання та матеріалів: Аналіз різних варіантів і вибір оптимальних за технічними та економічними показниками.

- Розробка технологічних регламентів: Визначення послідовності виконання робіт, параметрів процесів.

3. Робоче проектування:

- Розробка робочої документації: Створення детальних креслень, специфікацій, кошторисів, необхідних для будівництва або виробництва.

- Узгодження проекту: Отримання дозволів та узгоджень від відповідних органів.

- Підготовка до реалізації: Складання плану виконання робіт, організація постачання матеріалів та обладнання.

Значення кожної фази:

- Передпроектна фаза: Забезпечує чітке розуміння мети проекту та вимог замовника, дозволяє уникнути помилок на подальших етапах.

- Технічне проектування: Визначає технічні характеристики продукту або системи, забезпечує оптимальне використання ресурсів.

- Робоче проектування: Забезпечує готовність проекту до реалізації, мінімізує ризики відхилень від плану.

Переваги фазового підходу:

- Системність: Кожна фаза логічно пов'язана з попередньою та наступною.

- Контрольованість: Можливість оцінки результатів на кожному етапі.

- Гнучкість: Здатність вносити зміни на ранніх стадіях проекту.

- Ефективність: Оптимізація використання ресурсів.

Розглянемо модель ітераційного водоспаду, де фази проекту будуються в лінійний послідовний процес із зворотним зв'язком від попередньої фази, який представлено на рисунку 1.2. Кожен етап залежить від результатів попереднього кроку. Перший етап – це підготовка проекту, на якій виконується аналіз вимог, техніко-економічне обґрунтування, бюджет і розрахунок часу. Потім проект переходить у фазу розробки концепції, де

аналізуються кілька методів і технік реалізації цілей. Наступний етап – базове проектування. Тут концепція далі аналізується. Створюється базова робоча концепція системи, а модель звіряється з вимогами. Важливо усунути будь-які суперечливі вимоги. Наступний етап – детальне проектування. Розробляється основна робоча концепція. Наприкінці цього етапу зацікавлена сторона повинна перевірити всі вимоги, які були визначені раніше. Необхідні матеріали для виготовлення закуповуються на етапі закупівлі відповідно до визначеного кошторису. Спроектвана система виготовляється на етапі виготовлення. На етапі встановлення та введення в експлуатацію готовий виріб встановлюється та вводиться в експлуатацію на місці. Завершальна фаза – закриття проекту [3].

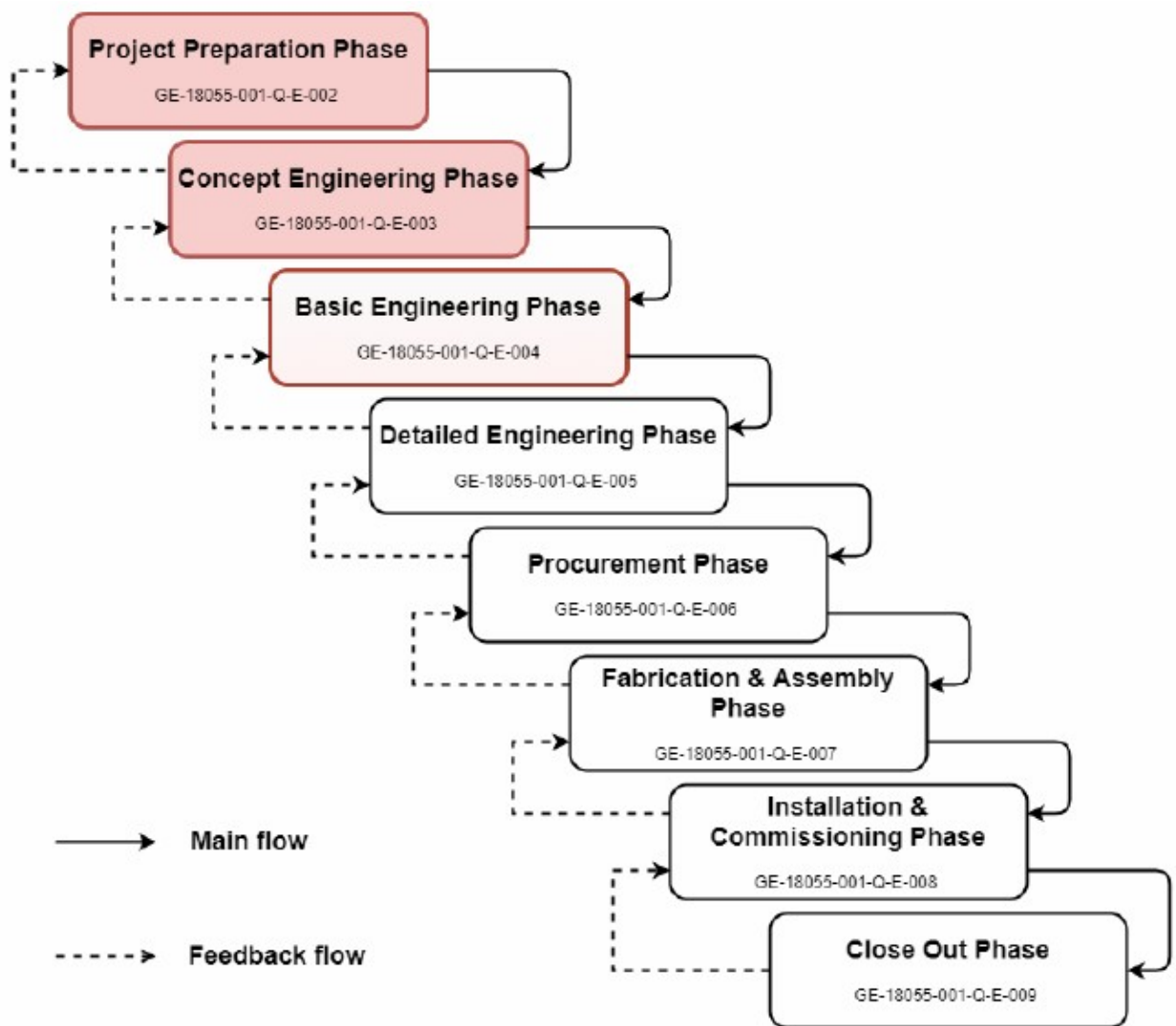


Рис. 1.2. Представлення фаз проекту

Проекти можуть бути різного характеру, але в першу чергу вони стосуються проектування. У проектах типу RD продукти продумані та розроблені для оптимізації певних процесів. У таких проектах команда розробників розробляє рішення для керівника підрозділу відділу інновацій, який виконує роль власника проекту. Обсяг цього проекту вважається фазою підготовки проекту, фазою розробки концепції та частковим внеском у базову фазу розробки, як виділено (червоним) на рисунку 1.2.

1.3. Аналіз проблеми щодо розробки вимог

Розробка вимог відіграє важливу роль в успіху проекту. Однак непорозуміння, відсутність розуміння та втрати часу є результатом недостатнього акценту на аналізі вимог.

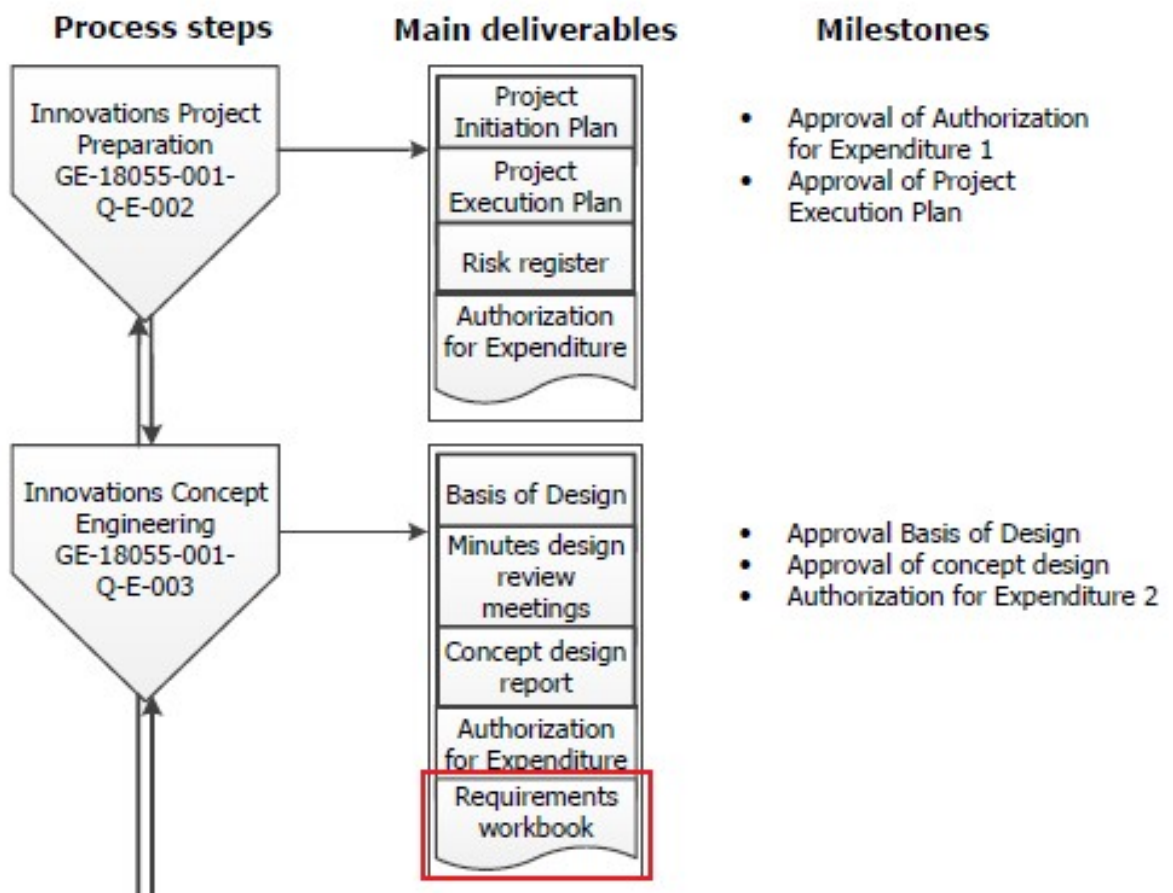


Рис. 1.3. Фрагмент ініціації проекту та етапу розробки концепції

На рисунку 1.4 показано, що аркуш пояснення не зосереджується на основних видах діяльності з розробки вимог, таких як виявлення та аналіз вимог, а зосереджується на класифікації вимог. Через недостатнє пояснення в робочому зошиті вимог більшість учасників проекту, як правило, пропускають робочий зошит або радше використовують звичайні документи MS Word для позначення вимог у відповідний спосіб. Це може призвести до неузгодженого реєстрації вимог і неправильного тлумачення вимог, що, у свою чергу, впливає на процес розробки. Крім того, після аналізу було виявлено, що часто буває багато неповної документації вимог для різних проектів.

У поточному процесі розробки серед дизайнерів більше переважає мислення, засноване на рішеннях, а не на функціональному, що скорочує простір розробки. Крім того, немає стандартних процедур, яких дотримуються дизайнери різних груп у розробці концепції, що робить процес розробки концепції неструктурованим на початкових етапах проекту.

На основі аналізу деяких виконаних проектів і кількох особистих зустрічей з різними учасниками проекту, існує відсутність діяльності з розробки вимог на етапах проекту, що пояснює неповну документацію щодо вимог виконаних проектів і значні відхилення від часу виконання. і бюджет. Ця проблема підтверджує причину вибраних дослідницьких запитань, зосереджується на методології побудови вимог, яка сприяє виявленню вимог, зменшує кількість помилок, покращує процес розробки концепції та допомагає процесу прийняття рішень. Удосконалення процесу проектування зрештою призведе до продукту, який підходить усім зацікавленим сторонам.

1.4. Огляд сутності інженерії вимог

Розробка вимог включає діяльність життєвого циклу, присвячену ідентифікації вимог, аналізу вимог для отримання додаткових вимог, документування вимог як специфікації та перевірку документованих вимог

на потреби зацікавлених сторін, а також процеси, які підтримують цю діяльність. Інженерія вимог вважається предметно-нейтральною дисципліною і використовується для програмного забезпечення, апаратного забезпечення та електромеханічних систем [4].

Інженерія вимог – це систематичний підхід до визначення, документування та управління потребами користувачів щодо розробки програмного забезпечення, системи або продукту. Це перший і один з найважливіших етапів будь-якого інженерного проекту, адже саме від чітко сформульованих вимог залежить успіх подальшої розробки.

Основні етапи інженерії вимог:

1. Збір вимог:

- Інтерв'ю з користувачами та замовником.
- Аналіз існуючих систем та документів.
- Спостереження за роботою користувачів.
- Проведення опитувань.

2. Аналіз вимог:

- Класифікація вимог за типами (функціональні, нефункціональні).
- Перевірка вимог на повноту, узгодженість та несуперечливість.
- Моделювання вимог (наприклад, за допомогою діаграм Use

Case).

3. Специфікація вимог:

- Формалізація вимог у вигляді чітких, вимірних та перевіряємих тверджень.

- Створення документа вимог.

4. Валідація вимог:

- Перевірка того, що вимоги правильно розуміють усі учасники проекту.

- Оцінка вимог на відповідність потребам користувачів.

5. Управління вимогами:

- Відстеження змін вимог протягом усього життєвого циклу проекту.

- Контроль за виконанням вимог.

Інженерія вимог – це фундамент, на якому будується успішний проект. Інвестуючи час і зусилля в цей етап, ви можете значно знизити ризики невдачі та отримати продукт, який повністю відповідає потребам користувачів.

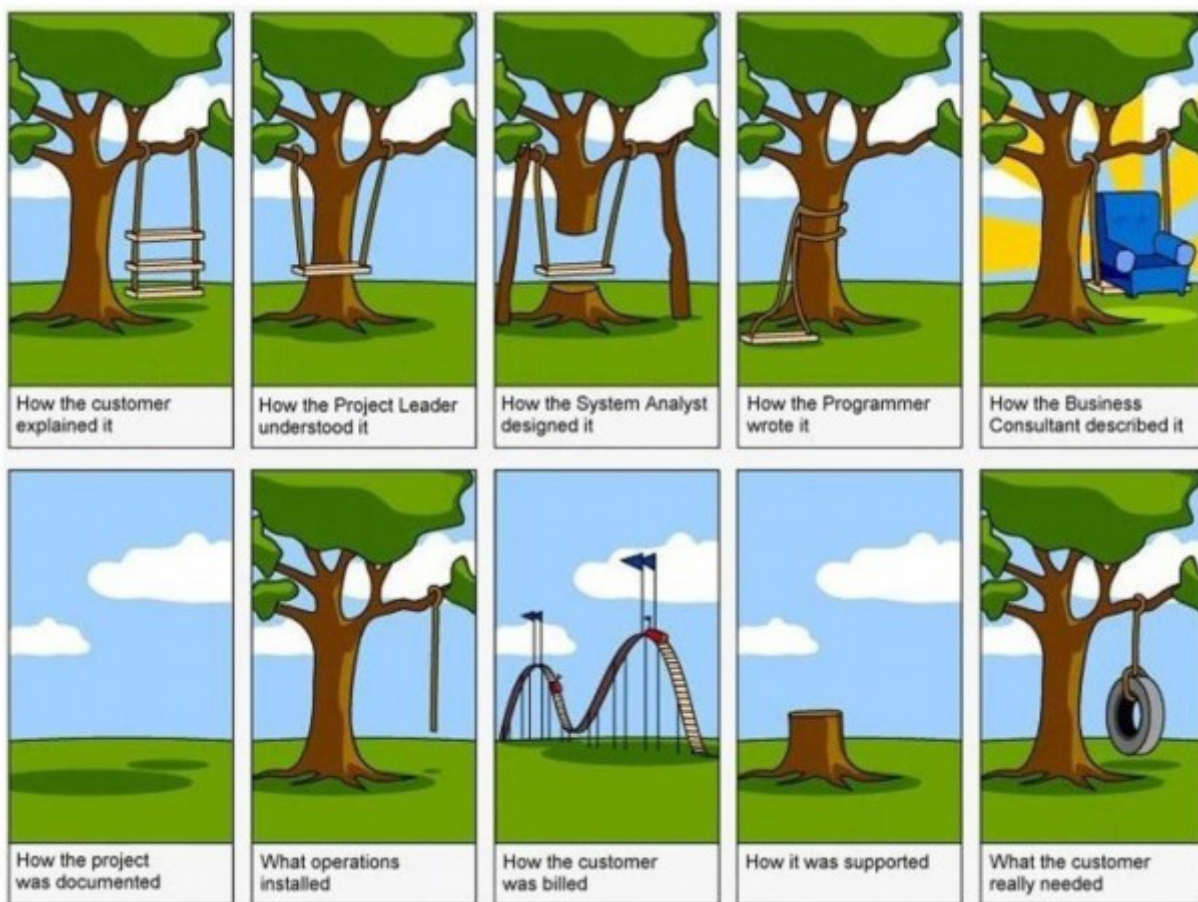


Рис. 1.5. Приклад помилкового розуміння інженерії вимог [5]

Вимоги - це специфікація того, що має бути реалізовано. Це опис того, як повинна поводитися система, або властивостей системи чи атрибутів. Це може бути обмеженням у процесі розробки системи. Це визначення визнає різні типи інформації, які разом називаються «вимогами». Вимоги охоплюють як погляд розробника на деякі внутрішні характеристики

системи, так і погляд зацікавленої сторони на зовнішню поведінку системи. Він включає в себе як поведінку системи в конкретних умовах, так і ті властивості, які роблять систему функціональною для використання за призначенням [6].

На рисунку 1.5 наведено огляд різних поглядів і процесів, задіяних протягом життєвого циклу проекту. Він описує важливість процесу розробки вимог в організації та те, як неправильне тлумачення вимог впливає на кінцевий результат проекту. Він складається з різних зацікавлених сторін, таких як клієнт, керівник проекту, системний аналітик, програміст, бізнес-консультант, і їхні погляди на проект або майбутній продукт. Кожна зацікавлена сторона в організації має свій погляд на продукт. Крім того, різні процеси розробки продукту, такі як проектна документація, операції з продуктом, підтримка продукту тощо, не виконуються належним чином. На рисунку 1.5 показано, що неправильне спілкування між різними зацікавленими сторонами та недотримання процесу розробки вимог призводить до неправильного дизайну продукту та розчарування всіх зацікавлених сторін. Це завдання має на меті визначити чіткий і послідовний процес вимог, щоб уникнути помилкових уявлень на ранніх стадіях розробки продукту.

Інженерія вимог – це не просто перелік функцій, які повинен виконувати продукт. Це комплексний процес, який включає в себе:

- Глибоке розуміння предметної області: Інженери вимог повинні добре розуміти, як працює бізнес замовника, які його цілі та завдання.
- Навички комунікації: Ефективна взаємодія з користувачами та замовником є ключовою для успішного збору вимог.
- Аналітичні здібності: Здатність структурувати інформацію, виявляти протиріччя та приймати обґрунтовані рішення.
- Знання інструментів: Для документування та управління вимогами використовуються спеціальні інструменти та методики.

1.4.1. Проблеми в розробці вимог

Розробка вимог є основним процесом розробки будь-якого продукту. Низькі вимоги призводять не лише до модифікації специфікацій вимог, але також потребують перепроєктування, повторного впровадження та повторного тестування всієї системи. Тому розробникам вимог доводиться долати незліченну кількість викликів, виявляючи ефективні та належні вимоги. Методологія розробки вимог спрямована на подолання цих проблем. Перераховано деякі з типових проблем у розробці вимог:

- Неповні/приховані вимоги – набір вимог, зосереджених на функціональних вимогах без належного врахування бізнес-вимог і нефункціональних вимог.
- Неузгоджені вимоги - набір вимог, який суперечить іншим, що ускладнює надання рішень.
- Термінологія. Набір вимог написаний різною термінологією, що призводить до різного розуміння вимог.
- Нечіткі обов'язки – під час циклу розробки пріоритети зацікавлених сторін не є чіткими. Це створює плутанину (з ким говорити) під час уточнення певних вимог.
- Комунікація – це один із ключових факторів у всіх організаціях, який передбачає короткі електронні листи, різні відділи, різні культури, різні мови тощо.
- Зміни – це стосується змін вимог з часом. Це вплине на вартість і час розробки.
- Технічно нездійсненні вимоги - набір вимог, які технічно неможливо реалізувати під час розробки продукту, які повинні бути виявлені на концептуальній фазі.
- Зацікавлені сторони. Команда проекту має надати відповідні дані зацікавленим сторонам, щоб отримати чіткі та послідовні вимоги, оскільки зацікавлені сторони не є технічними експертами.

- Занижені вимоги. Система, що розробляється, може виконувати більше завдань, але вказівка лише деяких може призвести до невизначеності.

1.5. Інструменти для управління інженерією вимог

Інструменти для управління інженерією вимог – це програмні засоби, які допомагають ефективно збирати, аналізувати, документувати та відстежувати зміни вимог до програмного продукту або системи. Вони є незамінними помічниками для бізнес-аналітиків, проєктувальників та розробників, оскільки дозволяють:

- Структурувати інформацію: Зберігати всі вимоги в єдиній базі даних, класифікувати їх за різними критеріями, встановлювати взаємозв'язки.
- Підвищити прозорість: Забезпечити доступ до вимог для всіх зацікавлених сторін та спростити комунікацію.
- Поліпшити контроль: Відстежувати виконання вимог, управляти змінами та аналізувати вплив змін на інші частини проєкту.
- Забезпечити трасування вимог: Встановити зв'язки між вимогами, тестовими випадками та іншими артефактами проєкту.

Основні функціональні можливості інструментів:

- Збір вимог: Інтерфейси для введення вимог, шаблони, підтримка різних форматів (текстові, табличні, графічні).
- Аналіз вимог: Можливості для перевірки вимог на повноту, узгодженість та несуперечливість, автоматична генерація звітів.
- Документування вимог: Створення структурованих документів, генерація різноманітних звітів.
- Управління змінами: Відстеження змін вимог, аналіз їх впливу, контроль версій.
- Трасування вимог: Встановлення зв'язків між вимогами, тестовими випадками, дефектами та іншими артефактами.

- Інтеграція з іншими інструментами: Інтеграція з системами керування проектами, системами контролю версій, інструментами тестування.

Розглянемо популярні інструменти для управління вимогами.

Visure Requirements: Потужний інструмент з широким набором функцій, підтримує різні стандарти (DO-178B, ISO 26262).

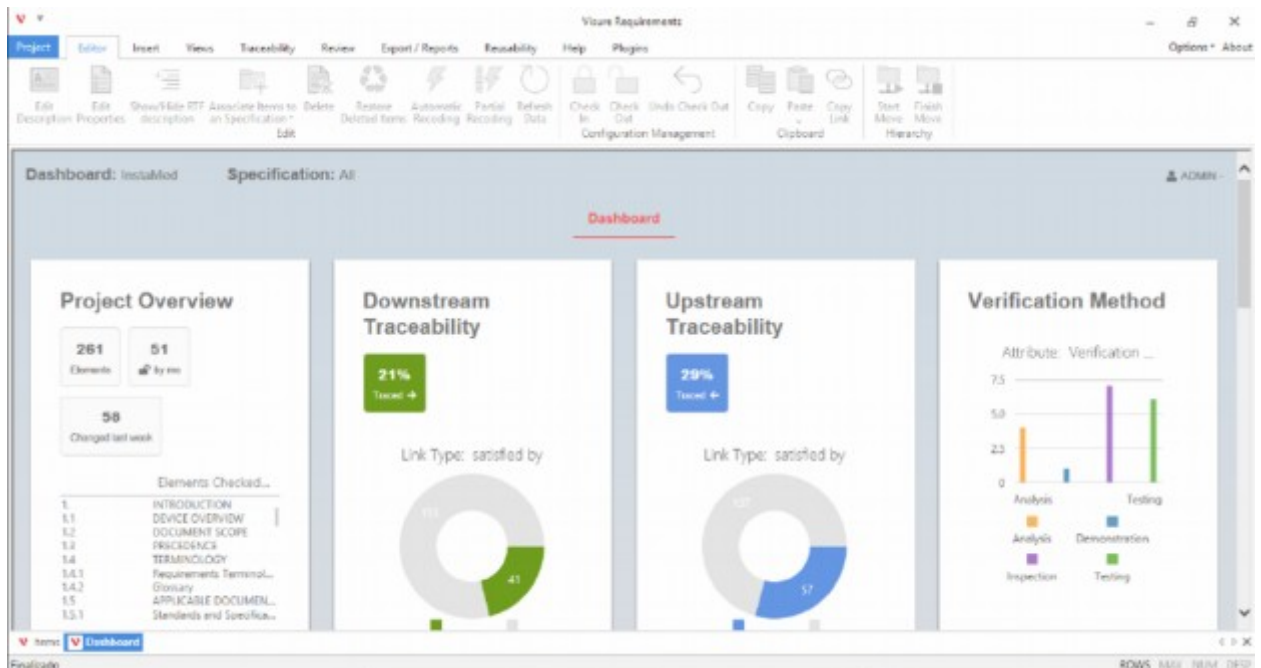


Рис. 1.6. Інструмент Visure Requirements

IBM DOORS Next: Класичний інструмент для управління вимогами, інтегрований з іншими продуктами IBM. IBM DOORS Next є потужним і гнучким інструментом для управління вимогами, який допомагає оптимізувати процес збору, аналізу, документування та відстеження вимог протягом усього життєвого циклу розробки ПЗ. Він пропонує широкий набір функцій, що дозволяють організаціям гарантувати відповідність їхніх продуктів потребам клієнтів та дотримуватися галузевих стандартів.

Цільова аудиторія:

- Системні інженери: Відповідальні за визначення та управління системними вимогами.

- Програмні інженери: Залучені до проектування, розробки та тестування програмних продуктів.
- Керівники проектів: Оглядають загальний життєвий цикл проекту.
- Інженери з забезпечення якості: Гарантують якість продукту та відповідність.

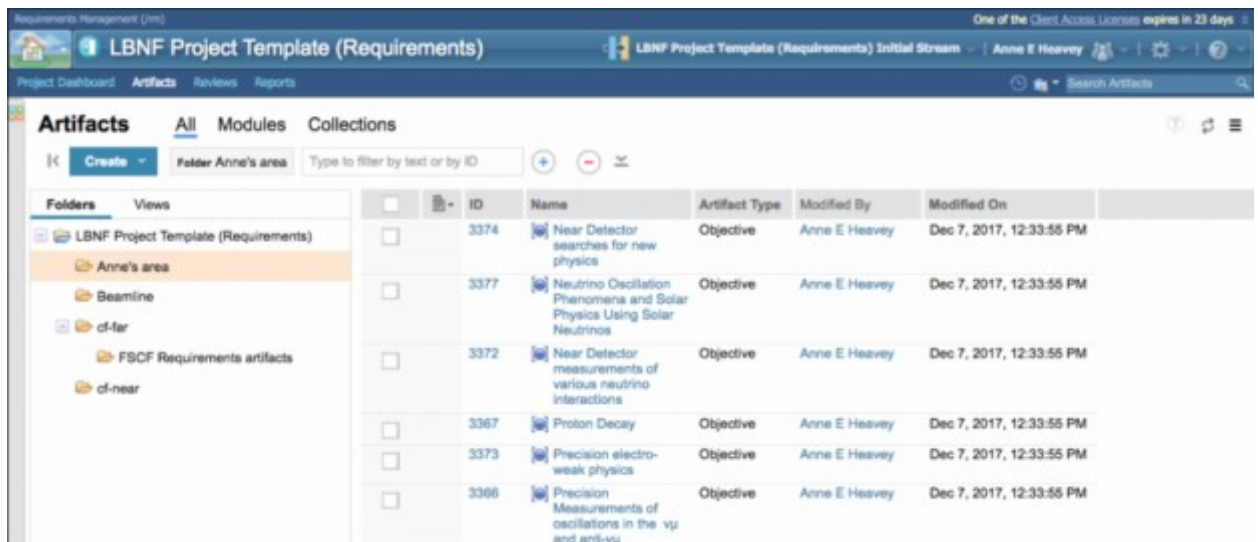


Рис. 1.7. Інструмент управління вимогами VM DOORS Next

Jama Software: Хмарний інструмент з інтуїтивним інтерфейсом та широкими можливостями для співпраці. Jama Software є потужним і гнучким інструментом для управління вимогами, який допомагає оптимізувати процес збору, аналізу, документування та відстеження вимог протягом усього життєвого циклу розробки програмного забезпечення. Він пропонує широкий набір функцій, що дозволяють організаціям гарантувати відповідність їхніх продуктів потребам клієнтів та дотримуватися галузевих стандартів.

Основні функції:

- Централізований репозиторій: Зберігає всі вимоги в єдиному місці, що полегшує доступ та управління ними.

- Авторство вимог: Надає середовище для створення, редагування та організації вимог, включаючи підтримку різних форматів документів та шаблонів.
- Пов'язання вимог: Дозволяє створювати зв'язки між вимогами, артефактами дизайну, тестовими випадками та іншими елементами проекту, що сприяє трасуваності та аналізу впливу змін.
- Управління вимогами: Пропонує функції для управління змінами вимог, відстеження їхнього статусу та призначення відповідальностей.
- Співпраця: Підтримує співпрацю між членами команди завдяки функціям коментування, версіонування та управління робочим процесом.
- Інтеграція: Інтегрується з іншими інструментами та платформами, забезпечуючи безшовний робочий процес та обмін даними.
- Підтримка відповідності: Допомагає організаціям дотримуватися галузевих стандартів та регулятивних вимог.

Heading	ID	Name	Created By	Modified Date	Release
5	UAV-CMP-1	TCS System Design	Becky Carlton	09/21/2016	
5.1	UAV-SET-6	High Level SW Requirements	Becky Carlton	09/21/2016	
5.1.1	UAV-HLR-2	CL read	Becky Carlton	09/22/2016	Unassigned
5.1.2	UAV-HLR-18	CL input	Becky Carlton	09/22/2016	Unassigned
5.1.3	UAV-HLR-19	CL define	Becky Carlton	09/22/2016	Unassigned
5.1.4	UAV-HLR-20	CL range limit	Becky Carlton	09/22/2016	Unassigned
5.1.5	UAV-HLR-21	CL calculate	Becky Carlton	09/22/2016	Unassigned
5.2	UAV-SET-7	Low Level SW Design	Becky Carlton	09/21/2016	
5.2.1	UAV-LLR-2	Read Function	Becky Carlton	09/22/2016	Unassigned
5.2.2	UAV-LLR-3	Input	Becky Carlton	09/22/2016	Unassigned
5.2.3	UAV-LLR-4	Define	Becky Carlton	09/22/2016	Unassigned
5.2.4	UAV-LLR-5	range limit	Becky Carlton	09/22/2016	Unassigned

Рис. 1.7. Інструмент управління вимогами Jama Software

PTC Integrity Modeler: Частина інтегрованої платформи ALM від PTC, підтримує весь життєвий цикл розробки програмного забезпечення.

PTC Integrity Modeler є потужним інструментом, який забезпечує інтегроване управління життєвим циклом розробки (ALM) для організацій, що працюють у галузях аерокосмічної промисловості, автомобільного виробництва, медичної техніки та інших регульованих областях. Він пропонує широкий набір функцій, включаючи управління вимогами, тестування, дефектами та змінами, що дозволяє організаціям ефективно розробляти та доставляти високоякісні продукти.

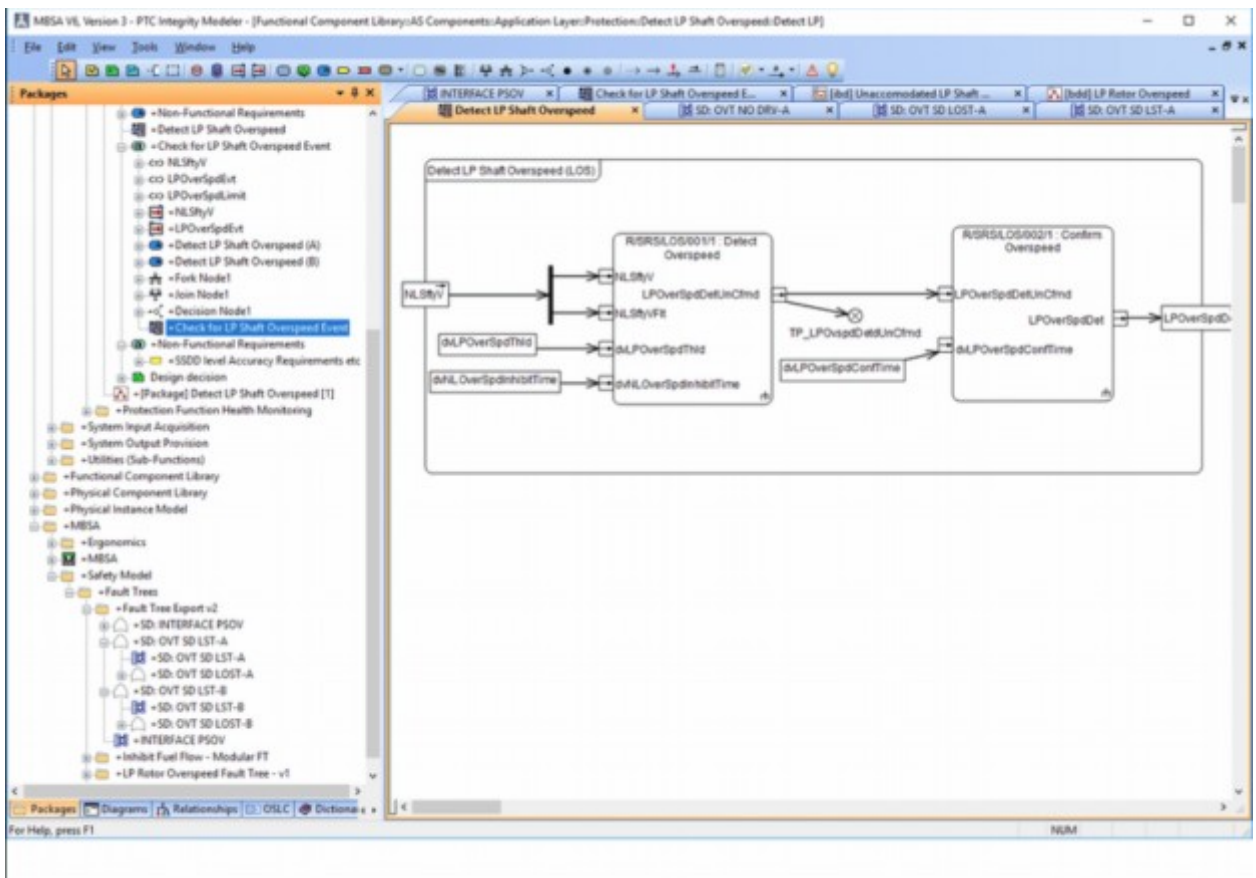


Рис. 1.8. Інтерфейс PTC Integrity Modeler

Polarion ALM: Ще одна комплексна платформа для управління вимогами, що включає в себе функції планування, тестування та управління дефектами.

Polarion ALM – це потужний інструмент, який допомагає компаніям ефективно керувати всіма етапами розробки програмного забезпечення, від початкової ідеї до випуску готового продукту. Він об'єднує в собі безліч функцій, дозволяючи командам розробників працювати більш злагоджено та продуктивно.

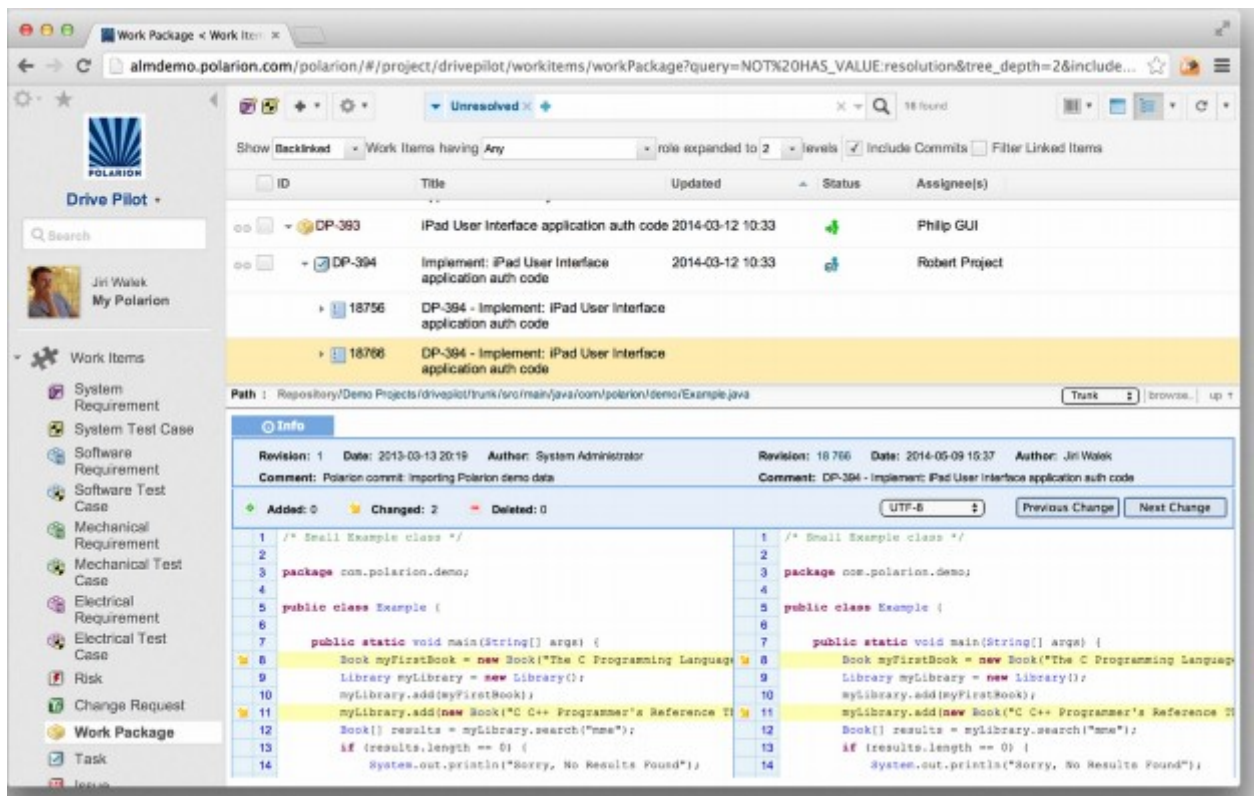


Рис. 1.9. Інтерфейс Polarion ALM

Основні можливості:

- Управління вимогами:
 - Створення, редагування та організація вимог до продукту.
 - Відстеження зв'язку між вимогами та іншими елементами проекту (наприклад, тестовими випадками, завданнями).
 - Аналіз впливу змін у вимогах на весь проект.
- Управління розробкою:
 - Планування та відстеження завдань розробки.
 - Інтеграція з системами контролю версій (Git, SVN та ін.).

- Автоматизація збірки та розгортання програмного забезпечення.
- Управління тестуванням:
 - Створення та управління тестовими сценаріями.
 - Автоматизація тестування.
 - Аналіз результатів тестування.
- Управління дефектами:
 - Відстеження помилок та дефектів.
 - Призначення завдань з виправлення дефектів.
- Аналітика та звітність:
 - Отримання актуальної інформації про стан проекту.
 - Створення кастомізованих звітів.

Висновки до розділу

В ході дослідження предметної області проектування систем на основі інженерії вимог були розглянуті ключові аспекти цього процесу, включаючи аналіз предметної області, опис фаз проекту, специфіку розробки вимог та інструменти для управління цим процесом.

Аналіз предметної області показав, що успішне проектування програмних систем вимагає глибокого розуміння не тільки технічних аспектів, а й бізнес-вимог, які впливають на архітектуру системи та її функціональність. Це підкреслює необхідність ретельної роботи з вимогами на початкових етапах проекту, щоб уникнути значних ризиків та витрат при зміні проекту на пізніших етапах.

Огляд сутності інженерії вимог і вивчення її проблематики підтвердили важливість систематизованого підходу до управління вимогами. Інженерія вимог забезпечує структурування, аналіз та документування потреб замовників, що сприяє більш ефективному управлінню змінами та гарантує якісне проектування системи.

Розгляд інструментів для управління інженерією вимог (наприклад, Polarion ALM, DOORS, та інші) показав, що вони значно полегшують процеси збору, відстеження та аналізу вимог. Ці інструменти забезпечують прозорість та трасованість змін, що допомагає уникати помилок і сприяє більш злагодженій роботі команд розробників.

Загалом, результати дослідження свідчать про те, що використання системного підходу до проєктування з акцентом на інженерію вимог є критично важливим для успішного виконання проєктів з розробки програмного забезпечення. Це дозволяє досягти високої якості розроблених систем та задовольнити потреби замовників.

РОЗДІЛ 2. МОДЕЛІ, МЕТОДИ ТА ПІДХОДИ ДО ПРОЕКТУВАННЯ НА ОСНОВІ ІНЖЕНЕРІЇ ВИМОГ

2.1. Порівняльний аналіз моделей інженерії вимог

Інженерія вимог (RE) — це набір різних процесів, які працюють на різних рівнях, які включені в проекти індивідуального та організаційного рівнів [8]. Інженери повинні мати належне розуміння процесу розробки вимог, щоб адаптувати діяльність. Детальне дослідження різних моделей процесу розробки вимог представлено з існуючого огляду літератури [9]. Проте кожна модель процесу має свої переваги та недоліки. У цьому огляді з багатьох моделей RE проаналізовано чотири різні типи широко прийнятих моделей процесу розробки вимог. Підводячи підсумок, можна сказати, що детальний огляд моделей процесів розробки вимог разом із їхніми сильними та слабкими сторонами допомагає у технічному виборі відповідної моделі для Allseas.

2.1.1. Лінійна модель Маколей

Це лінійна модель процесу розробки вимог, запропонована Маколем, яка показана на рисунку 2.1. Вона не підтримує збіг дій. Етапи моделі процесу є послідовними та включають концепцію, аналіз проблеми, аналіз і моделювання техніко-економічного обґрунтування та документацію вимог. Як запропонував Маколей, процес RE залежить від ситуації, а також від відносин клієнт-постачальник [10]. Це найпростіша модель RE, яка використовується для невеликих і менш складних проектів. Він не підходить для великих проектів.

Лінійна модель інженерії вимог є одним із найпростіших і найінтуїтивніших підходів до збору, аналізу та специфікації вимог до програмного забезпечення. Вона передбачає послідовне виконання етапів, починаючи від збору вимог від замовника і закінчуючи їх формалізацією.

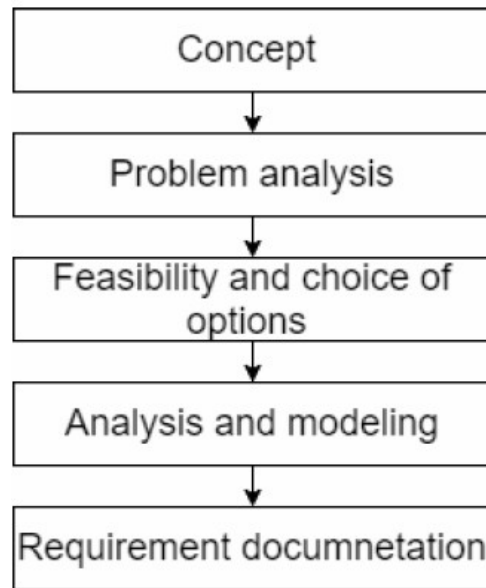


Рис. 2.1. Лінійна модель Маколея

Сильні та слабкі сторони моделі процесу розробки лінійних вимог наведено в таблиці 2.1.

Таблиця 2.1.

Сильні та слабкі сторони лінійної моделі

Strength	Weakness
<ul style="list-style-type: none"> • It provides support to analyze the feasibility of the system. • It provides the facility of validating customer requirements. • Pure linear model and does not involve any overlapping of activities. 	<ul style="list-style-type: none"> • It does not support the reverse engineering process. • It lacks the policies for performing risk management. • It lacks the activities of requirement pre-processes. • It lacks the facility of changing requirements. • It lacks the feedback activity.

2.1.2. Спиральна модель процесу розробки вимог

Спиральна модель запропонована Kotonya і Sommerville і представлена в на рисунку 2.2. Ключова особливість цієї моделі - спіраль. Кожна спіраль має чотири основні розділи: виявлення вимог, аналіз і узгодження вимог, документація вимог і підтвердження вимог. Кожна фаза починається з мети і закінчується підтвердженням прогресу розробником або клієнтом. Перевага

спіральної моделі полягає в тому, що вона дозволяє поетапно випускати та доводити вироби через кожну петлю спіральної моделі. Він підтримує створення прототипів у кожному циклі. Крім того, метою моделі є подолання наслідків, які впливають на якість і вартість проекту, які виникають на різних етапах розробки продукту.

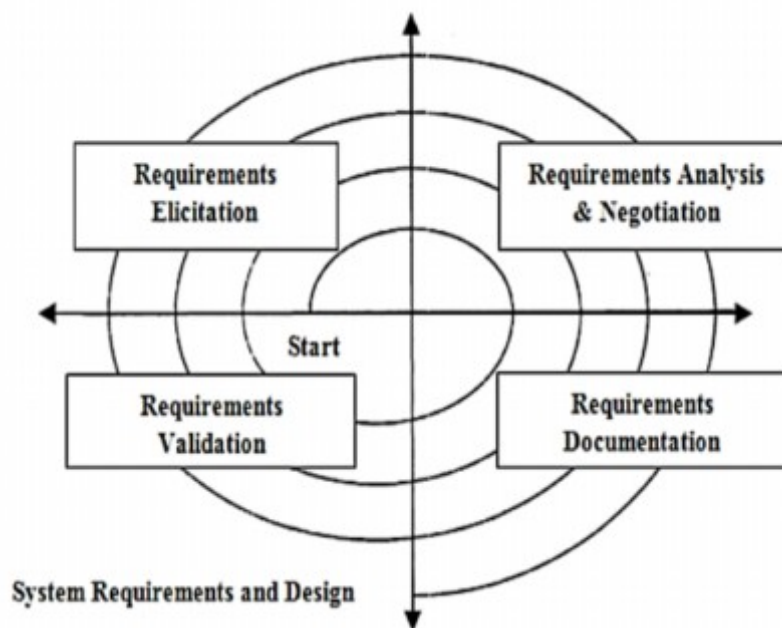


Рис. 2.2. Спіральна модель [11]

Сильні та слабкі сторони спіральної моделі процесу розробки вимог наведено в таблиці 2.2.

Таблиця 2.2.

Сильні та слабкі сторони спіральної моделі [9, 12]

Strength	Weakness
<ul style="list-style-type: none"> ● It supports the policies for performing risk management. ● It supports changing of requirements at later phases. ● It is well suited for large and complex projects. ● It supports prototyping and more suitable for agile work procedure. ● It provides the means of client's feedback. 	<ul style="list-style-type: none"> ● It is not suitable for small projects as it is expensive. ● The process is more complex than other models. ● Number of intermediate stages in the process require excessive documentation. ● It lacks the process of requirement prioritization.

2.1.3. Загальна модель Pandey & Suman

Загальна модель процесу, запропонована Дхірендрою Панді та У. Суманом, пов'язує процес розробки вимог із процесом розробки, який зображено на рис. 2.3. Вартий уваги аспект цієї моделі полягає в тому, що вона представляє важливі та невидимі точки зору - процесу розробки вимог, такі як обмеження, вимоги безпеки, бізнес-вимоги, вимоги клієнтів, вимоги до інформації, вимоги користувачів, стандарти тощо для виробництва якісної продукції. Крім того, модель надає більш широкі можливості для документування вимог і включає валідацію та верифікацію вимог. Крім того, вона охоплює етап управління вимогами та планування для підтримки зміни вимог [13].

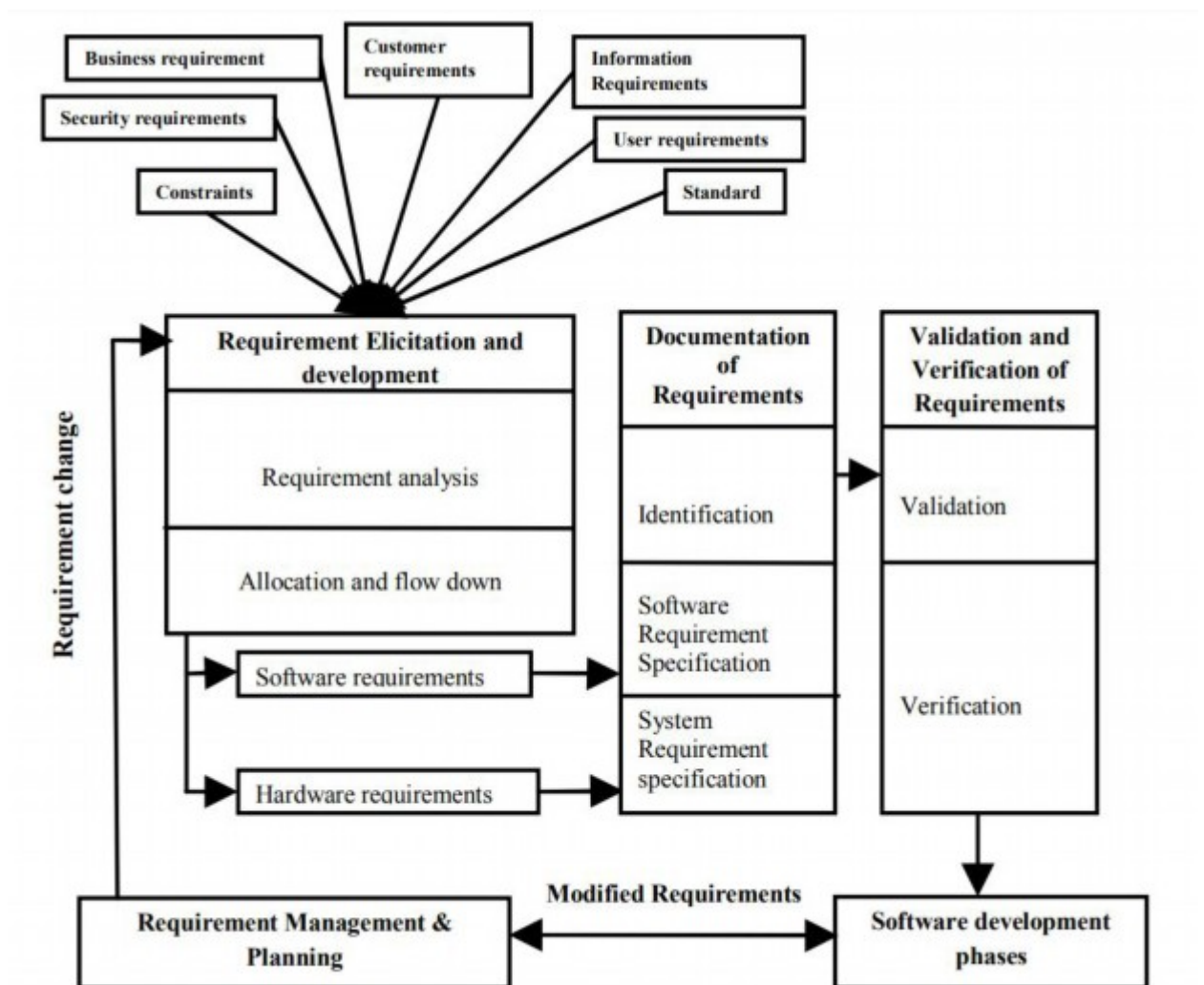


Рис. 2.3. Модель ефективного процесу інженерії вимог Pandey & Suman

Сильні та слабкі сторони моделі процесу розробки загальних вимог наведено в таблиці 2.3.

Таблиця 2.3. Сильні та слабкі сторони загальної моделі

Strength	Weakness
<ul style="list-style-type: none"> • It supports requirement management and planning for product development. • It provides the facility of changing requirements. • It supports different viewpoints during requirement elicitation. • It provides the means of client's feedback. 	<ul style="list-style-type: none"> • It lacks the facility of selecting an appropriate selection of elicitation technique. • It lacks the activities of requirements pre-processes. • It gives focus on requirements classification rather than requirements analysis. • It is more inclined towards software development.

2.1.4 Модель вимог Sommerville

Модель процесу розробки вимог, запропонована Яном Соммервілем, показана на рисунку 2.4. Першою фазою проекту є техніко-економічне обґрунтування, яке надає техніко-економічний звіт.

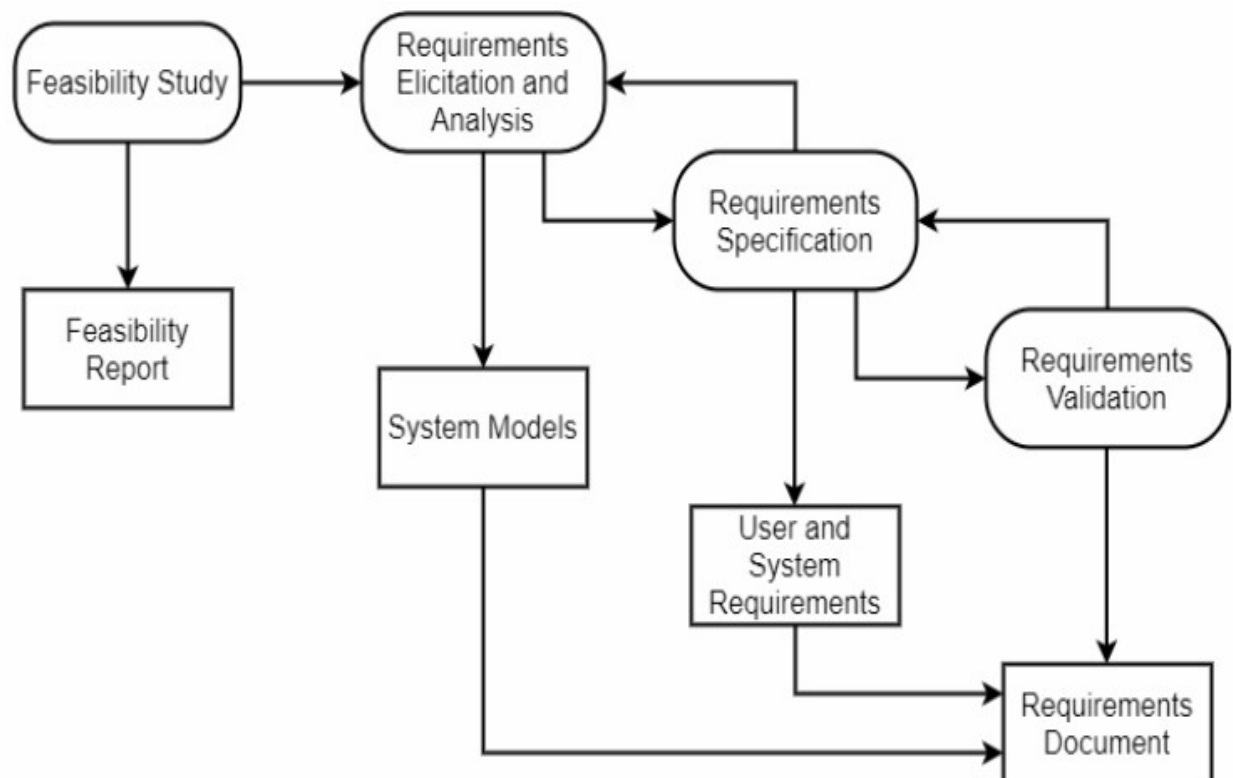


Рис. 2.4. Модель процесу розробки вимог від Sommerville [14]

Дослідження в основному зосереджено на плануванні бюджету з точки зору бізнесу. Потім фази виявлення та аналізу вимог, специфікації вимог і підтвердження вимог є ітераційними. Результатами фази виявлення та аналізу вимог є системні моделі, які є ключовим аспектом цього процесу розробки вимог. Він передбачає розробку однієї або кількох системних моделей для кращого розуміння системи, яка має бути визначена. Специфікація вимоги — це перетворення інформації, зібраної під час аналізу, у документ, який визначає набір вимог. У цьому процесі враховуються вимоги користувача та системи. Перевірка вимог перевіряє вимоги щодо реалістичності, послідовності та повноти. Нарешті, усі результати етапів вимог вносять свій внесок у документ вимог [14].

Сильні та слабкі сторони моделі процесу розробки ітераційних вимог наведено в таблиці 2.4 .

Таблиця 2.4.

Сильні та слабкі сторони моделі вимог Sommerville [9, 14]

Strength	Weakness
<ul style="list-style-type: none"> ● It supports feasibility study which involves budgetary analysis. ● It provides the facility of changing requirements. ● It supports higher-level system modeling for requirement analysis. ● It supports to identify errors in the early stage of process. ● It provides excellent support towards documentation. 	<ul style="list-style-type: none"> ● It lacks the process of detailed requirement prioritization. ● It lacks the policies for performing risk management. ● It needs more project management activities. ● It does not support different viewpoints.

2.2. Представлення інженерної моделі вимог на основі системного підходу

У цьому розділі обговорюється запропонована інженерна модель вимог, основи системного підходу, системного підходу, керованого моделлю, і аналіз методів RE для підтримки запропонованої моделі RE.

2.2.1. Пропонована інженерна модель вимог

Інженерна модель вимог — це систематичний підхід до визначення, документування, аналізу та управління вимогами для створення технічних систем чи програмного забезпечення. Вона використовується для того, щоб забезпечити зрозуміле та точне визначення того, що саме має робити система, щоб задовольнити потреби та очікування користувачів і зацікавлених сторін.

Основні аспекти інженерної моделі вимог включають:

- Збір вимог: Включає визначення та документування очікувань та потреб користувачів. Це може бути досягнуто через інтерв'ю, анкети, воркшопи або аналіз існуючих систем.

- Аналіз вимог: Вимоги аналізуються для того, щоб уточнити їх, перевірити їхню узгодженість та повноту, а також ідентифікувати можливі конфлікти між вимогами.

- Специфікація вимог: Формальне документування вимог у вигляді текстових документів, діаграм, моделей тощо. Специфікація повинна бути зрозумілою, точною та структурованою.

- Перевірка та валідація вимог: Це процес перевірки, чи відповідають вимоги потребам користувачів, чи правильно вони інтерпретовані, та чи реалістично їх виконати. Це включає рев'ю, тести та інші методи оцінки.

- Управління вимогами: Передбачає контроль змін у вимогах протягом життєвого циклу проекту, що допомагає уникнути проблем, пов'язаних зі змінами в проектах та забезпечує відслідковуваність вимог.

Інженерна модель вимог допомагає забезпечити ясність у спілкуванні між розробниками, дизайнерами та замовниками, мінімізує ризики неправильного розуміння та полегшує процес тестування й впровадження системи. Вона є основою для планування проекту та управління його виконанням.

На основі огляду інженерних моделей вимог в розділі 2.1, вибирається відповідна модель вимог. Модель RE вибирається за певними критеріями.

Широкі списки критеріїв відбору звужуються за рахунок пріоритетності потреб моделі RE. Критерії відбору ґрунтуються на певних вимогах. Основною вимогою є зменшення бюрократії, створення прагматичної методології, підтримка різних відділів і підтримка повторного визначення вимог. Зібрані вимоги перефразовано та додано як критерій відбору для оцінювання. Опис критеріїв:

- Придатність для фаз проекту : модель сумісна з ітеративною моделлю водоспаду для фаз проекту.
- Підтримка простого процесу : модель підтримує простий процес, який виводить окремі результати після кожного етапу. Не повинно бути збігів етапів.
- Підтримка мінливих вимог: модель підтримує мінливі вимоги. Якщо існують суперечливі вимоги, приймається зворотний зв'язок від дизайнерів із зацікавленими сторонами.
- Підтримка проектів різного розміру: модель підтримує різні відділи проектів різного розміру.
- Розробка концепції підтримки: модель підтримує та дає рекомендації на етапі розробки концепції.
- Підтримка економіки проекту: модель підтримує економічну цінність проекту.

Таблиця 2.5.

Критерії вибору моделі вимог

Criteria for selection	Linear Model	Spiral Model	Pandey Model	Sommerville Model
Suitability for project phases	✓	×	✓	✓
Support simple process	✓	×	×	×
Support changing requirements	×	✓	✓	✓
Support varied-size projects	×	×	✓	✓
Support concept development	✓	✓	×	✓
Support project economics	✓	×	×	×

Проведено детальний аналіз попередніх моделей процесів розробки вимог і зведено в таблицю 2.5. Порівняння проводиться з використанням вищезазначених критеріїв відбору. V означає, що модель відповідає критеріям, тоді як «X» означає, що модель не відповідає критеріям.

Детальна аргументація базується на аналізі, представленому в таблиці 2.5. Аргументація поділяється на такі підрозділи: оцінка моделей управління вимогами (RE), вибір основних видів діяльності RE, та запропонована модель управління вимогами, інтегрована у фазах проєкту.

По-перше, оцінка різних моделей RE ґрунтується виключно на критерії відбору. Згідно з таблицею 2.5, модель Маколі та Соммервіля має значну кількість оцінок "3". Лінійна модель Маколі є простим процесом, що підтримує етапи проєкту, розробку концепції та економічний аналіз проєкту, але підходить лише для невеликих проєктів і не підтримує змінні вимоги, що є критичними причинами для виключення цієї моделі. Далі, спіральна модель є однією з ключових моделей в організації розробки програмного забезпечення. Цю модель не було розглянуто через її складність та вимогу до великої кількості документації. Вона також потребує значних знань в системній інженерії та гнучкого підходу до роботи, що наразі не є досяжним. Наступною розглядалася модель Пандей, яка зосереджується переважно на невидимих аспектах вимог та фазі управління вимогами і планування. Однак, цю модель було відхилено через її недостатню підтримку розробки концепції. Нарешті, модель Соммервіля показала найбільшу відповідність обраним критеріям. Вона добре підходить для різних етапів проєкту, проєктів різних розмірів і підтримує змінні вимоги. Основною причиною вибору цієї моделі є її підтримка розробки концепції через системні моделі. Ці моделі сприяють кращій комунікації між зацікавленими сторонами та стимулюють функціональне мислення. Недоліком цієї моделі є її складність та відсутність підтримки економічного аналізу проєкту, тому модель потребує деяких адаптацій для мінімізації цих недоліків. У підсумку, компанія Allseas потребує вдосконалення процесів визначення та аналізу вимог, а також

розробки концепції, що підкреслюється в моделі Соммервіля. Більшість критеріїв відбору підтримуються, що посилює аргументацію та обґрунтовує вибір моделі Соммервіля.

По-друге, основні види діяльності в моделі Соммервіля включають: дослідження здійсненності, визначення та аналіз вимог, специфікацію та валідацію вимог. Ці види діяльності дають конкретні результати, такі як звіти про здійсненність, системні моделі, вимоги користувачів і системи, а також документацію щодо вимог. Процес було спрощено для забезпечення більшої абстракції та спрощення. Оскільки визначення та аналіз вимог потребують більшої уваги, їх було розділено на два основних види діяльності. Дослідження здійсненності було включено як піддіяльність аналізу вимог, щоб спростити підготовку звіту про здійсненність. Клієнтами компанії є керівники підрозділів, залежно від типу проєкту, що пояснює, чому етап визначення вимог користувачів не було включено до моделі. Основний акцент зроблено на визначенні та аналізі вимог, а не на їх класифікації. Для зменшення недоліків моделі етап вдосконалення економіки проєкту включено до піддіяльностей управління вимогами. Системні моделі, запропоновані в моделі Соммервіля, трансформовано в основну діяльність під назвою «Системне моделювання» у запропонованій моделі. Системні моделі розглядаються не лише як результати, але й вимагають дій та настанов для їх розробки. Валідація вимог є ключовою частиною методології, оскільки кожна вимога потребує перевірки на пізнішому етапі проєкту. Основні види діяльності в запропонованій моделі RE включають:

- Визначення вимог;
- Аналіз вимог;
- Системне моделювання;
- Валідація вимог.

Нарешті, запропоновані основні види діяльності щодо RE відповідно до фаз проєкту. На рисунку 2.5 зображено діяльність із RE, включену до робочої процедури. Як обговорювалося раніше, про фази проєкту, які

будуються в лінійний послідовний процес, які представлені в пунктирній сірій рамці на рисунку 2.5.

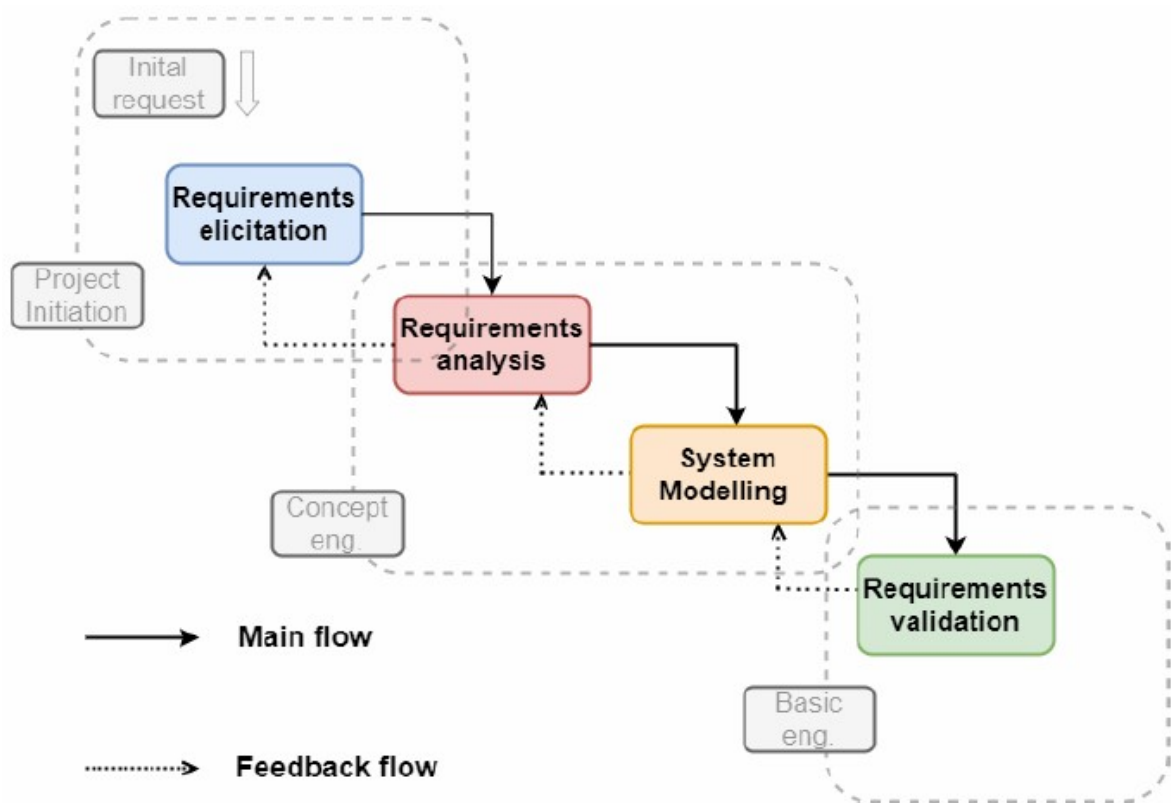


Рис. 2.5. Пропонований процес розробки вимог

Початковий запит від власника продукту обробляється в активності збору вимог і так далі. Важливо, що активності збору вимог (RE) в рамках проектних етапів також дотримуються послідовного процесу зі зворотним зв'язком, що зміцнює аргументи на користь вибору цієї моделі. Кодове позначення активностей RE зберігається подано в наступному розділі, що дозволяє зрозуміти активності RE на етапах методології. Крім того, моделювання системи на вищому рівні сумісне з концептуальним етапом інженерії. Вимоги документуються під час кожної активності, що частково сприяє основному етапу інженерії. Валідація вимог проводиться пізніше на етапах проекту, що не входить у сферу даного дослідження.

2.2.2 Основи системного підходу

Основи системного підходу можна класифікувати за двома категоріями, а саме когнітивно-психологічні відносини та загальні методичні принципи. Це допомагає структурувати запропоновані процедури та окремі методи, які застосовуються до проектування. Загальна робоча методологія має бути широко застосовною, незалежною від дисципліни та адаптивною. Більше того, для використання методології та підтримки структурованого та ефективного процесу мислення мають бути потрібні мінімальні технічні знання.

Існує багато загальних методичних принципів системного підходу. Деякі з важливих принципів, що мають відношення до цієї роботи, - це аналіз, абстракція, синтез, метод постійних запитань і метод систематичної варіації [16], які пояснюються:

- Аналіз — це розділення будь-чого складного на компоненти та вивчення компонентів та їхніх взаємозв'язків.
- Абстрагування полягає у пошуку взаємозв'язку вищого рівня, більш загального та всебічного. Це зменшує складність.
- Синтез — це об'єднання частин або елементів для отримання нових ефектів і демонстрації ефектів для створення загального порядку.
- Метод наполегливих запитань викликає свіжі думки та інтуїцію.
- Метод систематичної варіації передбачає поділ загальної проблеми на підпроблеми; знаходження індивідуальних рішень; об'єднання рішень у загальне рішення.

За допомогою системного підходу повноцінно заповнюються такі аспекти [16]:

- Визначення цілей - Це досягається формулюванням загальної мети та окремих підцілей. Це забезпечує розуміння проблеми і вирішення завдання.
- Уточнення граничних умов. Це виконується шляхом визначення обмежень, які допомагають зосередитися на визначених цілях.

- Усунення упередженого ставлення – це уникнення припущень чи упереджень у технічному середовищі чи проекті та уникнення логічних помилок.
- Пошук варіантів - це забезпечення широкого діапазону можливих рішень, з яких можна вибрати найкраще.
- Прийняття рішень – це досягається шляхом структурованої оцінки варіантів.

2.2.3 *Модельно керований системний підхід*

Запропонована модель RE дає абстрактний погляд на процес вищого рівня. Після вибору запропонованої моделі RE вибирається належна методологія, яка підтримує діяльність з RE. Вимоги методології мають покращити розуміння завдань, комунікацію зацікавлених сторін та розробку концептуальних моделей. Крім того, методологія повинна підтримувати модифікацію відповідно до конкретних цілей. Виходячи з вищезазначених потреб, вибрано методологію SYSMOD.

SYSMOD — це методологія загального призначення, яка добре працює з мовою моделювання SysML. Виходячи з конкретної мети, можна вивести методологію на основі SYSMOD (методологія, керована цілями). Елементи моделі SysML походять із запропонованої методології (моделювання на основі запитів) [17].

На рисунку 2.6 показано інтеграцію методології разом із мовою та інструментами. Важливою характеристикою SYSMOD є те, що його можна модифікувати відповідно до конкретної мети. Підхід SYSMOD відповідає широко використовуваному шаблону: ідентифікувати елемент, описати певний контекст (зовнішній вигляд), а потім занурити (внутрішній вигляд) [18]. Крім того, підхід SYSMOD зосереджений на системному моделюванні за допомогою таких мов, як UML, SysML, AutoStar тощо. З усіх доступних мов мова SysML належним чином підходить для моделювання вимог, оскільки вона надає конкретні позначення та діаграми для вимог. Різні

діаграми SysML допомагають і структурують системне моделювання вищого рівня. Крім того, зміна схеми управління проектом у бік етапного стробування, додатковою перевагою підходу SYSMOD є те, що він не залежить від підходу «водоспад із зворотним зв'язком» або «стадійного стробування». Це ортогональний аспект, і обидва можливі за допомогою SYSMOD [19]. Підсумовуючи, вищезазначені моменти підтверджують причину вибору методології SYSMOD.

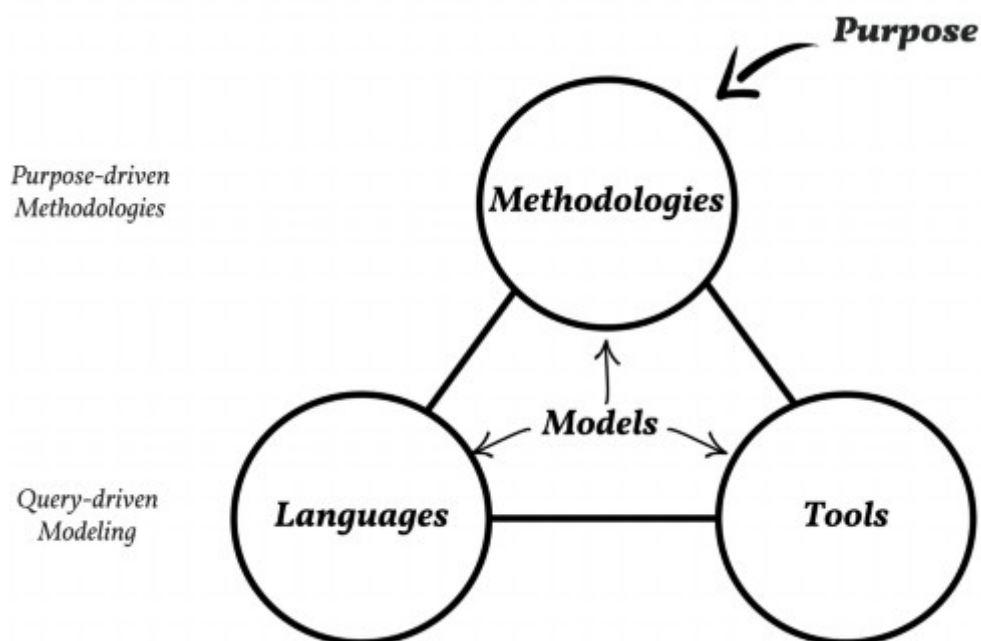


Рис. 2.6. Платформа SYSMOD зі своїм набором мов та інструментів

2.2.4. Аналіз інженерних методів вимог

Запропонована модель включає декілька видів діяльності з RE. Існує багато методів, доступних для виконання заходів з розробки вимог. Для вибору відповідної техніки сформульовані певні критерії. Техніки вибираються на основі певних критеріїв, які перераховані нижче:

- Проста і недорога техніка.
- Дотримуватися принципів системності.
- Сумісний з наявними інструментами.
- Забезпечує кращий огляд системи.

Виявлення вимог — це практика дослідження та виявлення вимог до системи від зацікавлених сторін [20]. Цього можна досягти багатьма способами. Існує багато методів виявлення вимог, таких як інтерв'ювання, створення прототипів, анкетування, рольові ігри, спостереження тощо дотримуються в багатьох організаціях [8]. На основі затребуваних критеріїв вибирається недорога і практична методика виявлення – анкетування. Це задовольняє один із принципів системного підходу, який називається методом постійних запитань, який викликає свіжі думки та інтуїцію серед зацікавлених сторін. Анкета також не потребує додаткових ресурсів чи інструментів, на відміну від прототипування та рольової гри. Щоб посилити техніку анкетування, введено *Zachman*-структуру разом із анкетною, яка стимулює максимальну кількість запитань. Крім того, *Zachman*-структура разом з опитувальником спрямована на вирішення потенційних непорозумінь, відсутності розуміння в конкретних технічних сферах і проблем, пов'язаних з ієрархією відповідальності з точки зору розробки машин.

Аналіз вимог фокусується на завданнях, які визначають потреби проекту. Такий аналіз допомагає усунути невизначені та суперечливі вимоги [12]. Аналіз зосереджується на визначенні доступних ресурсів, вивчає сучасні технології та аналізує набір уже зібраних вимог. Подібно до виявлення вимог, аналіз можна проводити за допомогою різних методів, таких як аналіз прогаєтин, тощо. Ці вищезгадані методи вимагають високого технічного досвіду та ресурсів. Техніка має бути простою для адаптації інженерів і може бути реалізована за допомогою доступних інструментів. Контрольний список є одним із найпростіших і доступних способів збору ресурсів і аналізу вимог. Детальне пояснення щодо використання контрольного списку в популяції ресурсів подано в розділі 3. Вивчення доступних ресурсів допомагає генерувати нові ідеї для мозкового штурму під час розробки концепції. Для аналізу вимог використовується техніка під назвою «формалізація вимог». Основною метою цієї методики є

співвіднесення зібраних вимог із попередньо заповненим контрольним списком для досягнення сформульованих вимог у формі специфікації проекту. Це допомагає виявити невідомі та суперечливі вимоги. Важливо, що він дотримується принципів системного підходу – аналізу. Крім того, форма вимог є простою та підходить для доступних інструментів.

Термін «моделювання системи» має кілька значень. У цьому контексті це стосується використання моделей для концептуалізації та побудови систем за допомогою сформульованих вимог із попереднього кроку. Створена модель вищого рівня допомагає інженерам аналізувати та перевіряти вимоги. Посилаючись на аналіз проблеми, одним із важливих недоліків, які зараз домінують, є пошук рішень, а не функціональне мислення. Цю проблему можна подолати за допомогою функціональної декомпозиції, яку можна виконати на етапі розробки концепції.

Концепція функціональної декомпозиції реалізована за допомогою різноманітних діаграм SysML. Вибір концепції та оцінка виконуються, як запропоновано в [16], що допомагає збільшити простір рішення продукту та уникнути непрактичних рішень. Крім того, системне моделювання дотримується принципів системного підходу, а саме абстракції, синтезу та методу систематичного варіювання.

Нарешті, перевірка вимог перевіряє вимоги, визначені для розробки, а необхідна документація враховує всі вимоги. Існують різні техніки та шаблони для цього. Перевірка вимог не входить до сфери цієї роботи. Перевірка вимог є основним джерелом документа вимог. Однак інші види діяльності з RE також сприяють документу вимог, як показано на рисунку 2.5. Діаграматичне представлення вимог і зв'язків між вимогами взято з діаграми вимог SysML. Документація є однією з критичних проблем компанії. Для покращення класифікації вимог під час документування пропонується шаблон із ISO: системна і програмна інженерія – Requirements engineering [21]. На цьому аналіз методів RE завершується.

2.3. Дослідження та опис сутності методології

Теоретичні основи забезпечують підтримку вибраних концепцій, що стосуються теми дослідження. Перш ніж переходити до методології для робочої процедури, необхідно ознайомитися з теоретичними передумовами щодо структури Zachman, формалізації вимог і розширеного підходу SYSMOD, щоб отримати чітке розуміння наступних розділів.

2.3.1. Опис фреймворку Zachman

Фреймворк Zachman — це онтологія, яка забезпечує формальний і структурований спосіб перегляду та визначення організації або системи. Онтологія є перетином між двома історичними класифікаціями. Перший — примітивні питальні: Що, Як, Коли, Хто, Де, Чому. Другий є похідним від реіфікаційних перетворень, якими є: ідентифікація, визначення, репрезентація, специфікація, конфігурація та інстанціювання. Фреймворк Zachman забезпечує структуру з різними поглядами зацікавлених сторін. Використання фреймворку зумовлено збільшенням складності та зміною природи промислових продуктів [22].

Фреймворк є загальною схемою класифікації для артефактів дизайну, тобто описових представлень будь-якої складної системи. Використання такої схеми класифікації дозволяє зосередитися на вибраних аспектах системи. Кожна клітинка в структурі (перетин стовпця та рядка) надає унікальне представлення або вигляд системи, як зображено на рисунку 2.7. Стовпці шаблону окреслюють основи системи залежно від контексту запитання (що, як, де тощо), тоді як рядки представляють перспективу кожної зацікавленої сторони, залученої до проекту. Матриця 6Х6 наповнена процесами, основними матеріалами, важливими ролями, відповідними місцями та цілями або правилами, пов'язаними з проектом, на основі фундаментального питання та точки зору, представленої в кожній клітинці [23].

ZFI Zachman Framework						
The Zachman Framework	DATA What	FUNCTION How	NETWORK Where	PEOPLE Who	TIME When	MOTIVATION Why
SCOPE (Contextual) Planner	Things Important to the Business 	Processes the Business Performs 	Locations in which the Business Operates 	Organizations Important to the Business 	Events/Cycles Significant to the Business 	Business Goals/Strategies
BUSINESS MODEL (Conceptual) Owner	Conceptual Data Model 	Business Process Model 	Business Logistics 	Work Flow Model 	Master Schedule 	Business Plan
SYSTEM MODEL (Logical) Designer	Logical Data Model 	Application Architecture 	Distributed System Architecture 	Human Interface Architecture 	Processing Structure 	Business Rule Model
TECHNOLOGY MODEL (Physical) Builder	Physical Data Model 	System Design 	Technology Architecture 	Presentation Architecture 	Control Structure 	Rule Design
DETAILED REPRESENTATIONS Sub-Contractor	Data Definition 	Program 	Network Architecture 	Security Architecture 	Timing Definition 	Rule Specification
FUNCTIONING ENTERPRISE	Data 	Function 	Network 	Organization Units 	Schedule 	Strategy

Рис. 2.7. Огляд фреймворку Zachman

2.3.2. Адаптований фреймворк Zachman

Структура Zachman в пропонованій методології в основному використовується для виявлення вимог. Запитання були змінені, щоб відобразити контекст компанії та типових зацікавлених сторін проекту. Ролі, відповідальність і відносини між зацікавленими сторонами в типовому проекті визначаються на початковій стадії проекту. Таким чином аналізується вхід від кількох відділів і різних проектів, щоб знайти спільні риси. Оскільки поточний проект в основному зосереджений на RE, пов'язаному з розробкою, що не має великого впливу на виконавчу модель (обсяг) компанії. У результаті було запропоновано вибрати 3 основні перспективи (власник, проектувальник і будівельник), як виділено на рисунку 2.7. Трьох вибраних точок зору достатньо для збору інформації про

проект і представлення відповідальності та стосунків із продуктом у компанії. Питання власника використовуються на початковому етапі проекту, тоді як питання дизайнера та будівельника використовуються на етапі розробки концепції. Оригінальна матриця була адаптована з 6 X 6 до 3 X 6 під час застосування Zachman структури в методології RE для спрощення. Різні запитання, сформульовані з точки зору різних зацікавлених сторін, призводять до кращого взаєморозуміння між зацікавленими сторонами та відкривають ширші можливості для участі в дискусіях. Це забезпечує структуроване мислення під час розробки машини. Універсальність запитань від різних міждисциплінарних груп на основі запропонованої системи покращує процедуру виявлення вимог.

The Zachman Framework	DATA What	FUNCTION How	NETWORK Where	PEOPLE Who	TIME When	MOTIVATION Why	
BUSINESS MODEL (Conceptual) Owner	Conceptual Data Model	Business Process Model	Business Logistics	Work Flow Model	Master Schedule	Business Plan	<i>Defining the scope</i> <ul style="list-style-type: none"> Defining the milestones Managing product backlog Evaluating product progress
SYSTEM MODEL (Logical) Designer	Logical Data Model	Application Architecture	Distributed System Architecture	Human Interface Architecture	Processing Structure	Business Rule Model	<i>Designing fully-functional product</i> <ul style="list-style-type: none"> Engineering Creating digital drawings Defining product specifications
TECHNOLOGY MODEL (Physical) Builder	Physical Data Model	System Design	Technology Architecture	Presentation Architecture	Control Structure	Rule Design	<i>Delivering physical product</i> <ul style="list-style-type: none"> Material handling, fabrication Building and handling product Product testing

Рис. 2.8. Розподіл відповідальності за Zachman моделлю

Фреймворк Zachman використовується для зображення ролей та відповідальності працівників у впорядкованому вигляді. З рисунка 2.8 видно, що в цій методології фокусуються лише три перспективи. Відповідальність власника полягає в тому, щоб визначити обсяг проекту та мету продукту, що включає встановлення етапів для команди проекту, передбачення потреб клієнта, критерії оцінки продукту та прогрес проекту. Відповідальність дизайнера полягає в тому, щоб розробити повністю функціональний продукт, що складається з інженерії продукту, створення ручних або цифрових

креслень та визначення специфікацій продукту. Відповідальність будівельника полягає в тому, щоб поставити фізичний продукт, що включає обробку матеріалів, виготовлення продукту, складання, обробку продукту, як-от транспортування, випробування продукту та введення продукту в експлуатацію.

2.3.3. Фреймворк Zachman як продукт

Кожен зацікавлений учасник має певний набір завдань та відповідальність за продукт під час його розробки. Фреймворк Закмана може бути використаний для зображення аспектів продукту у взаємозв'язку із зацікавленими учасниками.

The Zachman Framework	DATA What	FUNCTION How	NETWORK Where	PEOPLE Who	TIME When	MOTIVATION Why	
BUSINESS MODEL (Conceptual) Owner	Conceptual Data Model	Business Process Model	Business Logistics	Work Flow Model	Master Schedule	Business Plan	<p><i>Problem identification</i></p> <ul style="list-style-type: none"> • Conceptual view of the end product • Usage characteristics of the end product • Business entities and processes
SYSTEM MODEL Logician Designer	Logical Data Model	Application Architecture	Distributed System Architecture	Human Interface Architecture	Processing Structure	Business Rule Model	<p><i>Solution identification</i></p> <ul style="list-style-type: none"> • Logical process flows of the system • Use cases for the system • Stakeholders management
TECHNOLOGY MODEL Physical Builder	Physical Data Model	System Design	Technology Architecture	Presentation Architecture	Control Structure	Rule Design	<p><i>Solution implementation</i></p> <ul style="list-style-type: none"> • Constraints of tools, • Technology and programming languages • Input/output (I/O) devices • Materials

Рис. 2.9. Фреймворк Zachman як продукт

На рисунку 2.9 перший рядок фокусується на концептуальному баченні кінцевого продукту, оскільки власник може подумати про його реальне застосування. Описові представлення відображають характеристики використання кінцевого продукту. Іншими словами, це описує, що власник збираються робити з кінцевим продуктом або як вони його використовуватимуть. Другий рядок працює над перекладом концепцій у представлення вимог з точки зору дизайнера. Дизайнер визначає елементи даних, логічні процеси, варіанти використання та функціональність системи.

Третій рядок представляє перспективу будівельника з достатньою деталізацією, щоб зрозуміти обмеження інструментів, технологій, мов програмування, пристроїв введення/виведення (I/O) або інших необхідних підтримувальних технологій та знань матеріалів, пов'язаних із виробництвом продукту. Підсумком рисунку 2.9 є те, що власник фокусується на ідентифікації проблеми, дизайнер фокусується на ідентифікації рішення, а будівельник фокусується на реалізації рішення.

2.4. Формалізація вимог

Діяльність із розробки системи, така як виявлення вимог, заповнення специфіки проекту, генерація рішень і впровадження, чітко визначені в розробці програмного забезпечення. У розробці програмного забезпечення термін «специфікація проекту» використовується для позначення різних моделей, створених під час процесу проектування. Добре налагоджений модельний процес у розробці програмного забезпечення може призвести до автоматичного синтезу виконуваного коду [24]. Цей тип процесу формування ще не був досягнутий в інженерному проектуванні. Процес формалізації вимог є більш складним, оскільки знання з інженерного проектування більш різноманітні, а концепції апаратного забезпечення не визначені точно, як у сфері розробки програмного забезпечення. Крім того, у зв'язку зі збільшенням складності систем і більш високими вимогами, роль знань в інженерних дисциплінах на концептуальній фазі має бути покращена.

Необроблені вимоги (тобто вимоги, які не були проаналізовані), як правило, виражаються у форматі розповіді. Необроблені вимоги називаються описовими вимогами (NR). Традиційно будь-який процес розробки починається з NR. Так само NR від команди надходить у формі форми запиту на дії (ARF). Ці NR забезпечують основу для проектування. Але це, безперечно, не забезпечує повних знань, необхідних для подальшого процесу проектування. Тому важливо проаналізувати та сформулювати NR, щоб

стати абстрактним, однозначним і простежуваним, іншими словами, добре сформованим.

Існують різні підходи до формалізації NR у формалізовану специфікацію проекту. Більшість пропозицій документоцентричні та трудомісткі. Найбільш підходящим підходом для розробки вимог є орієнтований на знання та керований інформацією. На ранній стадії розробки проекту дизайнери ледве враховують необхідну інформацію та знання, необхідні для проекту. Це відбувається головним чином через відсутність офіційних комунікаційних ліній між різними відділами, легкість робити припущення на основі власного досвіду роботи та відсутність перевірки інформації із зацікавленими сторонами. Належний збір інформації з різних джерел відіграє життєво важливу роль не лише на етапі розробки, але й протягом усього проекту.

Підхід, представлений у запропонованій методології, спрямований на формалізацію NR, як правило, вимог клієнта. Це орієнтований на модель підхід до уточнення та розширення NR, який базується на операціях матриці взаємодії, описаних у [25]. По-перше, моделювання єдиної бази даних контрольного списку вимог (C) з поєднанням доступного контрольного списку вимог (RC) і джерел знань/інформації. По-друге, формалізація вимог до опису за допомогою бази даних контрольного списку вимог. У запропонованій методології більше уваги приділено формалізації NR. Весь підхід розглядається як розширення захоплення вимог за допомогою діаграм SysML. Діаграми SysML надають більше інформації про призначення та функціонування майбутньої системи, що допомагає зібрати більш детальний і точний набір системних вимог. Підводячи підсумок, можна сказати, що наявні знання інтегровані з формалізмом SysML, щоб забезпечити послідовний набір вимог, які використовуються на етапах розробки [26].

Інженерне проектування – це процес, який передусім включає знання фізики, логічне мислення та творчість. Розроблено багато систематичних інженерних методів та інструментів, які допомагають інженерам аналізувати

цілі проекту та стимулюють творчий процес вирішення проблем. Існують різні контрольні списки, доступні в книгах з різною номенклатурою. Для цього підходу контрольний список вимог був адаптований з [16], де пропонується перейти від підходу, орієнтованого на документ, до підходу, орієнтованого на модель. Контрольний список є одним із найважливіших аспектів у процесі розробки вимог для формалізації процесу. Основна перевага використання контрольних списків полягає в їх системності, що допомагає зацікавленим сторонам приймати обґрунтовані та об'єктивні проектні рішення. Крім того, це допомагає записувати інформацію в ієрархічний спосіб, що допомагає визначити пріоритети створення, розподілу та виконання завдань [27].

2.4.1. Логічний підхід до формалізації вимог

На рисунку 2.10 видно, що доступний контрольний список вимог (RC) з літератури [16] та існуюче джерело знань/інформації було використано для створення бази даних контрольних списків вимог (C). Він був створений для досягнення відкритого та простого інтерфейсу, який можна змінювати та розширювати в майбутньому разом із сяючим користувальницьким досвідом. Потім форма запиту на дії (ARF) від команди, відомої як NR, і інформація, зібрана з анкети Zachman framework, аналізуються за допомогою створеної бази даних контрольних списків вимог (C). Логічні/матричні операції означають пошук кореляції між вимогами та контрольним списком. Це допомагає розробити таблицю специфікацій проектування (DS) вищого рівня з доступною інформацією. Таблиця DS оновлюється відповідно до прогресу проекту. Контрольні списки вимог зведені в таблиці в спеціальному інструменті, щоб допомогти інженерам у процесі розробки. Приклад матриці взаємодії між описовими вимогами (NR) і базою даних контрольних списків вимог (C) для конкретного прикладу. Згодом функціональні та поведінкові моделі моделюються за допомогою створеного (DS) з використанням підходу SYSMOD [26].

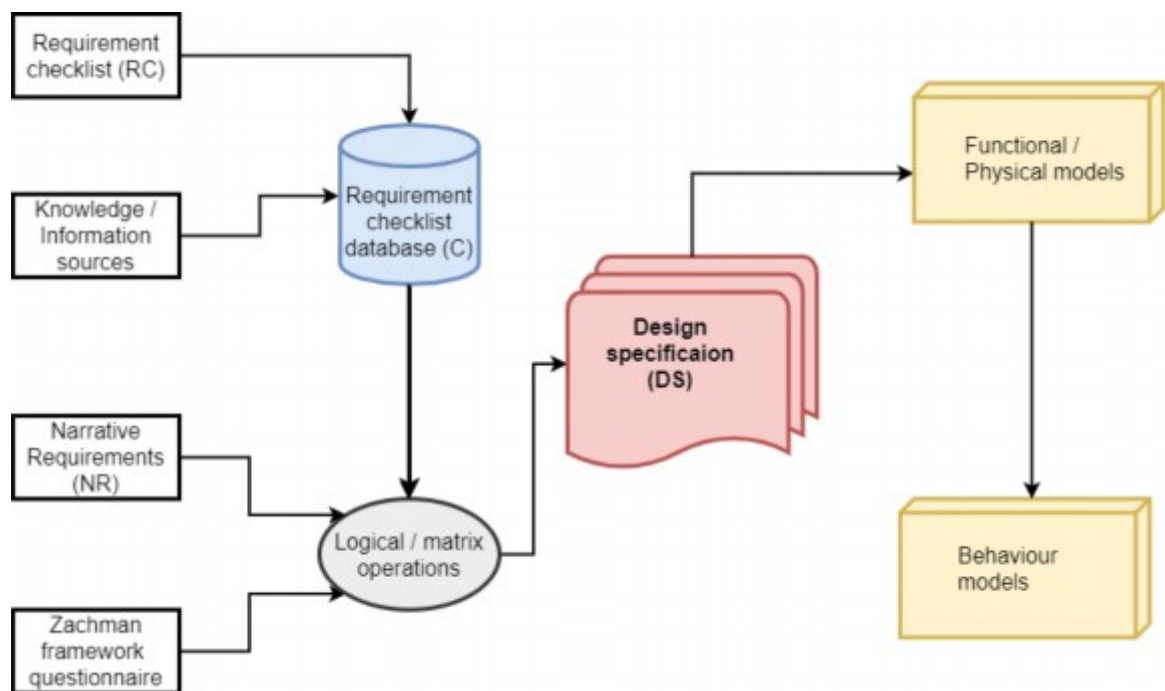


Рис. 2.10. Логічний підхід до формалізації вимоги

2.5. Розширена методологія моделювання систем

Методологія моделювання систем (SYSMOD) — це прагматичний підхід до вимог моделі та архітектури системи. Він надає набір інструментів завдань із вхідними та вихідними робочими продуктами, рекомендаціями та найкращими практиками. SYSMOD використовує мову моделювання систем (OMG SysML) групи керування об'єктами (OMG) [19]. На малюнку 2.11 зображено різні етапи розширеної методології SYSMOD. Теоретичні основи структури Захмана для виявлення вимог і формалізації вимог для аналізу вимог описані вище, які включені як розширення методології SYSMOD. Таке ж колірне кодування як в рисунку 2.5 використовується для того, щоб отримати розуміння кроків, пов'язаних із діяльністю RE. Інші кроки щодо реалізації, перевірки та документу вимог не входять до сфери цієї роботи, яка відрізняється меншою непрозорістю на рисунку 2.11. Різні кроки, які беруть участь у методології SYSMOD, детально описані в наступному підрозділі:

Етапи розширеної методології SYSMOD:

- Виявлення та аналіз вимог. Як уже обговорювалося в попередньому розділі, визначення вимог здійснюється за допомогою системи Захмана за допомогою анкети, а аналіз вимог виконується за допомогою аркуша ідентифікації ресурсу та техніки формування вимог.

- Контекст проекту. Цей крок призначений для визначення контексту проекту для системи. Він представляє спільну мету, над якою працюватиме окрема особа або вся команда. Контекст проекту містить параграф місії, межі проекту та пріоритети з попередніх кроків. Визначення контексту проекту є основним у визначенні наступних кроків.

- Системний контекст. Наступним кроком є системний контекст. Він представляє всі зовнішні сутності, які можуть взаємодіяти з системою. Зовнішні об'єкти включають середовище, акторів, інші системи тощо. Мета діаграми контексту системи полягає в тому, щоб зосередити увагу на зовнішніх факторах і подіях, які слід враховувати при розробці системи.

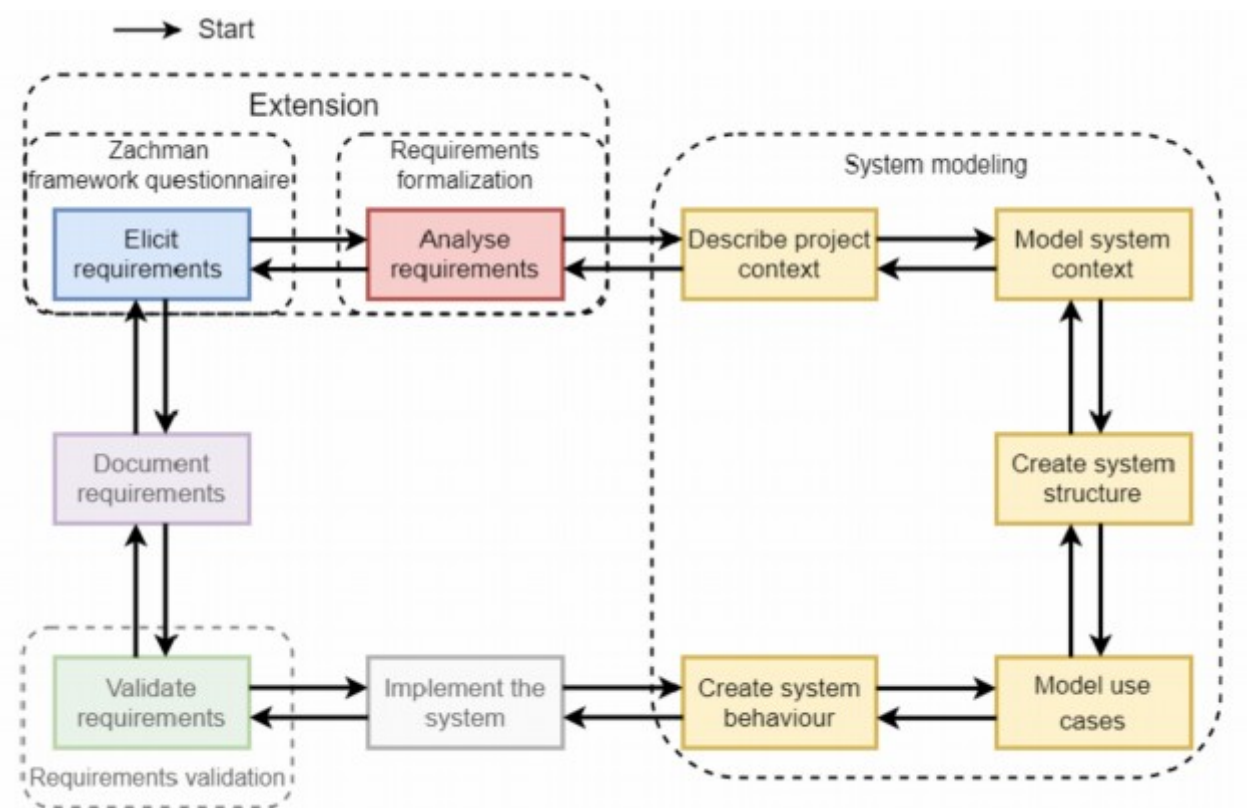


Рис. 2.11. Розширена методологія SYSMOD

- Структура системи. Цей крок призначений для визначення структури системи. Структура системи показана у вигляді блок-схеми визначення. Ця діаграма представляє класифікацію систем та їх ієрархії. Він показує компоненти системи, їхній зміст (властивості, поведінку, обмеження), інтерфейси та зв'язки. Взаємодія між підсистемами також може бути представлена внутрішньою схемою визначення блоку.

- Варіант використання. Наступним важливим кроком є визначення сценарію використання системи. Випадок використання — це опис поведінки системи. Його можна представити у вигляді послідовності простих кроків. Діаграма варіантів використання передбачає взаємодію з варіантом використання та акторами, а також взаємодію між варіантами використання з різними зв'язками. Він забезпечує спрощене та графічне представлення вищого рівня системи.

- Поведінка системи - цей крок призначений для визначення динамічної поведінки системи. Поведінка системи представлена різними діаграмами, такими як діяльність, послідовність, кінцевий автомат тощо, які використовуються для візуалізації динамічних аспектів і поведінки системи або процесу.

- Впровадження. Цей крок полягає в тому, щоб реалізувати розроблені концепції та зробити систему працездатною моделлю. Реалізація суб'єктивна відповідно до визначення системи, яке не входить до сфери цієї дисертації.

- Перевірка вимог і документація. Нарешті, перевірка вимог виконується на наступній стадії проекту, а класифікація вимог для документа вимог адаптована з шаблону ISO [21].

Висновки до розділу

У цьому розділі, на основі численних аналізів та огляду наукової літератури, була обрана методологія, яка теоретично найбільш підходить для етапу проектування в контексті інженерії вимог. Основний акцент

запропонованої методології зосереджено на діяльності, пов'язаній із розробкою вимог, зокрема на виявленні, аналізі та моделюванні системи, оскільки ці процеси є ключовими для успішної реалізації будь-якого проєкту. Під час впровадження цієї методології були адаптовані види діяльності, які відповідають потребам компанії, а також визначено керівні принципи для їх виконання. Враховуючи відсутність спеціалізованих системних інженерів, методологію було розроблено таким чином, щоб її могли ефективно використовувати інженери без глибоких знань у галузі системної інженерії.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ МЕТОДИКИ СИСТЕМНОГО ПІДХОДУ ДО ПРОЕКТУВАННЯ НА ОСНОВІ ІНЖЕНЕРІЇ ВИМОГ

3.1. Особливості пропонованої методології

В цьому розділі сформульовано методологію процедури проектування та обговорено керівні принципи.

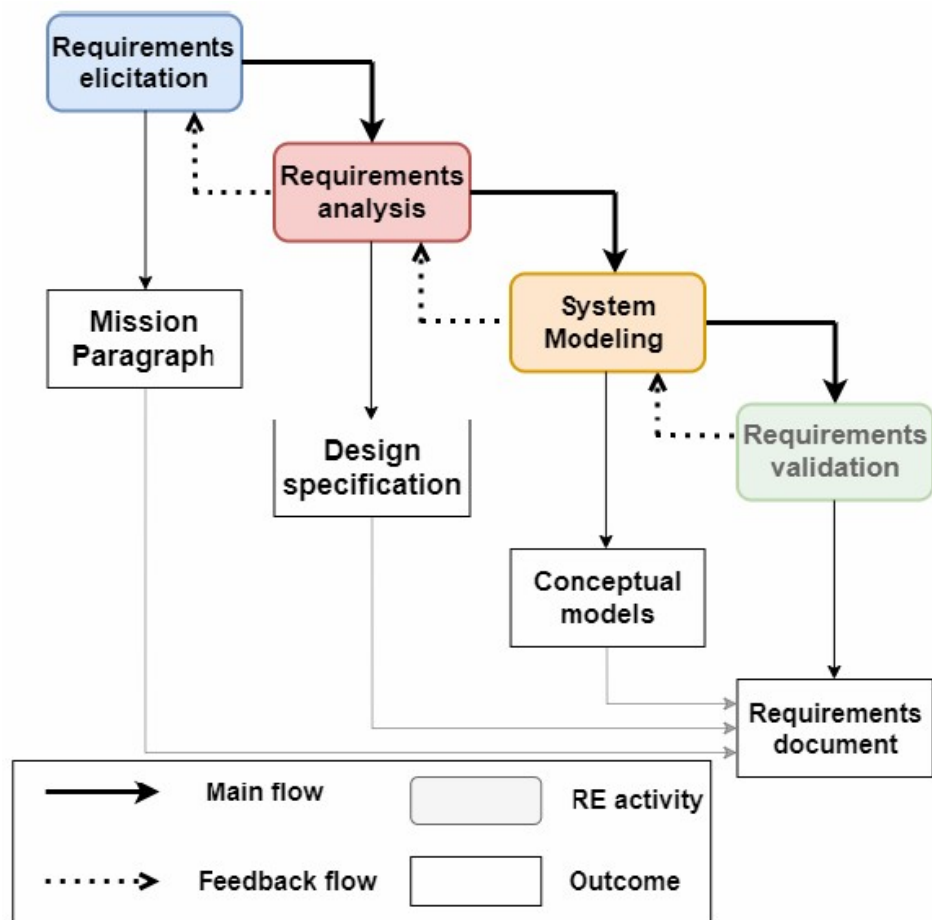


Рис. 3.1. Результати діяльності RE

На основі аналізу інженерних моделей вимог запропонована модель яка показана на рисунку 2.5 у попередньому розділі. На рисунку 3.1 показано абстрактне уявлення про асоціації та результати вибраних видів діяльності з RE. Діяльність RE, така як виявлення, аналіз і моделювання системи, виконується за допомогою вибраних методів із попереднього розділу для

отримати розуміння основних видів діяльності RE. Послідовні кроки методології вписуються в етап ініціювання проекту та розробки концепції. Інформація з кожного етапу методології вноситься до бази даних вимог, яка підтримує базову фазу проектування. Методологія містить цикл перегляду попереднього кроку, який дозволяє вносити зміни або модифікації. Наприклад, якщо на кроці специфікації проекту є суперечливі вимоги, дизайнер повертається до попереднього кроку та змінює вимоги після обговорення їх з необхідними зацікавленими сторонами. Як обговорювалося раніше, методологія виконується з використанням розширеного підходу SYSMOD.

3.2. Представлення та опис етапів методології

Параграф місії є першим кроком у методології. Головна мета цього кроку – зрозуміти контекст проблеми. Це досягається за допомогою відповідних запитань відповідним зацікавленим сторонам у компанії, аналізу відкритих відповідей і вилучення ключових ідентифікаторів, які пояснюються в наступних підрозділах.

Анкета є одним із недорогих і швидких способів отримати інформацію від зацікавлених сторін. Це також практичний спосіб збору даних. Крім того, анкета дозволяє легко аналізувати дані. Включення правильних запитань залежно від проекту і дає позитивні результати. Крім того, система Захмана в поєднанні з анкетною гарантує, що інформація щодо процесу проектування розглядається цілісно. Як обговорювалося в попередньому розділі, сам фреймворк Zachman діє як інструмент для виявлення вимог. Вертикальні колонки містять запитання, які корисно застосовувати на будь-якому рівні [30]. Питальні слова показані в порядку зменшення важливості на рисунку 3.3. Однак під час виконання запитання розташовуються в оригінальному порядку. Як обговорювалося раніше, горизонтальний ряд каркаса було зведено до трьох. У спрощених термінах горизонтальний ряд є ієрархією

працівників. Він починається з точки зору власника і закінчується точкою зору розробника.

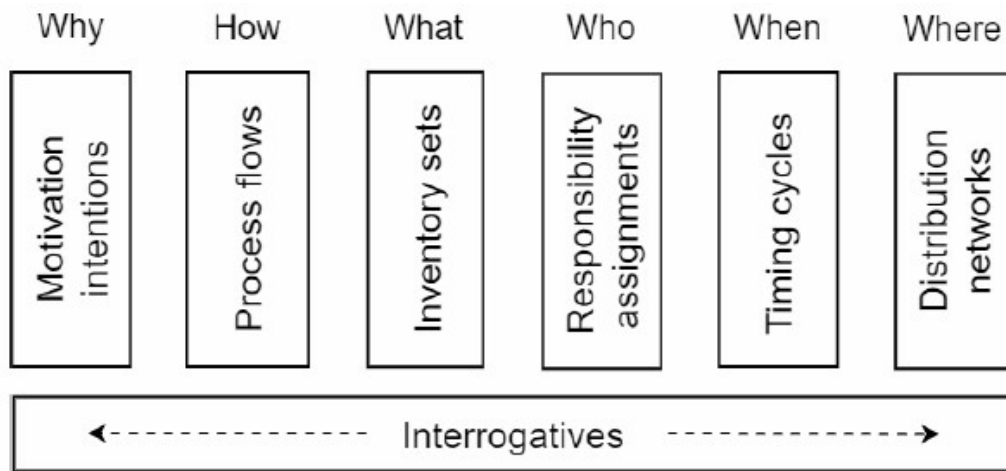


Рис. 3.3. Структура запитань згідно Zachman фреймворк

Питання дизайнера та розробника заповнюються на пізнішому етапі методології. Основна мета цих питань з цієї точки зору полягає в тому, щоб знайти ідентифікацію проблеми, а не ідентифікацію рішення. З точки зору продукту ці питання відповідають на такі аспекти:

- Концептуальне бачення кінцевого продукту.
- Характеристики використання кінцевого продукту.
- Реальне застосування продукту.

На запитання відповідають власник і різні зацікавлені сторони. По-перше, ключовими аспектами є розуміння запитальних категорій та аналіз відповідей. Це полегшує пошук питань, на які немає відповіді. Однак на всі запитання неможливо відповісти описово на початку проекту. По-друге, деякі питання, на які немає відповіді, з'ясовуються з клієнтом, що робить спілкування зрозумілим і відповідальним. Відповіді записуються зв'язно за чергою. Аналіз проводиться за допомогою ідентифікаторів.

Ключові ідентифікатори виводяться для отримання необхідної інформації про проект. Аспект аналізу полягає у зв'язуванні ідентифікаторів

із отриманими відповідями з анкети. Зразок списку ідентифікаторів наведено нижче:

- Мета/ціль цього проекту;
- Основна причина проблеми;
- Переваги нового продукту;
- Для вирішення проблеми використано попереднє рішення;
- Спільні риси, які можна використовувати в новому продукті, якщо такі є;
- Деталі зустрічі;
- Розклад;
- Відповідальність.

Параграф місії є результатом даного етапу дослідження. Він формулюється шляхом інтеграції відповідей, отриманих після аналізу. Параграф місії відповідає на перелік ідентифікаторів, зазначених раніше. Чіткий та логічний параграф місії продукту має потенціал для оптимізації процесу розробки продукту. Він виконує роль орієнтиру під час процесу проектування та сприяє наступним аспектам проекту:

- спрощенню конструкції продукту;
- розробці привабливих продуктів;
- ефективній комунікації між зацікавленими сторонами;
- підвищенню розуміння продукту командою;
- раціоналізації та оптимізації внутрішніх процесів.

3.3. Структуризація та формалізація вимог

У цьому розділі основна увага полягає в досягненні таблиці специфікацій проектування (DS) вищого рівня. Технічне завдання на проектування — це документ, що містить перелік пунктів щодо продукту. Існує етап дій, які виконуються для досягнення DS продукту вищого рівня. Як обговорювалося раніше, формалізація вимог допомагає формалізувати

необроблені та описові вимоги до належної таблиці DS. Існують також інші дії, такі як визначення ресурсів і вивчення теми, що допомагає інженерам розробляти економічний продукт. Нарешті, також охоплено уявлення про структурування вимог і використання правильної термінології під час формування вимог.

Дослідження є невід'ємною частиною розробки продукту. Результати дослідження дизайнерських рішень дають проекту конкурентну перевагу, а також більш розвинене розуміння довгострокової стратегії проектування значно підвищує шанси на створення успішного продукту [31]. Дослідження можна проводити різними способами, наприклад, за допомогою даних у реальному часі (проспективні дослідження) і минулих даних (ретроспективні дослідження). Цього можна досягти багатьма способами, як показано на рисунку 3.4.

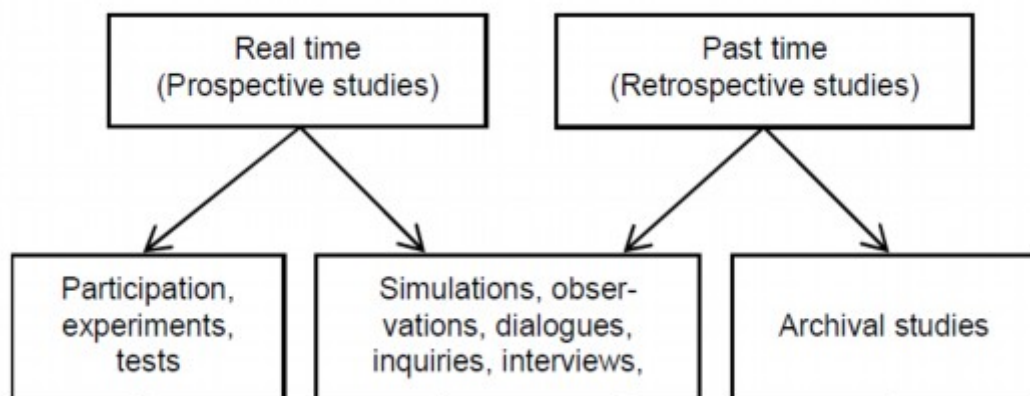


Рис. 3.4. Варіанти схем досліджень

По-перше, треба перевірити основну причину проблеми, як впливає з параграфа місії, що допомагає зрозуміти призначення. Під час дослідження враховується інформація з ідентифікації ресурсів. Список ключових термінів може бути отриманий залежно від проекту, який можна досліджувати в базі даних наукових статей. Нові сучасні рішення та технології можуть бути отримані за допомогою патентів і публікацій, змінених залежно від проекту, і можуть бути повторно використані. По-друге, найпоширенішим способом є

проведення додаткового дослідження через інтранет-базу, де можна отримати інформацію про проекти різних відділів. Цю інформацію можна ефективно адаптувати та повторно використовувати для поточного проекту. Концептуальні ідеї вищого рівня можна обговорити з відповідним фахівцем, щоб спрямувати процес розробки в правильному напрямку.

Формалізація вимог — це техніка для перетворення описової вимоги (NR) у таблицю специфікації проекту (DS) за допомогою бази даних контрольного списку вимог (C). Теоретичні основи формалізації вимог та логічний підхід до формалізації вимог викладені в другому розділі. Вся методика виконується послідовно за такими етапами:

- Початкове налаштування вимог;
- Відображення понять сутності на абстрактні поняття;
- Створення матриці взаємодії;
- Створення таблиці специфікації конструкції.

Під час процедури спочатку збирається описова вимога (NR) у формі документа який називається «Формою запиту на дії». Опис запитаної дії містить необхідну інформацію про NR. Додаткова інформація з параграфу місії додається до NR. Після збору NR формується початкова таблиця налаштування вимог (R). Початкова таблиця вимог складається з номера вимоги, опису вимоги та функцій і параметрів. По-друге, NR класифікується за номерами вимог. Індивідуальна вимога розрізняється дієсловами та прислівниками (f_i - поняття функції) та іменниками та прикметниками (p_i - поняття параметра). Частина контрольного списку вимог показано на рисунку 3.5.

У цій роботі основні критерії проектування називаються «аксіомами», а підрозділи цих критеріїв — «сутностями». Рисунок 3.5 показує зразки списків аксіом і список понять сутності. Сутності e_i з C використовуються для отримання абстрактних понять (f_i , p_i в R). Це називається відображенням понять сутності на абстрактні поняття. По-третє, створюється матриця взаємодії. Вісь X матриці – це вимоги, а вісь Y – попередньо визначений

контрольний список (аксіоми). Вони корелюються для формування матриці (тобто взаємодія між R і C моделюється за допомогою матриці взаємодії). Нарешті, таблиця специфікації проекту (DS) вищого рівня створюється для визначених вимог (R) у різних попередньо визначених категоріях (аксіомах). Існують різні переваги створення таблиці специфікації проекту вищого рівня. Перераховуємо основні переваги DS:

- Призначення відповідальності за критеріями дизайну.
- Увага на неповну інформацію про вимоги.
- Виявлення суперечливих вимог.

Axioms	abbr.	Entity concepts	abbr.
Function	C ₁	Branch	e ₁
		Channel	
		Convert	
		
Geometry	C ₂	Size	e ₂
		Number	
		Breadth	
		
.....

Рис. 3.5. Фрагмент переліку вимог

Цей процес є ітеративним, і описана вище процедура застосовується, щоб зробити інформацію DS вищого рівня правильно сформованою без будь-якої двозначності та неповноти. Таблиця DS повинна містити відповідну інформацію, щоб перейти до етапу концептуальної розробки. Суперечливі вимоги обговорюються із зацікавленими сторонами та вирішуються. У міру просування проекту таблиця DS відповідно оновлюється.

Вимога – це твердження, яке перекладає або виражає потребу та пов’язані з нею обмеження та умови. Вимога - це опис системи, яка має бути розроблена. Опис завжди пишеться природною мовою і має містити підмет, дієслово та об’єкт. Синтаксис для визначення напівформальних вимог адаптовано з ISO29148 [21], який описано нижче:

[Умова] [Тема] [Дія] [Об'єкт] [Обмеження]

- [Умова] – коли застосовна вимога, наприклад, отримати сигнал.
- [Предмет] – актор, наприклад, «програма», «система», «програмне забезпечення».
- [Дія] - дія або вимоги, наприклад, «виконаю», «відправлю».
- [Об'єкт] - дії, наприклад, «повідомлення», «книга», певний стан.
- [Обмеження дії] - обмеження дії, наприклад, часовий ліміт.

Умову або обмеження можна опустити в синтаксисі, якщо це не є необхідним для вимог. Цей синтаксис дає загальний огляд, але не містить суворих правил адаптації. Однак природна мова сприйнятлива до помилок. Під час написання вимог уникайте розпливчастих і загальних термінів. Вони призводять до вимог, які часто важко або навіть неможливо перевірити. Крім того, це створює проблему для багатьох інтерпретацій.

3.4. Розробка концепції на основі інженерії вимог

Цей крок у методології є одним із найважливіших кроків. Він передбачає створення концептуальної моделі продукту за допомогою функціонального аналізу. Питання дизайнера та розробника сформульовані, щоб допомогти етапу розробки концепції. Рекомендації щодо функціональної декомпозиції продукту обговорюються з використанням підходу SYSMOD. Потім пропонується схема вибору концепції. Нарешті, процедури оцінки концепції обговорюються за допомогою таблиці систематичної оцінки. Результатом цього кроку є повністю визначена концептуальна модель, яка завершується різними ітераціями командою проектувальників та відповідними зацікавленими сторонами.

Функції системи та взаємозв'язок входу та виходу впливають із вимог. Загальна функція зображена блок-схемами на основі входів і виходів системи. По-перше, система розглядається з точки зору її входів і виходів без

будь-яких знань про її внутрішню роботу. Це називається концепцією чорної скриньки, і вона походить від підходу SYSMOD. Потім створюється структура системи з точки зору функціональності системи.

Залежно від складності комплексна загальна функція розбивається на підфункції меншої складності. Кількість підфункціональних рівнів визначається на основі відносної складності проблеми. Поєднання окремих підфункцій призводить до функціональної структури, яка -представляє загальну функцію системи. Переваги поділу складних функцій:

- Визначення підфункцій, а також допомагає подальший пошук підфункціональних рішень.
- Підхід до масштабованого проектування.

Після функціональної декомпозиції створюється таблиця підфункцій. Як правило, функція системи може бути досягнута n кількістю способів. Так само кожна підфункція має численні рішення. У цій таблиці показано підфункції та їхні відповідні рішення (рис. 3.6) який був адаптований з [16]. Рисунок 3.6 також символізує можливі комбінації принципів, які можуть бути потенційним варіантом рішення для продукту. Варіанти рішення оцінюються на наступному кроці.

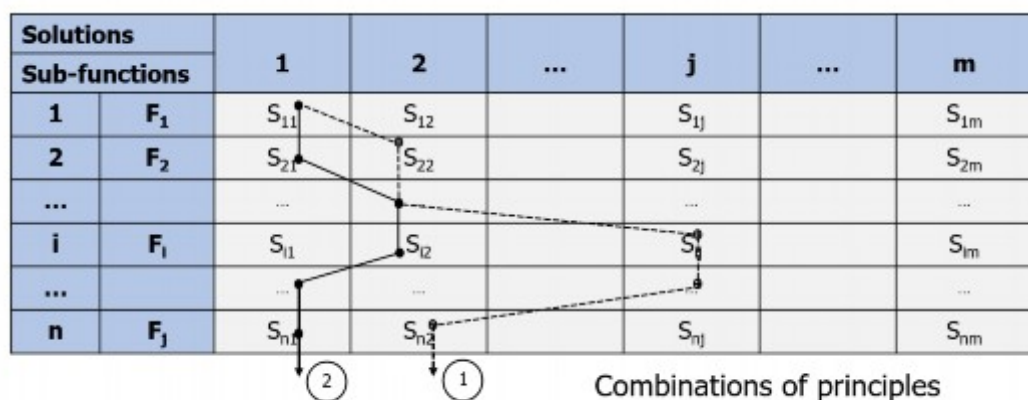


Рис. 3.6. Діаграма вибору підфункції

Використання процедури системного відбору сприяє вибору перспективних рішень. Вибір підфункції здійснюється двома етапами, а саме

виключенням і перевагою. По-перше, усуваються всі недоречні пропозиції. Якщо залишається занадто багато можливих рішень, перевага надається тим, які явно кращі за решту. Розробляються та оцінюються лише можливі рішення. Таблиця систематичного відбору подана на рисунку 3.7, де 1, 2, 3 тощо є варіантами рішень пропозицій, а стовпці є критеріями відбору. Також символами позначають варіанти рішення в таблиці вибору.

Allseas evaluation chart						
Solution variants	Selection criteria:				Decision:	
	(++) Excellent (+) Good (?) Lack of information (-) Bad (--) Worst				(+) Pursue solution (-) Eliminate solution (?) Re-evaluate solution	
	Selection criteria				Remarks	Decision
Demand of Design specification	Realisable in principle	Within cost allocated	Resource and expertise availability			
1						
2						
3						
4						

Рис. 3.7. Діаграма оцінки підфункції

Критерії відбору встановлюються на основі рішень «так/ні», кількісного розгляду, безпеки тощо. До таблиці вибору додаються деякі основні критерії, такі як сумісність, повне виконання вимог, реалізованість у принципі та в межах допустимої вартості. Критерії вибираються на основі оцінки технічних, безпечних, екологічних та економічних цінностей. На основі проекту змінюються критерії та їх порядок.

Діаграма вимог — це статична структурна діаграма, яка показує ієрархію вимог системи. SysML має спеціальну діаграму вимог, яка використовується для представлення вимог та їхніх зв'язків. Попередньо визначені позначки на діаграмі вимог забезпечують суттєву та центральну частину переглядів простежуваності та є фундаментальним аспектом підходу на основі моделі (SYSMOD) до розробки вимог. Важливим є те, що модель,

створена на концептуальній фазі, може бути підтверджена вимогами. Це робить концептуальну модель раціональною та придатною для перевірки.

Нотація, яка використовується в діаграмах вимог SysML, зведена в одну діаграму та показана на рисунку 3.8. Кожна вимога в SysML відображається в прямокутнику зі стереотипом «вимога». Він складається з двох властивостей `id#` і `txt`. Ідентифікатор `#` є унікальним для кожної вимоги, а текстовий файл містить опис вимоги.

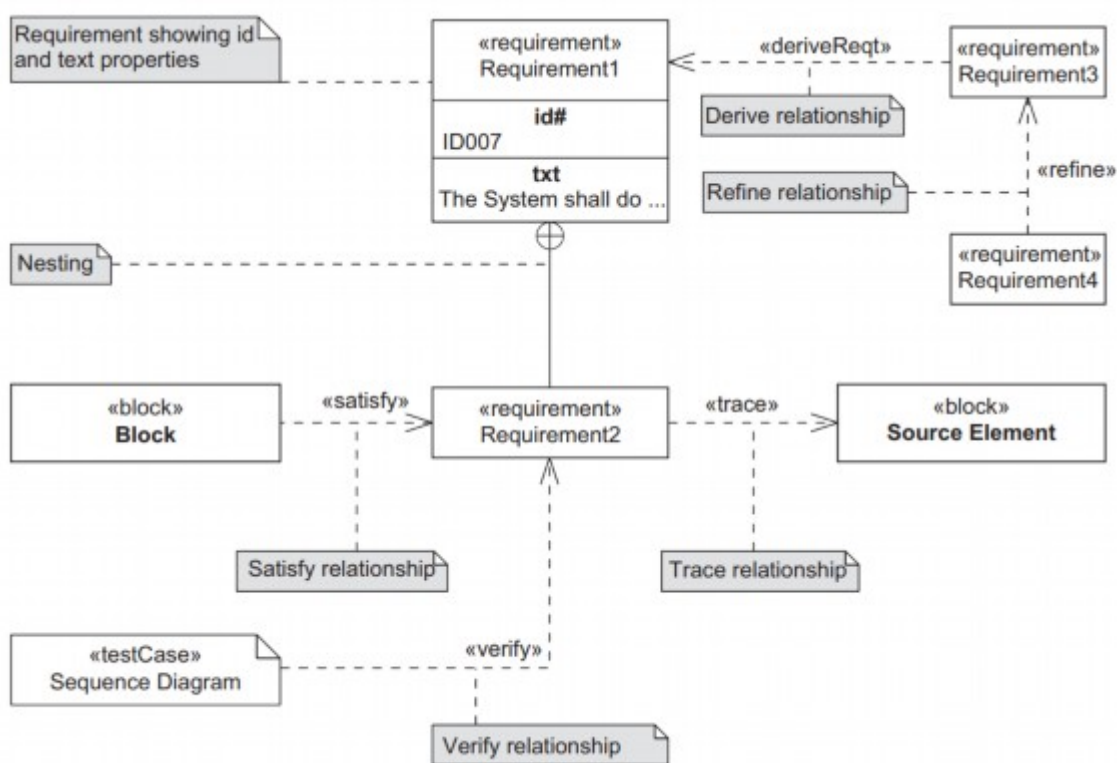


Рис. 3.8. Коротка нотація діаграми вимог

Діаграма вимог SysML підтримує різні типи зв'язків, які показано на рисунку 3.8, які використовуються таким чином [32]:

- Вкладений зв'язок – вимоги розкладаються на одну або декілька підвимог. У SysML ця декомпозиція відома як вкладеність, яка позначається зв'язком вкладеності, таким як показано між «Вимога1» і «Вимога2».
- Похідний зв'язок – похідна вимога не вказана зацікавленою стороною.

- Відношення задовольняє – використовується для зв'язку концептуальної моделі з вимогами, які елементи мають задовольнити.
- Зв'язок трасування - це забезпечує зв'язок загального призначення, який дозволяє зв'язувати елементи моделі та вимоги один з одним. Прикладом є зв'язок трасування між «Requirement2» і «Source Element» .
- Уточнення зв'язку – використовується, щоб показати, як елементи моделі або вимоги можна використовувати для подальшого вдосконалення інших елементів моделі або вимог..
- Перевірка зв'язку - використовується, щоб показати, що окремий тестовий приклад перевіряє дану вимогу. Це показано на рисунку 3.8 зв'язком перевірки між тестовим випадком під назвою «Діаграма послідовності» (діаграма поведінки) та «Вимога2».

Підсумовуючи, діаграми вимог SysML пропонують кілька переваг, а саме:

- Стандартизація способу визначення вимог через визначену семантику.
- Краща систематизація вимог і показ різноманітних взаємозв'язків між різними вимогами .
- Легко побудувати та зрозуміти.
- Покращує комунікацію між міждисциплінарною командою.

Висновки до розділу

У розділі 3 було детально розглянуто реалізацію методики системного підходу до проєктування на основі інженерії вимог, що включає кілька важливих етапів і аспектів. Зокрема:

- Особливості пропонованої методології: Визначено основні особливості методології, які полягають у комплексному підході до системного проєктування з акцентом на чітке визначення вимог. Пропонована методологія спрямована на забезпечення інтеграції процесів

аналізу, моделювання та реалізації вимог для забезпечення високої якості кінцевого продукту.

- Представлення та опис етапів методології: У розділі проведено детальний опис етапів методології, зокрема аналізу, розробки та верифікації вимог. Кожен з етапів структуровано з метою досягнення чіткої послідовності дій, що забезпечує ефективність процесу проєктування та мінімізує ризики невідповідності вимогам.

- Структуризація та формалізація вимог: Описано підходи до структуризації та формалізації вимог, які дозволяють забезпечити їх однозначність, повноту та узгодженість. Застосування формальних методів опису вимог сприяє підвищенню точності моделювання та полегшує комунікацію між членами проєктної команди, що позитивно впливає на загальний процес розробки.

- Розробка концепції на основі інженерії вимог: Розглянуто процес розробки концепції на основі детального аналізу вимог, що включає створення моделей системи, прототипів та їх верифікацію. Такий підхід дозволяє формувати обґрунтовані концептуальні рішення, які відповідають потребам зацікавлених сторін і забезпечують функціональність розроблюваної системи.

В цілому, розділ 3 демонструє, що запропонована методологія системного підходу до проєктування на основі інженерії вимог є ефективною для забезпечення точного відображення вимог замовників та їхньої реалізації в проєкті. Методика дозволяє підвищити якість системного проєктування, зменшити ризики, пов'язані з невідповідністю вимог, і забезпечити високий рівень організованості та узгодженості на всіх етапах розробки

ВИСНОВКИ

В магістерській роботі, в результаті проведеного дослідження на тему "Методика системного підходу до проектування на основі інженерії вимог" було розроблено та обґрунтовано ефективний підхід до проектування складних систем, що враховує всі ключові аспекти інженерії вимог. Основні результати дослідження представлені у трьох розділах.

Розділ 1: Дослідження предметної області проектування систем на основі інженерії вимог. Проведено детальний аналіз предметної області, що включає вивчення особливостей фаз проєкту, проблеми розробки та управління вимогами, а також сутності інженерії вимог. Проаналізовано основні проблеми, які виникають на етапі формування вимог, що часто стають причиною невідповідності кінцевого продукту очікуванням замовників. Огляд наявних інструментів для управління інженерією вимог дозволив визначити оптимальні рішення для їхнього застосування в різних типах проєктів.

Розділ 2: Моделі, методи та підходи до проектування на основі інженерії вимог. У цьому розділі здійснено порівняльний аналіз різних моделей інженерії вимог, таких як лінійна модель Маколея, спіральна модель, модель Pandey & Suman, та модель вимог Sommerville. Це дало змогу виділити переваги і недоліки кожної моделі, що стало основою для розробки власної інженерної моделі вимог. Також досліджено фреймворк Zachman і його адаптацію до потреб системного підходу. Особливу увагу приділено формалізації вимог, що дозволяє створити логічну та послідовну базу для розробки та моделювання системи.

Розділ 3: Реалізація методики системного підходу до проектування на основі інженерії вимог. Описано пропоновану методологію, її етапи та підходи до структуризації та формалізації вимог. У розділі розроблено концепцію на основі детального аналізу вимог, що дозволяє забезпечити узгодженість між вимогами та їх реалізацією у проєкті. Методика сприяє

покращенню управління процесом розробки та забезпечує високий рівень інтеграції між різними учасниками проєкту.

У цілому, проведені дослідження підтвердили, що використання системного підходу та інженерії вимог у процесі проєктування забезпечує підвищення якості розробки, зниження ризиків і оптимізацію управління процесом. Запропоновані методики та моделі є надійним інструментом для проєктувальників, що дозволяє ефективно реагувати на потреби замовників і забезпечувати розробку складних систем з високим ступенем відповідності встановленим вимогам.

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Emma Rudeck. Concurrent Engineering Blog. <https://www.concurrent-engineering.co.uk/Blog/bid/90476/Why-Requirements-Engineering-should-matter-to-your-company>.
2. Ion Barosan. SysML lecture slides - Real-time Software Systems Engineering (2IN70).
3. Allseas. Innovations Project Departmental Instructions Overview. [Document number: GE-18055-001-Q-E-001]. 2017.
4. U.S. Department of Defense. Software Technology Strategy. December 1991.
5. John A Dotchin and John S Oakland. "Theories and concepts in total quality management". In: Total Quality Management 3.2 (1992), pp. 133{146.
6. I. Sommerville and P. Sawyer. Requirements Engineering: A Good Practice Guide. Wiley, 1997. isbn: 9780471974444. url: <https://books.google.co.in/books?id=5NnP-VODEc8C>.
7. Ralph Rowland Young. The requirements engineering handbook. Artech House, 2004.
8. Dharendra Pandey, Ugrasen Suman and A.K. Ramani. "An Effective Requirement Engineering Process Model for Software Development and Requirements Management". In: Nov. 2010, pp. 287{291. doi: 10.1109/ARTCom.2010.24.
9. Mona Batra and Archana Bhatnagar. "A Comparative Study of Requirements Engineering Process Model." In: International Journal of Advanced Research in Computer Science 8.3 (2017).
10. Linda A. Macaulay. Requirements Engineering. Berlin, Heidelberg: Springer-Verlag, 1996. isbn: 3540760067.
11. Gerald Kotonya and Ian Sommerville. Requirements Engineering: Processes and Techniques. English. John Wiley Sons, Inc., 1998. isbn: 0471972088.

12. Rajkumar Upadhyay. Spiral model. <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-using-spiral-model/>.
13. Dhirendra Pandey, Ugrasen Suman and A Kumar Ramani. "An effective requirement engineering process model for software development and requirements management". In: 2010 International Conference on Advances in Recent Technologies in Communication and Computing. IEEE. 2010, pp. 287{291.
14. Ian Sommerville. Software Engineering. 9th ed. Harlow, England: Addison-Wesley, 2010. isbn: 978-0-13-703515-1.
15. Gerhard Pahl and Wolfgang Beitz. Engineering design: a systematic approach. Springer Science & Business Media, 2013. url: <https://www.springer.com/gp/book/9781447135814>.
16. Model Based Systems Engineering. <https://mbse4u.com/sysmod/>.
17. Tim Weilkiens. Systems engineering with SysML/UML: modeling, analysis, design. Elsevier, 2011.
18. Tim Weilkiens. SYSMOD-The Systems Modeling Toolbox-Pragmatic MBSE with SysML. Lulu.com, 2016.
19. Sommerville Ian and Sawyer Pete. "Requirements Engineering: A good practice guide". In: China Machine Press, CITIC PUBLISHING HOUSE (1997), pp. 177{206.
20. ISO. IEEE/ISO/IEC 29148-2018 - ISO/IEC/IEEE International Standard - Systems and software engineering - Life cycle processes - Requirements engineering. <https://standards.ieee.org/standard/29148-2018.html>.
21. J.A. Zachman. John Zachman's Concise Denition of the Zachman Framework. <https://www.zachman.com/16-zachman/the-zachman-framework/35-the-concise-definition>.
22. J.A. Zachman. The framework for enterprise architecture: background, description and util-ity. <https://www.zachman.com/resources/ea-articles-reference/327-the-framework-for-enterprise-architecture-background-description-and-utility-by-john-a-zachman>. 2016.

23. S. Douglas. In: Model-Driven Engineering. IEEE Computer, vol. 39, no. 2, pp. 25{31, 2006.
24. Tetsuo Tomiyama, Masaharu Yoshioka and Akira Tsumaya. "A knowledge operation model of synthesis". In: Engineering Design Synthesis: Understanding, Approaches and Tools. Ed. by Amaresh Chakrabarti. London: Springer London, 2002, pp. 67{90. isbn: 978-1-4471-3717-7. doi: 10.1007/978-1-4471-3717-7_5. url: https://doi.org/10.1007/978-1-4471-3717-7_5.
25. W. Brace and K. Thramboulidis. "FROM REQUIREMENTS TO DESIGN SPECIFICATIONS - A FORMAL APPROACH". In: 2010.
26. Nigel Cross. Engineering design methods: strategies for product design. John Wiley & Sons, 2021.
27. Ralph Katz. "The effects of group longevity on project communication and performance". In: Administrative science quarterly (1982), pp. 81{104.
28. IBM. IBM Engineering Systems Design Rhapsody. <https://www.ibm.com/products/systems-design-rhapsody>.
29. Eric Lutters et al. "Tools and techniques for product design". In: CIRP Annals 63.2 (2014), pp. 607{630. issn: 0007-8506. doi: <https://doi.org/10.1016/j.cirp.2014.05.010>. url: <https://www.sciencedirect.com/science/article/pii/S0007850614001929>.
30. Stig Ottosson et al. "Research Approaches on Product Development Processes". In: 9th International Design Conference, DESIGN 2006 (Jan. 2006).
31. Jon Holt and S. Perry. SysML for Systems Engineering, 2nd Edition: A Model-Based Approach. Nov. 2013, pp. 1{935. isbn: 9781849196512. doi: 10.1049/PBPC010E.
32. Friedenthal, S., Moore, A., & Steiner, R. (2015). A practical guide to SysML: The systems modeling language (3rd ed.). Morgan Kaufmann.
33. Long, D., & Scott, Z. (2011). A primer for model-based systems engineering (2nd ed.). Vitech Corporation.

34. Wymore, A. W. (1993). Model-based systems engineering: An introduction to the mathematical theory of discrete systems and to the tricategory theory of system design. CRC Press.
35. Holt, J., Perry, S. A., & Brownsword, M. (2011). Model-based requirements engineering. Institution of Engineering and Technology (IET).
36. Pandian, M. K. S., et al. (2017). Towards industry 4.0: Gap analysis between current automotive MES and industry standards using model-based requirement engineering. In 2017 IEEE International Conference on Software Architecture Workshops (ICSAW) (pp. 29-35). IEEE.
37. Zeigler, B. P., Muzy, A., & Kofman, E. (2018). Theory of modeling and simulation: Discrete event & iterative system computational foundations (3rd ed.). Academic Press.
38. Micouin, P. (2008). Toward a property-based requirements theory: System requirements structured as a semilattice. *Systems Engineering*, 11(2), 101-117.
39. Kossiakoff, A., Sweet, W. N., Seymour, S. J., & Biemer, S. M. (2011). *Systems engineering principles and practice* (2nd ed.). John Wiley & Sons.
40. Bellagamba, L. (2012). *Systems engineering and architecting: Creating formal requirements*. CRC Press.
41. Salado, A., & Nilchiani, R. (2013). Assessing the impacts of uncertainty propagation to system requirements by evaluating requirement connectivity. In 23rd INCOSE International Symposium. Philadelphia, PA.
42. Friedenthal, S., & Oster, C. (2017). *Architecting spacecraft with SysML: A model-based systems engineering approach*. CreateSpace Independent Publishing Platform.
43. INCOSE. (2020). *Systems engineering body of knowledge (SEBoK)*. Retrieved from <https://sebokwiki.org>
44. Douglass, B. P. (2016). *Agile systems engineering*. Morgan Kaufmann.

45. Hause, M., Hummell, J., & Grelier, F. (2018). MBSE driven IoT for smarter cities. In 2018 13th Annual Conference on System of Systems Engineering (SoSE) (pp. 365-371). IEEE.
46. Weiss, E., Chung, L., & Nguyen, L. (2019). A MBSE approach to satellite clock time and frequency adjustment in highly elliptical orbit. In MILCOM 2019 - IEEE Military Communications Conference (pp. 65-69). IEEE.
47. Wach, P., & Salado, A. (2020). The need for semantic extension of SysML to model the problem space. In Conference on Systems Engineering Research (CSER). Redondo Beach, CA.
48. Micouin, P., et al. (2018). Property Model Methodology: A landing gear operational use case. INCOSE International Symposium, 28(1), 321-336.
49. SPEC Innovations. (2020). The ultimate guide to model-based systems engineering (MBSE). Retrieved from <https://specinnovations.com>
50. Friedenthal, S., Moore, A., & Steiner, R. (2015). A practical guide to SysML (3rd ed.). Morgan Kaufmann.
51. Committee, L. S. (2015). Lifecycle modeling language (LML) specification. Lifecycle Modeling Language website. Retrieved from <https://lifecyclemodeling.org>