

**Корнієнко А.М., Коваленко М.А.,**  
ст. гр. 12171,  
Новокаховський політехнічний коледж  
Одеського національного політехнічного університету.  
Науковий керівник – Тиха Т.В.,  
викладач економічних дисциплін

## **КОНФЛІКТ ФУНКЦІОНАЛЬНИХ ТА НЕФУНКЦІОНАЛЬНИХ ВИМОГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Проблеми, які доводиться вирішувати фахівцям у процесі створення програмного забезпечення, зазвичай дуже складні. Опис функціональних можливостей і обмежень, що накладаються на програмну систему, називається вимогами до цієї системи, а сам процес формування, аналізу, документування і перевірки цих функціональних можливостей і обмежень □ розробкою вимог (requirements engineering).

Вимоги до програмної системи класифікуються як функціональні, нефункціональні і вимоги предметної області.

Функціональні вимоги – це перелік сервісів, які повинна виконувати система, причому повинно бути зазначене, як система реагує на ті або інші вхідні дані, як вона поводить себе в певних ситуаціях.

Нефункціональні вимоги – описують характеристики системи і її оточення, а не поведінку системи. Тут також може бути наведений перелік обмежень, що накладаються на дії і функції, виконувані системою.

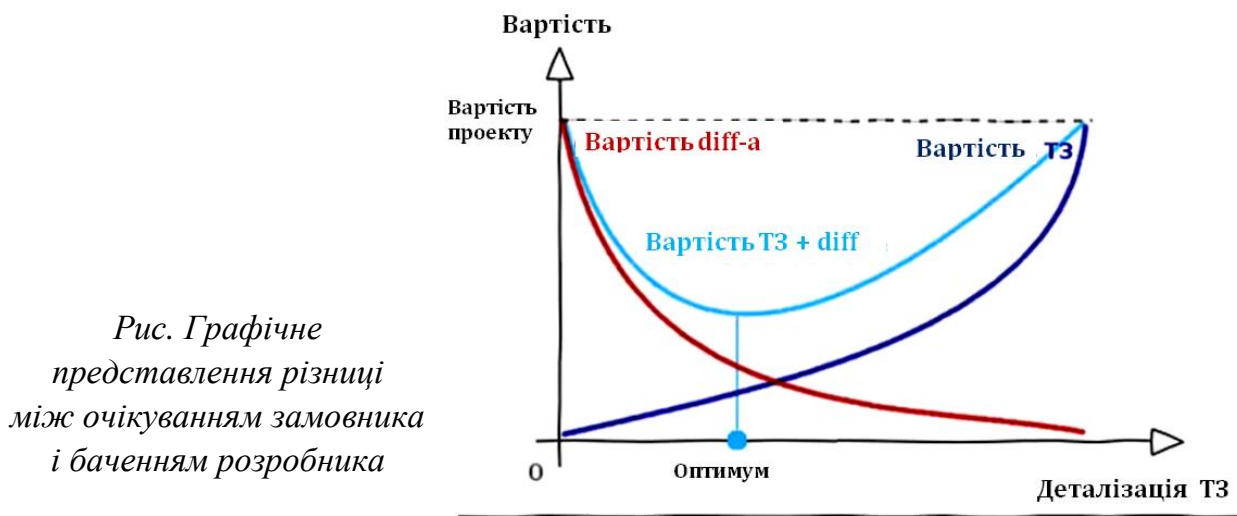
Багато проблем, що виникають при розробці систем, пов'язані з неточністю і “розмитістю” специфікації вимог. Звичайно, розробники інтерпретують вимоги, що допускають двояке тлумачення, так, щоб систему було простіше реалізувати. Але це тлумачення може не збігатися з очікуваннями замовника. Така ситуація приводить до розробки нових вимог і внесенню змін у систему. Це, у свою чергу, веде до затримки здачі готової системи і її подорожчання.

Нефункціональні вимоги часто вступають у конфлікт із іншими вимогами, пропонованими системою. Наприклад, відповідно до однієї із системних вимог розмір системи не повинен перевищувати, наприклад 5 Мбайт, оскільки вона повинна повністю поміститися в постійний запам'ятовувальний пристрій обмеженої ємності. Інша вимога зобов'язує використовувати для написання системи мову програмування, яка займає більше 5 Мбайт. Отже, одночасне виконання цих вимог неможливо. У цій ситуації слід відмовитися від однієї з вимог. Можна або

застосувати іншу мову програмування, або збільшити обсяг пам'яті, що виділяється для системи.

Розглянемо приклад вимоги предметної області системи автоматичного гальмування поїзда: ця система повинна автоматично зупинити поїзд на червоний сигнал семафора. Дана вимога вказує спосіб обчислення швидкості поїзда при гальмуванні. Тут використана термінологія, застосовувана при розрахунках швидкостей поїзда. Щоб розібратися в ній, необхідні відповідні знання про системи керування поїздами і їх характеристики.

Для чого необхідно детально описувати вимоги? 10 сторінок або одну? Проблема трудомісткості і вартості програмного продукту вирішується саме на цьому етапі розробки ТЗ.



Різниця (different) між очікуванням замовника і баченням розробника повинна прагнути до оптимального значення. На графіку синя парабола – це вартість технічного завдання, а червона – це вартість відхилення (different), величина якого повинна бути не більше 5-10% прєктного бюджету. Коригувальні зміни в будь-якому програмному забезпеченні неминучі. Тому конфлікти функціональних та нефункціональних вимог у процесі створення програмного забезпечення важливо вміти вирішувати на фундаментальному етапі технічного завдання.

### Література

1. Смірнов В.В. *Технологія проектування програмних систем. Лабораторні роботи* / В.В. Смірнов, Н.В. Смірнова. – Кіровоград: КНТУ, 2012. – 53 с.
2. Смірнов В.В. *Технологія проектування програмних систем. Лекції* / В.В. Смірнов, Н.В. Смірнова. – Кіровоград: КНТУ, 2012. – 95 с.