

УДК 303.7

Суха К.С.<sup>1</sup>, Зайко Т.А.<sup>2</sup>

<sup>1</sup> студ. гр. КНТ-114м ЗНТУ

<sup>2</sup> канд. техн. наук, доц. ЗНТУ

## **ПРИНЦИПИ БОРотьБИ З ПРОТИРІЧЧЯМИ В ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЯХ ЗА ДОПОМОГОЮ ТЕХНОЛОГІЙ ТБВЗ**

При створенні програмного забезпечення (ПЗ) розробники стикаються зустрічаються з безліччю проблем, завдань та протиріч, які необхідно швидко і ефективно вирішити. На кожному етапі життєвого циклу програмних продуктів необхідна якісна робота програмістів, що беруть участь в створенні цих продуктів. Не завжди виходить знайти швидке рішення проблем розробки ПЗ, складності перенесення та непомічені помилки досить сильно уповільнюють прогрес, в свою чергу це породжує відставання від графіка, перевищення кошторису тощо.

Одним із способів вдосконалення процесів створення та розвитку програмних продуктів можуть бути методи, розвинені в теорії вирішення винахідницьких задач (ТБВЗ). Технології ТБВЗ створено для знаходження ідеального кінцевого результату при виникненні технічних протиріч, коли спроба поліпшити одну характеристику технічної системи провокує погіршення іншої. Для програмних продуктів інструменти ТБВЗ також

можуть бути застосовані, тому що вони є технічними системами передачі й обробки інформації [1].

На думку Генріха Альтшулера, основоположника ТБВЗ, можливо було виявити з досвіду попередників часто повторювані прийоми успішних винаходів. Для цього було проведено аналіз близько сорока тисяч авторських свідоцтв та патентів, і за допомогою виявлених закономірностей розвитку технічних систем і прийомів винахідництва розробили ТБВЗ. Згодом Альтшулер довів, що в сутності більшості винахідницьких задач лежить відносно мала кількість протиріч між універсалізованими характеристиками технічних систем. Тобто існують типові технічні протиріччя, такі як «вага – міцність», «точність – продуктивність» [2].

Вивчивши багато винаходів вищого класу, Альтшулер зауважив, що протиріччя в таких завданнях усувалися особливими прийомами – способами зміни вихідної системи. На основі отриманих результатів Г. С. Альтшулер склав таблицю застосування цих прийомів, що складається з сорока прийомів, які залежать від типу протиріч, в якій по вертикалі розташовуються характеристики технічних систем, які за умовами завдання необхідно поліпшити, а по горизонталі – характеристики, які при цьому неприпустимо погіршуються. На перетині граф таблиці вказані номери прийомів, які з найбільшою ймовірністю можуть усунути виникле технічне протиріччя [3]. І хоча адаптація таблиці протиріч вимагає і уточнення формулювань принципів таким чином, щоб вони легко сприймалися програмістами, це зовсім не заважає її активному використанню вже сьогодні. Так, «принцип посередника» для використання у сфері розробки ПЗ слід переформулювати як «принцип віртуалізації», «принцип дроблення» – як «принцип підвищення модульності», а «принцип винесення» – як «принцип виділення функціональності». Розглянемо декілька технічних протиріч, які виникають у сфері розробки ПЗ і відшукаємо способи їх усунення, використовуючи таблицю протиріч.

Одне з типових протиріч, при використанні в якості мови розробки JavaScript, можна визначити наступним чином: ЯКЩО ізолювати контекст виконання додатків, ТО можна істотно підвищити безпеку корпоративних середовищ, АЛЕ при цьому виникнуть втрати в продуктивності. Згідно з таблицею застосування прийомів вирішення протиріч визначаємо відповідну пару: зміна (30, «Шкідливі фактори, що діють на об'єкт») – погіршення (39, Продуктивність). З набору прийомів, які пропонуються в якості варіантів рішення (22, 35, 13, 24) може бути обрано 24, «Принцип посередника», згідно з яким, слід на час приєднати до об'єкту інший (легко видаляємий) об'єкт. У даному випадку таким легко видаляємим об'єктом буде слугувати комплекс апаратних засобів підтримки віртуалізації, впровадження якого призведе до усунення протиріччя.

Інше протиріччя, що виникає при тестуванні інструменту розробки веб-інтерфейсу, полягає в тому, що ЯКЦО тестована програма знає структуру веб-сторінки, ТО можна виконати перевірку, АЛЕ при цьому тестування стає залежним від веб-інтерфейсу, який постійно змінюється. Згідно з таблицею застосування прийомів вирішення протиріч можна визначити відповідну пару: зміна (35, Адаптація, універсальність) – погіршення (30, Шкідливі фактори, що діють на об'єкт). З набору прийомів (35, 11, 32, 31) можна обрати прийом 11, «Принцип заздалегідь підкладеної подушки», реалізація якого буде полягати в використанні заздалегідь ідентифікаційних полів для різних елементів сторінки.

Таким чином, можна зробити висновок, що використання принципів ТВВЗ для усунення технічних протиріч у сфері розробки ПЗ є ефективним та активно запровадженим рішенням.

### **СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

1. Підвищення ефективності розробки програмних продуктів на основі методів ТВВЗ [Текст]: матеріали наук.-метод. конф. «ТВВЗ-фест 2009», Санкт-Петербург, 27-29 липня 2009 р.: збірник праць / [редкол.: А.В. Єфімов (відпов. ред.) та ін.]. – СПб.: СПбПУ, 2009. – 302 с.
2. Одинцов І.О. Професійне програмування. Системний підхід [Текст] / І.О. Одинцов ; 2-е вид. перероб. і доп. – СПб.: БХВ, 2004. – 624 с.
3. Альтшулер Г.С. Алгоритм винаходу [Текст] / Г.С. Альтшулер ; 2-е вид. перероб. і доп. – М.: Московський робочий, 1973. – 296 с.