

Вінницький національний технічний університет  
Міністерство освіти і науки України

Кваліфікаційна наукова  
праця на правах рукопису

**ПРИЙМАК НАТАЛІЯ ВАСИЛІВНА**

УДК 004.8:004.415

**ДИСЕРТАЦІЯ**

**ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПОШУКУ АСОЦІАТИВНИХ  
ПРАВИЛ ЩОДО ТРИВАЛОСТІ РОЗРОБКИ ПРОГРАМНОГО  
ЗАБЕЗПЕЧЕННЯ**

05.13.06 – інформаційні технології  
Технічні науки

Подається на здобуття наукового ступеня кандидата технічних наук

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

\_\_\_\_\_ Н. В. Приймак

Науковий керівник Савчук Тамара Олександрівна  
к.т.н., професор

Вінниця - 2020

## АНОТАЦІЯ

*Приймак Н.В.* Інформаційна технологія пошуку асоціативних правил щодо тривалості розробки програмного забезпечення. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.06 «Інформаційні технології». – Вінницький національний технічний університет, Вінниця, 2020.

Дисертаційна робота присвячена підвищенню точності визначення тривалості виконання завдання розробником, а також скороченню тривалості пошуку асоціативних правил (АП) щодо тривалості розробки програмного забезпечення (ПЗ), які для цього використовуються. Це досягається за рахунок використання відповідної інформаційної технології (ІТ). Дана ІТ базується на удосконаленому методі визначення частих предметних наборів Frequent Pattern Growth (FP-Growth), серед яких здійснюється пошук таких правил. Об'єктом дослідження є процес пошуку асоціативних правил щодо тривалості розробки програмного забезпечення. Загальний науковий результат роботи – вирішення актуальної науково-практичної задачі підвищення точності визначення тривалості виконання завдання розробником, а також скорочення тривалості пошуку асоціативних правил (АП) щодо тривалості розробки програмного забезпечення (ПЗ), які для цього використовуються. Такий результат досягається за допомогою використання запропонованої інформаційної технології пошуку таких асоціативних правил.

Оскільки програмне забезпечення відіграє важливу роль в сучасному житті людини, тому й процес його розробки та засоби, необхідні для цього, постійно удосконалюються з метою покращення даного процесу.

**В першому розділі** дисертаційного дослідження було здійснено аналіз сучасного стану розвитку процесу розробки програмного забезпечення.

Виконано порівняльний аналіз засобів, що використовуються для управління процесом розробки програмного забезпечення, а також розглянуто моделі процесу розробки програмного забезпечення. Зроблено висновок, що відповідні засоби, не дозволяють спрогнозувати тривалість виконання завдання, базуючись на накопичених даних. Також встановлено, що при розробці програмного забезпечення, з використанням розглянутих моделей, враховується лише пріоритет завдання, а тому існує необхідність у створенні інформаційної моделі, що також враховує такі характеристики, як складність, критичність та тривалість виконання завдання.

Було досліджено методи інтелектуального аналізу даних, які використовуються щодо розробки програмного забезпечення, та доведено доцільність їх застосування під час такого процесу. Також проведено дослідження методів визначення частих предметних наборів, які використовуються для пошуку асоціативних правил щодо тривалості розробки програмного забезпечення. Встановлено, що FP-Growth є методом, який доцільно застосовувати для пошуку АП щодо тривалості розробки ПЗ. Даний метод потрібно удосконалити з метою підвищення точності визначення тривалості виконання завдання розробником, а також скорочення тривалості пошуку асоціативних правил, які для цього використовуються; а також його адаптації до відповідної предметної області. Удосконалення полягає у класифікації завдань залежно від їх характеристик (складність, критичність, пріоритет, тривалість виконання завдання) перед пошуком асоціативних правил, що дозволить пришвидшити процес пошуку таких правил.

Здійснено обґрунтування доцільності пошуку і використання асоціативних правил щодо тривалості розробки програмного забезпечення. Підтверджено актуальність наукової та практичної задачі розробки інформаційної технології пошуку асоціативних правил щодо тривалості розробки програмного забезпечення, що базується на удосконаленому

методі визначення частих предметних наборів для пошуку АП FP-Growth.

У другому розділі запропоновано інформаційну модель процесу пошуку асоціативних правил щодо тривалості розробки програмного забезпечення, яку можна подати у вигляді:

$$AR = \langle T, D, \text{minsupp}, \text{minconf}, \mu, \Omega, NT, PT \rangle,$$

де  $T$  – множина завдань, що були виконані під час розробки програмного забезпечення;  $D$  – множина розробників, які можуть виконати поставлене завдання;  $\text{minsupp}$  – мінімальне значення підтримки асоціативних правил;  $\text{minconf}$  – мінімальне значення достовірності асоціативних правил;  $\mu$  – метод, за допомогою якого буде здійснюватися пошук асоціативних правил;  $\Omega$  – множина знайдених асоціативних правил;  $NT$  – множина завдань, для яких необхідно визначити тривалість їх реалізації розробником певного рівня кваліфікації;  $PT$  – множина завдань з визначеним часом, який необхідний на їх виконання розробником, що маю певну кваліфікацію.

Вхідними даними для інформаційної моделі є множина завдань, серед яких буде здійснюватися пошук асоціативних правил з використанням удосконаленого методу визначення частих предметних наборів FP-Growth; вихідними даними є множина завдань з часом, необхідним розробникам для реалізації таких завдань.

Удосконалено метод визначення частих предметних наборів FP-Growth, що використовуються для пошуку асоціативних правил, за рахунок класифікації вхідної множини завдань на три класи, відповідно до характеристики завдань: завдання категорії 1 (завдання, які легко реалізувати), завдання категорії 3 (завдання, які важко реалізувати), завдання категорії 2 (завдання, по складності між легкими і важкими). Таке удосконалення підвищило точність визначення часу, необхідного на виконання завдання з використанням знайдених асоціативних правил, а також скорочення тривалості пошуку таких залежностей.

Для визначення часу, необхідного на реалізацію завдання розробником певної кваліфікації, розроблено алгоритм використання знайдених асоціативних правил щодо тривалості розробки програмного забезпечення. Це допоможе менеджерам проектів при плануванні та управлінні процесом розробки програмного забезпечення. Наведено UML-діаграму діяльності процесу використання асоціативних правил, знайдених щодо тривалості розробки програмного забезпечення.

**У третьому розділі** дисертаційного дослідження було розроблено інформаційну технологію пошуку асоціативних правил щодо тривалості розробки програмного забезпечення, що базується на запропонованій інформаційній моделі даного процесу.

Здійснено відображення процесу розробки програмного забезпечення за допомогою марковських ланцюгів з дискретними станами та дискретним часом. Таке відображення дозволило, з використанням єдиного математичного апарату, відобразити даний процес на різних рівнях його деталізації.

Також здійснено моделювання процесу пошуку асоціативних правил щодо тривалості розробки програмного забезпечення з використанням удосконаленого методу визначення частих предметних наборів FP-Growth. Результати моделювання показали зростання швидкості пошуку АП до 2%, якщо кількість завдань для аналізу перевищує 750. Також було встановлено, що точність визначення тривалості виконання завдань залежить від кількості знайдених асоціативних правил: точність зросла у 2,3 рази при збільшенні кількості асоціативних правил у 11 разів. Результати моделювання процесу використання асоціативних правил з метою визначення тривалості виконання завдання розробником певної кваліфікації показали зростання швидкості визначення часу, необхідного на виконання завдання розробником до 5 разів.

**У четвертому розділі** дисертаційного дослідження розроблено інформаційну систему пошуку асоціативних правил щодо тривалості

розробки програмного забезпечення з використанням запропонованої інформаційної технології. Для цього було наведено рекомендації щодо її розробки, запропоновано її структуру та спроектовано БД відповідного програмного забезпечення.

Аналіз результатів впровадження та використання інформаційної технології пошуку асоціативних правил щодо тривалості розробки програмного забезпечення показав підвищення точності визначення тривалості виконання завдання розробником, а також скорочення тривалості пошуку асоціативних правил, які для цього використовуються. А саме: збільшення точності визначення тривалості виконання завдання в 1,7 рази при збільшенні знайдених асоціативних правил до 100; скорочення тривалості пошуку АП на 1,2% якщо кількість записів, серед яких здійснюється пошук АП, від 500 до 750, та на 2%, якщо кількість записів від 750 до 100; зростання швидкості визначення часу необхідного на виконання завдання розробником певної кваліфікації від 2 до 6 разів в залежності від кількості завдань, для яких цей час визначався.

Ключові слова: програмне забезпечення, процес розробки програмного забезпечення, інформаційна технологія, інформаційна модель, асоціативні правила, метод FP-growth, класифікація, дерева рішень, інтелектуальний аналіз даних.

## ABSTRACT

*Pryimak N. V.* The information technology of the associative rules search for the duration of software development – Qualification research paper, manuscript copyright.

Thesis for the degree of a candidate of technical sciences in specialty 05.13.06 «Information technology». – Vinnytsia National Technical University, Vinnytsia, 2020.

The thesis is dedicated to the study of increasing the accuracy of determining the duration of the task development, as well as of reducing the duration of the associative rules search, which are used for this. It can be reached by the development and implementation of relevant information technology. This information technology is based on the advanced FP-Growth method of generating frequent pattern sets, among which, the search of such rules is performed. The object of the research is the process of the associative rules search for the duration of software development. The general scientific result of the work is a solution for the actual scientific and practical task of increasing the accuracy of determining the duration of the task development, as well as of reducing the duration of the associative rules search which are used for this, due to the development and implementation of relevant information technology.

As the software plays a significant role in the modern life, the process of its development and the tools necessary, used for this purpose, are constantly being developed to improve this process.

In **the first section** of the thesis, an analysis of the current state of the software development process has been performed.

A comparative analysis of the tools, used for the software management process was done, and also the models of the software development process was reviewed. It is concluded that the reviewed tools do not allow to predict the duration of the task, based on the accumulated data. It was also found that usage of considered models during the software development, take into account only the priority of the task, and therefore there is a need to create an information model,

which will also take into account such characteristics as the complexity, criticality and the duration of task implementation.

It is established that during software development, usually several models are used, and they have defined stages. Appropriate tools are used to manage this process, but they do not allow to predict the duration of the task implementation, based on the accumulated data.

The methods of the data mining, used during the software development, were investigated, and the feasibility of their use was proved. Also, the methods of generating frequent pattern sets, which are used for associative rules search for the duration of software development, were studied. It was found that FP-Growth is the method that should be used for associative rules search for the duration of software development. This method needs to be improved in order to increase the accuracy of determining the duration of the task development, as well as to reduce the duration of the associative rules search which are used for this, and as well as its adaptation to the relevant subject area. The improvement is based on the data classification before the associative rules search, which will allow to accelerate the process of such rules searching.

The feasibility of searching and using associative rules for the determining the duration of software development is substantiated. The relevance of the scientific and practical task of developing information technology of searching associative rules for the software development duration, that is based on an advanced method of determining frequent pattern sets FP-Growth, is confirmed.

In **the second section** an information model of the associative rules search for the duration of software development is proposed. This model can be submitted as a set:

$$AR = \langle T, D, \text{minsupp}, \text{minconf}, \mu, \Omega, NT, PT \rangle,$$

where  $T$  is the set of tasks performed during the software development;  $D$  is a set of developers, who can solve the task;  $\text{minsupp}$  is the minimum value of the associative rules support;  $\text{minconf}$  is the minimum value of the associative



rules confidence;  $\mu$  is a method which will be used to search the associative rules;  $\Omega$  is the set of the found associative rules;  $NT$  is a set of tasks, for which it is necessary to determine the duration of their implementation by a developer with particular level of proficiency;  $PT$  is a set of tasks with the defined time necessary for their implementation by a developer with particular professional skill.

The inputs for this information model is a set of tasks, among which the associative rules will be searched by using the advanced FP-Growth method of generating frequent pattern sets; the outcome data is a set of tasks with the defined time necessary for developers to solve them.

The method FP-Growth of the generation of the frequent pattern sets used for the associative rules search is improved by classifying the input set of tasks into three classes, depending on their characteristics: tasks of category 1 (tasks that are easy to implement), tasks of category 3, which are difficult to implement, tasks of category 2 (tasks, with the complexity between category 1 and category 3). This improvement helped to increase the accuracy of determining the duration of the task development, as well as to reduce the duration of the search of the associative rules, which are used for this.

To determine the time required for the task implementation by a specific developer, an algorithm of using the found associative rules for the duration of software development was created. It will help project managers with planning and managing the software development process. The UML activity diagram, which describes the process of using such associative rules, was introduced.

In **the third section** of the research, an information technology of the associative rules search for the duration of software development was created. It's based on the proposed information model of this process. The software development process was demonstrated through the use of Markov chains with discrete states and discrete time. Such simulation allows to use a single mathematical apparatus to display this process at various levels of it details.

The simulation of the associative rules searches for the duration of software development by using the advanced FP-Growth method for generating frequent pattern sets has been implemented. The simulation results showed an increase of the speed of the associative rules search to 2%, in case the number of tasks for analysis exceed 750. It was also found that the accuracy of predicting the tasks duration depends on the number of found associative rules: accuracy increased in 2.3 times with an increase of associative rules number in 11 times. The results of modeling the process of using associative rules to determine the duration of the task development by the developer of a certain qualification, showed an increase of speed to determine the time, required to complete the task, up to 5 times.

In **the fourth section** of the thesis the information system for the associative rules search for the duration of software development was developed, using the proposed information technology. To archive this, recommendations for its development were given, its structure was proposed, and a database of the related software was designed.

The analysis of the results of the integration and use of information technology of the associative rules search for the duration of software development showed: an increase of the accuracy of the prediction of the task implementation duration in 1,7 times with the increase of found associative rules to 100; a decrease of the search duration of the associative rules by 1.2% if the number of records, among which they are searched, is between 500 and 750, and by 2% if the number of records is from 750 to 1000; increase in the speed of determining the time required to complete the task by the developer of a certain qualification from 2 to 6 times, depending on the number of tasks for which this time was determined.

Keywords: software, the process of software development, information technology, information model, associative rules, method FP-growth, classification, decision trees, data mining.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	15
ВСТУП.....	16
РОЗДІЛ 1 СУЧАСНІ МОДЕЛІ, МЕТОДИ ТА ЗАСОБИ ПІДТРИМКИ ПРОЦЕСІВ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	23
1.1 Аналіз засобів, що використовуються для управління процесом розробки програмного забезпечення .....	23
1.2 Сучасні моделі розробки програмного забезпечення.....	30
1.3 Особливості використання методів інтелектуального аналізу даних щодо розробки програмного забезпечення.....	35
1.4 Обґрунтування доцільності пошуку і використання асоціативних правил щодо тривалості розробки програмного забезпечення.....	40
1.5 Постановка задачі.....	54
1.6 Висновки до розділу 1.....	55
РОЗДІЛ 2 РОЗРОБКА ТЕОРЕТИКО-МЕТОДИЧНИХ ЗАСАД ПРОЦЕСУ ПОШУКУ АСОЦІАТИВНИХ ПРАВИЛ ЩОДО ТРИВАЛОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	57
2.1 Інформаційна модель процесу пошуку асоціативних правил щодо тривалості розробки програмного забезпечення .....	57
2.2 Удосконалений метод FP-Growth визначення частих предметних наборів для пошуку асоціативних правил .....	66
2.3 Алгоритм використання асоціативних правил для визначення часу, необхідного для виконання завдання розробником.....	73
2.4 Висновки до розділу 2.....	78
РОЗДІЛ 3 ПОШУК АСОЦІАТИВНИХ ПРАВИЛ ЩОДО ТРИВАЛОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВИКОРИСТАННЯМ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ.....	80
3.1 Інформаційна технологія пошуку асоціативних правил щодо тривалості розробки програмного забезпечення .....	80
3.2 Моделювання процесу розробки програмного забезпечення.....	84
3.3 Моделювання процесу пошуку асоціативних правил щодо тривалості розробки програмного забезпечення з використанням удосконаленого методу FP-Growth .....	93

	12
3.4 Висновки до розділу 3.....	101
РОЗДІЛ 4 ІНФОРМАЦІЙНА СИСТЕМА ПОШУКУ АСОЦІАТИВНИХ ПРАВИЛ ЩОДО ТРИВАЛОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	103
4.1 Рекомендації щодо розробки інформаційної системи пошуку асоціативних правил щодо тривалості розробки програмного забезпечення.....	103
4.1.1 Структура інформаційної системи.....	103
4.1.2 База даних програмного забезпечення пошуку асоціативних правил щодо тривалості розробки .....	112
4.1.3 Графічний інтерфейс користувача інформаційної системи пошуку асоціативних правил.....	115
4.2 Результати впровадження інформаційної технології пошуку асоціативних правил щодо тривалості розробки програмного забезпечення .....	118
4.3 Висновки до розділу 4.....	123
ВИСНОВКИ.....	125
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	128
ДОДАТКИ.....	143
Додаток А Акти впровадження результатів дисертації.....	144
Додаток Б Список публікацій за темою дисертації.....	146
Додаток В Ймовірності переходів між станами для трьох рівнів деталізації процесу розробки програмного забезпечення.....	148
Додаток Г Лістинг програмної реалізації основних модулів програмного забезпечення пошуку асоціативних правил щодо тривалості розробки програмного забезпечення.....	149
Додаток Д Інструкція користувача інформаційної системи пошуку асоціативних правил щодо тривалості розробки програмного забезпечення.....	156
Додаток Е Вибірка завдань для пошуку асоціативних правил.....	162

## ВСТУП

### Обґрунтування вибору теми дослідження

У сучасному світі програмне забезпечення (ПЗ) використовується для роботи, навчання, спілкування; для здійснення покупок, платежів, подорожей тощо. Програмне забезпечення має важливе значення для різних галузей бізнесу та виробництва: використовується в банках та страхових компаніях; в енергетиці та сільському господарстві; у торгівлі, медицині, навчанні тощо.

Оскільки програмне забезпечення стає все більш важливим у всіх галузях промисловості та послуг, то процес його розробки та інженерні засоби, необхідні для цього, перебувають у постійному розвитку [1]. Так як програмні продукти різноманітні та мають різний контекст використання: застосування прикладного програмного забезпечення в інформаційній системі чи програмного забезпечення, вбудованого в інші продукти, або ігрового програмного забезпечення тощо, то виникає необхідність оцінки відповідності системи, компонента чи процесу визначеними вимогам: функціональним чи не функціональним [2].

Однією з нефункціональних вимог до застосування програмного забезпечення, згідно зі стандартом IEEE Computer 1985, є вимога до його життєвого циклу: визначення обмежень щодо задіяних людських ресурсів та тривалості процесу розробки такого ПЗ [3], [4]. Точна оцінка графіка закінчення розробки програмного забезпечення необхідна менеджерам для планування роботи команди розробників та ефективного управління даним процесом. Окрім вказаної вимоги важливим показником є організація процесу розробки та якість кінцевого програмного забезпечення.

Одним із способів перевірки, що програмне забезпечення надійне – є контроль усіх етапів його розробки. Традиційна модель розробки програмного забезпечення передбачає перевірку кожного етапу розробки (вимоги до документації, проектування, аналіз, кодування, перевірка та тестування) перед переходом до наступного [5]. Сучасні програмні проекти

розробляються більш динамічно, із застосуванням більш гнучких технологій та методологій розробки; але етапи, перераховані вище, присутні в будь-якому процесі, оскільки усе програмне забезпечення повинно бути розроблене, інтегроване, перевірене, випущене та впроваджене [6].

У зв'язку з тим, що складність і різноманітність програм постійно зростає, то їх якість має важливе значення для споживача. В процесі написання програмного забезпечення його код часто змінюється з метою поліпшення функціональності або для виправлення помилок, які викликають збій у його роботі. Використання систем управління версіями програмного коду та баз даних (БД) завдань, що ставляться під час розробки ПЗ, дозволяють зібрати інформацію про усі зміни програмного коду та використати її для подальшого аналізу [7]. Наявні бази даних містять значну кількість інформації, яку неможливо проаналізувати вручну, а, отже, виникає необхідність у автоматизованому аналізі. Для вирішення даної задачі використовують різноманітні методи інтелектуального аналізу даних (ІАД) до яких належить і пошук асоціативних правил (АП). Такий метод дозволяє знаходити асоціативні залежності, які можна використати для визначення часу, необхідного для реалізації конкретного завдання розробником. Отримана інформація може бути застосована менеджерами проектів для планування розробки ПЗ.

Сучасні методи пошуку асоціативних правил щодо тривалості розробки програмного забезпечення розглянуті в роботах Г. П'ятецький-Шапіро [8], В. Жоу [9], Ф. Жанг [10], Х. Жанг [11], П. Бхатачар'я [12], А. Ламканхфі [13], Е. Гігер [14]. Питання моделювання систем представлено у роботах А. А. Молчанова [15], В.І. Скуріхіна [16], Н. Бусленко [17], В. М. Трояновського [18], Р. Н. Кветного [19], В. М. Дубового [20], Б. І. Мокіна [21], В. Б. Мокіна [22], О. В. Бісікала [23] та інших дослідників. Один із підходів до вирішення задачі прогнозування трудомісткості розробки програмних систем запропоновано у роботі С. Д. Штовби [24].

Отже, наукова задача створення інформаційної технології (ІТ) пошуку

асоціативних правил щодо тривалості розробки програмного забезпечення є актуальною, оскільки використання такої інформаційної технології дозволить підвищити швидкість важливого процесу розробки програмного забезпечення.

### **Зв'язок роботи з науковими програмами, планами, темами**

Дисертаційна робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій».

Результати дисертаційного дослідження увійшли у звіт з держбюджетної науково-дослідної роботи на тему «Розробка інформаційної системи пошуку асоціативних правил щодо тривалості розробки програмного забезпечення» № 47/5В (номер договору про творчу співпрацю №47/5 від 15 січня 2018 року), де здобувач був у складі авторів проміжних та заключного звітів. Також впроваджено інформаційну технологію пошуку асоціативних правил щодо тривалості розробки програмного забезпечення в ТОВ «Бізнес-Програми». Відповідні акти впровадження наведено в додатку А.

### **Мета і завдання дослідження**

*Метою* дисертаційного дослідження є підвищення точності визначення тривалості виконання завдання розробником, а також скорочення тривалості пошуку асоціативних правил, які для цього застосовуються, за рахунок використання відповідної інформаційної технології.

Для досягнення поставленої мети необхідно розв'язати такі *завдання*:

1. Дослідити сучасний стан розвитку процесу розробки програмного забезпечення та проаналізувати існуючі методи визначення частих предметних наборів для пошуку асоціативних правил щодо тривалості даного процесу. Розробити інформаційну модель (ІМ) процесу пошуку асоціативних правил щодо тривалості розробки програмного забезпечення.

2. Удосконалити метод визначення частих предметних наборів FP-

Growth для пошуку асоціативних правил щодо тривалості розробки програмного забезпечення.

3. Розробити інформаційну технологію, яка проводитиме пошук асоціативних правил щодо тривалості розробки програмного забезпечення з метою їхнього подальшого використання при розробці програмного забезпечення.

4. Провести моделювання процесу пошуку асоціативних правил з використанням удосконаленого методу FP-Growth.

5. Розробити інформаційну систему пошуку асоціативних правил щодо тривалості розробки програмного забезпечення, що базується на запропонованій інформаційній технології.

*Об'єктом дослідження* є процес пошуку асоціативних правил щодо тривалості розробки програмного забезпечення.

*Предметом дослідження* є інформаційна технологія пошуку асоціативних правил щодо тривалості розробки програмного забезпечення.

### **Методи дослідження**

Використано теоретичні та змішані методи, серед них: методи системного аналізу (порівняльний та структурний аналіз) для визначення властивостей процесу пошуку асоціативних правил щодо тривалості розробки програмного забезпечення; методи інтелектуального аналізу даних (пошук асоціативних правил та класифікацію даних) для розробки інформаційної моделі вищевказаного процесу та удосконаленого методу визначення частих предметних наборів для пошуку асоціативних правил, що базується на цій моделі; методи об'єктного моделювання процесу пошуку асоціативних правил, а також розробки програмного забезпечення; комп'ютерне моделювання для розробки програмного забезпечення пошуку асоціативних правил.

### **Наукова новизна отриманих результатів**

1. Уперше запропоновано інформаційну модель процесу пошуку асоціативних правил щодо тривалості розробки програмного забезпечення,



яка, на відміну від існуючих, враховує такі характеристики завдань (завдань, які потрібно вирішити при розробці програмного забезпечення), як: складність, пріоритет, критичність та тривалість їх виконання. Це дозволило підвищити точність визначення часу, необхідного на виконання завдання розробником певної кваліфікації.

2. Удосконалено метод визначення частих предметних наборів для пошуку асоціативних правил FP-Growth щодо тривалості розробки програмного забезпечення за рахунок введення етапу класифікації завдань відповідно до таких характеристик завдань: складність, пріоритет, критичність та тривалість їх виконання. Це дозволило скоротити час, необхідний для пошуку асоціативних правил щодо тривалості розробки програмного забезпечення, які в подальшому використовуються для визначення часових характеристик реалізації завдання розробником певної кваліфікації.

3. Уперше розроблено інформаційну технологію пошуку асоціативних правил щодо тривалості розробки програмного забезпечення в основі якої, на відміну від існуючих, використовується інформаційна модель та удосконалений метод визначення частих предметних наборів для пошуку асоціативних правил FP-Growth, що дозволило збільшити точність визначення тривалості виконання завдання розробником певної кваліфікації та скоротити тривалість пошуку асоціативних правил щодо тривалості розробки програмного забезпечення.

### **Практичне значення отриманих результатів**

Практична цінність результатів дисертаційного дослідження полягає в тому, що:

– розроблено алгоритм, що реалізує удосконалений метод визначення частих предметних наборів для пошуку асоціативних правил щодо тривалості розробки програмного забезпечення за рахунок введення етапу класифікації завдань відповідно до їх характеристик (складність, пріоритет, критичність та тривалість їх виконання), що дозволило скоротити тривалість пошуку таких

асоціативних правил;

– розроблено алгоритм використання знайдених асоціативних правил щодо тривалості розробки програмного забезпечення, що дозволило автоматизувати процес розробки програмного забезпечення та підвищити точність визначення тривалості виконання завдання розробником певної кваліфікації;

– розроблено інформаційну систему для пошуку асоціативних правил щодо тривалості розробки програмного забезпечення, що реалізує розроблену інформаційну технологію.

Отримані на основі наукових досліджень практичні результати впроваджено в ТОВ «Бізнес-Програми» у вигляді інформаційної системи пошуку асоціативних правил при розробці програмного забезпечення (акт впровадження № 47/5В від 17 вересня 2018 року), а також інформаційної технології пошуку асоціативних правил щодо тривалості розробки програмного забезпечення (акт впровадження від 27 липня 2018 року). Це дозволило приймати ефективні управлінські рішення щодо планування та управління розробкою програмного забезпечення. Впровадження результатів дослідження підтверджено відповідними актами в додатку А.

### **Особистий внесок здобувача**

Основні теоретичні та експериментальні дослідження та висновки дисертаційної роботи були отримані автором особисто.

У друкованих працях, опублікованих у співавторстві, автору дисертації належать: аналіз доцільності використання асоціативних правил щодо тривалості розробки програмного забезпечення [25], [26], аналіз доцільності використання методів ІАД при тестуванні програмного забезпечення [27] – [29], аналіз доцільності використання методу FP-Growth щодо тривалості розробки програмного забезпечення [30], [31], розробка інформаційної моделі процесу пошуку асоціативних правил щодо тривалості розробки програмного забезпечення [32], [33], удосконалення методу пошуку асоціативних правил FP-Growth, за рахунок класифікації даних [34],

представлення процесу розробки програмного забезпечення з використанням марковських процесів [35], [36], розробка інформаційної технології процесу пошуку асоціативних правил щодо тривалості розробки програмного забезпечення [37], [38], комп'ютерне моделювання [39].

### **Апробація матеріалів дисертації**

Результати проведеного дослідження були апробовані на 10-ій та 11-ій міжнародних науково-практичних конференціях «Інтернет – Освіта – Наука» (2016, 2018), XLV, XLVI, XLVIII науково-технічних конференціях професорсько-викладацького складу, співробітників та студентів ВНТУ (2016, 2017, 2019), IX міжнародній науково-практичній конференції Грузинської математичної асоціації, Тбілісі, Грузія (2018), Всеукраїнській науково-практичній конференції здобувачів вищої освіти й молодих учених «Комп'ютерна інженерія і кібербезпека: досягнення та інновації», Кропивницький (2018).

### **Публікації**

Матеріали досліджень опубліковані в 15 наукових працях: 1 стаття, що опублікована в іноземному фаховому журналі, який входить до наукометричної бази SCOPUS, 4 статті, що опубліковані в наукових виданнях з переліку фахових видань України, з них, в тому числі, 1 стаття у журналі, який входить до наукометричної бази SCOPUS; результати дослідження апробовано та опубліковано у вигляді 9 тез на шістьох науково-технічних конференціях (4 публікації у збірниках праць, 1 публікація у збірнику матеріалів конференції, 3 – у вигляді тез доповідей) та одній науково-практичній конференції; отримано 1 свідоцтво на реєстрацію авторського права на комп'ютерну програму (додаток Б).

### **Структура та обсяг дисертації**

Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Основний зміст роботи викладено на 108 сторінках друкованого тексту, містить 34 рисунки та 16 таблиць. Список використаних джерел містить 161 найменування. Загальний обсяг роботи становить 164 сторінки.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] S. Konrad, “Requirements engineering in the development of large-scale systems” in *16th IEEE International Requirements Engineering Conference*, Barcelona, 2008, pp. 217 – 222.
- [2] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, NY, USA: McGraw-Hill, 2014.
- [3] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*, NY, USA: Springer US, 2000.
- [4] K. Wiegers, *Software Requirements 3rd Edition*, NY, USA: Microsoft Press, 2013.
- [5] S. J. Greenspan, and C. L. McGowan, “Structuring software development for reliability”, *Microelectronics Reliability*, vol. 17, no. 1, pp. 75 – 83, 1987.
- [6] G. Metkalfе, *Lean Software Development: Avoiding project mishaps: An introduction*, Michigan, USA: Independently published, 2018.
- [7] G. E. Kaiser, P. H. Feiler, and S. S. Popovich, “Intelligent assistance for software development and maintenance”, *IEEE Software*, vol. 5, no. 3, pp. 40 – 49, 1988.
- [8] G. Piatetsky-Shapiro, and W. Frawley, *Knowledge Discovery in Databases*, CA, USA: MIT Press, 1991.
- [9] W. Zou, X. Xia, W. Zhang, Z. Chen, “An Empirical Study of Bug Fixing Rate”, in *IEEE 39th Annual Computer Software and Applications Conference*, Taichung, 2015, pp. 125 – 136.
- [10] F. Zhang, F. Khomh, Y. Zou, and A. Hassan, “An Empirical Study on Factors Impacting Bug Fixing Time”, in *19th Working Conference on Reverse Engineering*, Washington, 2012, pp. 225 – 234.
- [11] H. Zhang, L. Gong, and S. Versteeg, “Predicting Bug-Fixing Time: An Empirical Study of Commercial Software Projects”, in *35th International Conference on Software Engineering*, San Francisco, 2013, pp. 1042 – 1051.

- [12] P. Bhattacharya, and I. Neamtiu, “Bug-fix Time Prediction Models: Can We Do Better?” *MSR '11 Proceedings of the 8th Working Conference on Mining Software Repositories*, Honolulu, 2011, pp. 207 – 210.
- [13] A. Lamkanf, and S. Demeyer, “Filtering Bug Reports for Fix-Time Analysis”, *16th European Conference on Software Maintenance and Reengineering*, Szeged, 2012, pp. 379 – 384.
- [14] E. Giger, M. Pinzger, and H. Gall, “Predicting the Fix Time of Bugs”, *RSSE '10: Proceedings of the 2nd International Workshop on Recommendation System for Software Engineering*, NY, 2010, pp. 52 – 56.
- [15] А. А. Молчанов, *Моделирование и проектирование сложных систем*. Київ, Україна: Вища школа, 1988.
- [16] В. И. Скурихин и др. *Математическое моделирование*. Київ, Україна: Техніка, 1983.
- [17] Н. П. Бусленко. *Моделирование сложных систем*. Москва, Россия: Наука, 1978.
- [18] В. М. Трояновский. *Математическое моделирование в менеджменте*. Москва, Россия: РДЛ, 2002.
- [19] Р. Н. Кветний, І. В. Богач, О. Р. Бойко, та О. Ю. Софіна, *Комп'ютерне моделювання систем та процесів. Методи обчислень*. Вінниця, Україна: ВНТУ, 2012.
- [20] В. М. Дубовой, *Моделювання систем контролю та керування. Навчальний посібник*. Вінниця, Україна: ВНТУ, 2005.
- [21] Б. І. Мокін, та А. В. Камінський, *Математичне та комп'ютерне моделювання процесів оптимізації центрування електричних мереж*. Вінниця, Україна: УНІВЕРСУМ, 2005.
- [22] В. Б. Мокін, та Ю. С. Гавриков, “Новий підхід до двовимірного моделювання динаміки змішування річкових вод зі стічними, що надходять з берега”, *Вісник Херсонського національного технічного університету*, № 2, с. 340 – 343, 2002.

[23] О. В. Бісікало, “Класифікація образного пошуку та моделювання інсайту”, *Вісник Сумського державного університету. Серія Технічні науки*. №2, с.53 – 59, 2008.

[24] Д. С. Штобва, та А. А. Яковенко, “Прогнозування трудомісткості розробки програмних систем за допомогою нечіткої гібридної моделі” *Наукові праці ВНТУ. Серія Інформаційні технології та компютерна техніка*, №1, с. 1 – 7, 2014.

[25] T. O. Savchuk, and N. V. Pryimak, “Feasibility study of using associative rules during the software development”, on *IX international Conference of the Georgian Mathematical Union*, Batumi, 2018, pp. 203 – 204.

[26] Т. О. Савчук, та Н. В. Приймак, “Використання fpg-алгоритму для пошуку асоціативних правил при прийнятті рішень в управлінні процесами”, на *XLV НТК проф.-викл. складу, співробітників та студентів університету*, Вінниця, 2016, с. 929 – 930.

[27] T. O. Savchuk, N. V. Pryimak, A. Assembay, T. Zyska, M. Junisbekov, and A. Annabaev “The technology of searching the associative rules while developing the software”, *Proc. SPIE 10445, Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments*, 2017, doi: 10.1117/12.2280900.

[28] Т. О. Савчук, та Н. В. Приймак, “Використання методів інтелектуального аналізу даних при тестуванні програмного забезпечення”, на *X міжнародній конференції «Інтернет-освіта-наука-2016»*, Вінниця, 2016, с. 28 – 29.

[29] Т. О. Савчук, та Н. В. Приймак, “Використання методів штучного інтелекту для прогнозування часу для виправлення дефектів розробки програмного забезпечення”, на *X міжнародній конференції «Інтернет-освіта-наука-2016»*, Вінниця, 2016, с. 30 – 31.

[30] Т. О. Савчук, та Н. В. Приймак, “Обґрунтування доцільності використання методу fr-growth (fpg) при розробці програмного

забезпечення”, *Вчені записки ТНУ імені В.І. Вернадського. Серія Технічні науки*, № 5, с. 41 – 45, 2018.

[31] Т. О. Савчук, та Н. В. Приймак, “Обґрунтування вибору методу визначення частих предметних наборів для пошуку асоціативних правил при розробці програмного забезпечення”, на *XI міжнародній конференції «Інтернет-освіта-наука-2018»*, Вінниця, 2018, с. 45 – 46.

[32] Т. О. Савчук, та Н. В. Приймак, “Інформаційна модель процесу пошук і використання асоціативних правил при розробці програмного забезпечення”, на *XI міжнародній конференції «Інтернет-освіта-наука-2018»*, Вінниця, 2018, с. 43 – 44.

[33] Т. О. Савчук, та Н. В. Приймак, “Розробка інформаційної моделі процесу пошуку асоціативних правил при розробці програмного забезпечення”, *Інформаційні технології та комп’ютерна інженерія*, № 2 (42), с. 43 – 48, 2018.

[34] Т. О. Савчук, та Н. В. Приймак, “Метод пошуку асоціативних правил при розробці програмного забезпечення, що базується на попередньому розбитті множини завдань” на *XLVIII НТК проф.-викл. складу, співробітників та студентів університету*, Вінниця, 2019, с. 655 – 656.

[35] T. O. Savchuk, and N. V. Pryimak, “Modeling of software development process with the Markov processes”, *Eastern-European journal of Enterprise Technologies . Information technology*, vol. 3, no. 2, pp. 33 – 38, 2017.

[36] Т. О. Савчук, та Н. В. Приймак, “Використання Марковських ланцюгів при моделюванні процесу розробки програмного забезпечення”, на *XLVI НТК проф.-викл. складу, співробітників та студентів університету*, Вінниця, 2017, с. 1019 – 1020.

[37] Т. О. Савчук, та Н. В. Приймак, “Розробка інформаційної технології процесу пошуку асоціативних правил при розробці програмного забезпечення”, *Наукові праці: наук. журн. / Чорном. нац. ун-т ім. П. Могили. Комп’ютерні технології*, вип. 305, т. 317, с. 6 – 11, 2018.

[38] Т. О. Савчук, та Н. В. Приймак, “Інформаційна технологія пошуку асоціативних правил при розробці програмного забезпечення”, на *Всеукраїнській наук.-практ. конф. здобувачів вищої освіти й молодих учених*, Кропивницький, 2018, с. 100 – 102.

[39] Т. О. Савчук, та Н. В. Приймак, “Інформаційна система пошуку асоціативних правил при розробці програмного забезпечення”, *Свідомство про реєстрацію авторського права на твір №82173*, Жов. 11, 2018.

[40] В. К. Батоврин, *Толковый словарь по системной и программной инженерии*. Москва, Россия: ДМК Пресс, 2012.

[41] IEEE Std 829 – 2008 IEEE Standard for Software and System Test Documentation.

[42] A. Stellman, and J. Greene, *Applied Software Project Management*. California, USA: O'Reilly Media, 2005.

[43] M. El Bajta, A. Idri, J. N. Ros, and J. L. Fernández-Alemán, “Software project management approaches for global software development: a systematic mapping study”, *Tsinghua Science and Technology*, vol. 23, no. 6, pp. 690 – 714, 2018.

[44] Ch. Chapman, and St. Warn, *Project risk management: processes, techniques and insights*. Chichester, UK: John Wiley, 1996.

[45] D. White, and J. Fortune, “Current practice in project management – an empirical study”, *International Journal of Project Management*, vol. 20, no. 1, pp. 1 – 11, 2002.

[46] Сетевые методы планирования и управления, 2019. [Электронный ресурс]. Доступно: <http://arprime.ru/planirovanie/setevoye-planirovaniye-i-upravleniye>.

[47] A. Wren, *Project Management A-Z: A Compendium of Project Management Techniques and How to Use Them*, London, UK: Routledge, 2003.

[48] В. Д. Секерин, Р. М. Нижегородцева, А. Е. Горохова, и Д. В. Секерин. *Инновации в маркетинге*. Москва, Россия: МГУИЭ, 2013.



- [49] Поиск критического пути модели, 2019. [Электронный ресурс].  
Доступно: <http://projectimo.ru/planirovanie-proekta/kriticheskij-put.html>.
- [50] T. J. Havranek, *Modern Project Management Techniques for the Environmental Remediation Industry*, NY, USA: Routledge, 2017.
- [51] F. Harrison, and D. Lock, *Advanced Project Management A Structured Approach*, London, UK: Routledge, 2017.
- [52] 6 Benefits of Network Diagrams, 2019. [Online]. Available:  
<https://blog.masterofproject.com/network-diagrams/>.
- [53] Как диаграммы Ганта упрощают работу с проектами, 2019.  
[Электронный ресурс]. Доступно: <https://habr.com/ru/company/blog/415271/>.
- [54] Microsoft Project, 2019. [Online]. Available:  
<https://products.office.com/fr/project/project-and-portfolio-management-software?>
- [55] Jira, 2019. [Online]. Available:  
<https://www.atlassian.com/software/jira>.
- [56] Sinnaps, 2019. [Online]. Available:  
<https://www.sinnaps.com/en/product>.
- [57] Optical tools. WBS Schedule Pro, 2019. [Online]. Available:  
<http://criticaltools.com/Network%20Charts.html>.
- [58] Organaza, 2019. [Online]. Available: <https://organaza.com/features/>.
- [59] 43 полезных сервиса для управления проектами. Без эпитетов,  
2019. [Электронный ресурс]. Доступно: <https://habr.com/ru/post/276873/>.
- [60] Acunote, 2019. [Online]. Available: <https://www.acunote.com/>.
- [61] Application Development, 2019. [Online]. Available:  
<http://www.bestpricecomputers.co.uk/glossary/application-development.html>.
- [62] S. Wang, D. Samadhiya, and D. Chen, “Software Development and Quality Problems and Solutions by TRIZ”, in *International Symposium on Frontiers in Ambient and Mobile Systems*, MI, 2011, pp. 749 – 753.
- [63] Ещё раз про семь основных методологий разработки, 2019.  
[Электронный ресурс]. Доступно: <https://habr.com/company/edison/blog/2697/>.

- [64] 12 best software development methodologies with pros and cons, 2019. [Online]. Available: <https://acodez.in/12-best-software-development-methodologies-pros-cons/>
- [65] Waterfall Methodology in Project Management, 2019. [Online]. Available: <https://www.projectmanager.com/software/use-cases/waterfall>.
- [66] What is V-model- advantages, disadvantages and when to use it?, 2019. [Online]. Available: <http://tryqa.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it/>.
- [67] What is Incremental model- advantages, disadvantages and when to use it?, 2019. [Online]. Available: <http://tryqa.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/>.
- [68] What is iterative model?, 2019. [Online]. Available: <http://www.professionalqa.com/iterative-model>.
- [69] Е. А. Мирошниченко, *Технология программирования*. Томск, Россия: ТПУ, 2008.
- [70] R. Murch, *The Software Development Lifecycle – A Complete Guide*, Amazon Digital Services LLC, 2012.
- [71] Разработка программного обеспечения: этапы и принципы, 2019. [Электронный ресурс]. Доступно: <https://habr.com/ru/company/blog/267671/>.
- [72] K. Roebuck, *Systems Development Life Cycle (SDLC): High-impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors*, TN, USA: Lightning Source, 2011.
- [73] R. L. Glass, *Facts and Fallacies of Software Engineering*. Boston, USA: Addison-Wesley Professional, 2002.
- [74] M. Halkidi, and G. Tsatsaronis. “Data mining in software engineering”, *Intelligent Data Analysis*, pp. 413 – 441, 2011.
- [75] T. Xie, “Data Mining for Software Engineering”, *Computer*, vol. 42, no. 7, pp. 55 – 62, 2009.

[76] Q. Taylor, and Ch. Giraud-Carrier, “Applications of data mining in software engineering”, *Data Analysis Techniques and Strategies*, vol. 2, no. 3, pp. 243 – 257, 2010.

[77] А. А. Барсеагян, М. С. Куприянов, В. В. Степаненко, та И. И. Холод. *Технологии анализа данных: Data Mining, Visual Mining, text minig, OLAP*, Санкт-Петербург, Россия: БХВ-Петербург, 2007.

[78] Лекция 3: Методы и стадии Data Mining, 2019. [Электронный ресурс]. Доступно: <https://www.intuit.ru/studies/courses/6/6/lecture/162?page=2>.

[79] И. А. Чубукова, *Data Mining. Курс лекций интернет-университета INTUIT*, 2006.

[80] J. Anvik, “Automating bug report assignment”, in *Proceedings of the 28th International Conference on Software Engineering*, pp. 937 – 940, 2006.

[81] G. Canfora, and L. Cerulo, “Impact analysis by mining software and change request repositories”, in *Proceedings of the 11th IEEE International Software Metrics Symposium*, 2005.

[82] M. Christodorescu, S. Jha, and C. Kruegel, “Mining specifications of malicious behavior”, in *Proceedings of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering*, pp. 5 – 14, 2007.

[83] W. Dickinson, D. Leon, and A. Podgurski, “Finding failures by cluster analysis of execution profiles”, in *Proceedings of the 23rd International Conference on Software Engineering*, pp. 339 – 348, 2001.

[84] J. S. Shirabad, T. C. Lethbridge, and S. Matwin, “Supporting software maintenance by mining software update records”, in *Proceedings of the IEEE International Conference on Software Maintenance*, pp. 22 – 31, 2001.

[85] D. Hand, H. Mannila, and P. Smyth, *Principles of data mining (adaptive computation and machine learning)*, Cambridge, UK: The MIT Press, 2001.

- [86] M. Sharma, and M. Kumari. “Bug Assignee Prediction Using Association Rule Mining”, *ICCSA 2015. Lecture Notes in Computer Science*, pp. 444 – 457, 2015.
- [87] A. Berson, and S. Smith, *Data Warehousing, Data Mining, and OLAP*, New York, USA: McGraw-Hill Education, 2007.
- [88] Q. uz Zaman, A. Nadeem, and M. A. Sindhu, Formalizing a Use Case to a Kripke Structure, *Proceedings of the IASTED International Symposium Software Engineering and Applications*, 2015, pp. 232 – 239.
- [89] C. Stirling, “Modal and temporal logics”, Great Britain: University of Edinburgh, Department of Computer Science, 1991.
- [90] M. Sindhu, “Algorithms and Tools for Learning-based Testing of Reactive Systems”, PhD thesis, KTH Royal Institute of Technology, Stockholm, 2013.
- [91] С. СИНИЦЫН, и Н. НАЛЮТИН. *Верификация программного обеспечения*. Москва, Россия: МИФИ, 2006.
- [92] D. Dranidis, K. Tigka, and P. Kefalas, “Formal modelling of use cases with X-machines”, *Proceedings of the 1st South-East European Workshop on Formal Method*, no. 3, pp. 72 – 83, 2003.
- [93] User Story vs Use Case, 2019. [Online]. Available: <https://www.visual-paradigm.com/guide/agile-software-development/user-story-vs-use-case/>
- [94] А. В. Марков, “Совокупное использование сетей петри и uml-диаграмм при разработке программного обеспечения”, *Сборник научных трудов НГТУ*, № 2, с. 85 – 94, 2011.
- [95] G. Denaro, M. Pezze, “Petri Nets and Software Engineering”, *in Computer Science*, vol. 29, pp. 439 – 466, 2003.
- [96] Th. Weng Jie, and M. Ariff Ameen, “A Model Driven method to represent Free Choice Petri Nets as Sequence Diagram”, *Software Engineering and Computer Systems*, №14, pp. 23 – 32, 2015.

- [97] H. Singh, and Pr. Pal. Software Reliability Testing using Monte Carlo Methods International, *Journal of Computer Applications*, no. 4, pp. 41 – 44, 2013.
- [98] Т. Зайко. Ассоциативные правила в интеллектуальном анализе данных, *Вестник Национального технического университета Харьковский политехнический институт*, № 39, с. 82 – 96, 2013.
- [99] C. Zhang. *Association Rule Mining, Models and Algorithms*. Berlin, Germany: Springer, 2001.
- [100] J. Adamo, *Data Mining for Association Rules and Sequential Patterns: Sequential and Parallel Algorithms*, NY, USA: Springer, 2001.
- [101] Y. S. Koh, and N. Rountree, *Rare Association Rule Mining and Knowledge Discovery: Technologies for Infrequent and Critical Event Detection*. Hershey, PA: IGI Publishing, 2009.
- [102] X. Piao, Z. Wang, and G.Liu. “Research on Mining Positive and Negative Association Rules Based on Dual Confidence” in *Fifth International Conference on Internet Computing for Science and Engineering*, Heilongjiang, 2010, pp. 102 – 105.
- [103] Y. Ke, J. Cheng, and W. Ng, “An information-theoretic approach to quantitative association rule mining”, *Knowledge and Information Systems*, pp. 213–244, 2008.
- [104] H. Nam, “Identification of temporal association rules from time-series microarray data sets”, *Proceedings of the 2nd international workshop on Data and text mining in bioinformatics*, California, 2008, pp. 21 – 28.
- [105] R. Agrawal, and R. Srikant, “Fast Algorithms for Mining Association Rules in Large Databases”, *Proceedings of the 20th International Conference on Very Large Data Bases*, Chile, 1994, pp. 487 – 499.
- [106] R. Agrawal, and R. Srikant, “Fast algorithms for mining association rules”, *Proceedings of the 20th International Conference on Very Large Data Bases*, San Francisco, pp. 487 – 499, 1994.

[107] J. H. Mojdeh, and R. Z. Osmar, “A Study on Interestingness Measures for Associative Classifiers”, in *Proceedings of the 2010 ACM Symposium on Applied Computing*, pp. 1039 – 1046, 2010.

[108] Sh. Shang, X. Dong, and J. Li, “Algorithms for mining negative association rules in multi-database”, *Computer Engineering and Applications*, vol. 45, no. 24, pp. 150 – 152, 2009.

[109] Ассоциативные правила. Сравнительный анализ инструментария, 2019. [Электронный ресурс]. Доступно: [https://www.researchgate.net/publication/315629798\\_Associativnyye\\_pravila\\_Sravnitelnyj\\_analiz\\_instrumentaria](https://www.researchgate.net/publication/315629798_Associativnyye_pravila_Sravnitelnyj_analiz_instrumentaria).

[110] R. Natarajan, and B. Shekar, “Interestingness of association rules in data mining: Issues relevant to e-commerce”, *Sadhana*, pp. 291 – 309, 2005.

[111] J. Manimaran, Analysing the quality of Association Rules by Computing an Interestingness Measures, *Indian Journal of Science and Technology*, vol.8, no.15, 2015.

[112] P. Fournier-Viger. How to auto-adjust the minimum support threshold according to the data size, Philippe Fournier-Viger. 2019. [Online]. Available: <http://data-mining.philippe-fournier-viger.com/how-to-auto-adjust-the-minimum-support-threshold-according-to-the-data-size/>.

[113] P. Fournier-Viger. “Un modèle hybride pour le support à l’apprentissage dans les domaines procéduraux et mal-définis”, Ph.D. Thesis, University of Quebec in Montreal, Montreal, Canada, 2010.

[114] J. Manimaran, and T. Velmurugan, “A survey of Association Rule Mining in Text Applications”, in *IEEE International Conference on Computational Intelligence and Computing Research*, pp. 698 – 702, 2013.

[115] R. Agrawal, and R. Srikant. “Mining sequential patterns”, *ICDE '95 Proceedings of the Eleventh International Conference on Data Engineering*, Washington, 1995, pp. 3 – 14.

[116] J. Han, J. Pei, and Y. Yin. “Mining frequent patterns without candidate generation”, *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, New York, 2010, pp. 1 – 12.

[117] R. Raval, J. Jeet Rajput, and V. Gupta. “Survey on several improved Apriori algorithms”, *IOSR Journal of Computer Engineering*, 2013, pp. 57 – 61.

[118] F. Sha, and Zh. Hangjun, “The Research and Improvement on the Apriori Algorithm of the Association Rule Mining”, *2nd International Workshop on Intelligent Systems and Applications*, Wuhan, 2013, pp. 110 – 114.

[119] J. Hipp. “Algorithms for Association Rule Mining – A General. Survey and Comparison”, *ACM SIGKDD Explorations Newsletter*, New York, 2000, pp. 58 – 64.

[120] D. Tsur, J. Ullman, S. Abiteboul, and Ch. Clifton, “Query flocks: a generalization of association-rule mining”, in *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pp.1 – 12, 1998.

[121] A. Swami, R. Agrawal, and T. Imielinski, “Mining Association Rules between Sets of Items in Large Databases”, *International Conference on Management of Data*, Washington, 1993, pp. 207 – 216.

[122] M. Danubianu, St. Pentiu, and I. Tobolcea, “Set-Oriented Mining for Association Rules in Relational Databases”, *The Sixth International Multi-Conference on Computing in the Global Information Technology*, Luxembourg, pp. 14 – 19, 2011.

[123] D. J. Lubinsky, “Discovery from Databases: A Review of AI and Statistical Techniques”, in *IJCAI-89 Workshop on Knowledge Discovery in Databases*, pp. 204 – 218, 1989.

[124] A. Trupti, and V. Santoch, “An Overview of Association Rule Mining Algorithms”, *International Journal of Computer Science and Information Technologies*, vol. 5, no. 1, pp. 927 – 930, 2014.

[125] A. Savasere, E. Omiecinski, and B. Navathe. “An Efficient Algorithm for Mining Association Rules in Large Databases”, *Proceedings of the 21th International Conference on Data Bases*, San Francisco, 1995, pp. 432 – 444.

[126] M. Zaki, and S. Parthasarathy, M. Ogihara, and Wei Li, “New Algorithms for Fast Discovery of Association Rules”, *KDD-97 Proceedings*, pp. 283 – 286, 1997.

[127] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. Inkeri Verkamo. “Finding interesting rules from large sets of discovered association rules”, in *Proceedings of the Third International Conference on Information and Knowledge Jlanagement*, pp. 401 – 407, 1994.

[128] G. Grahne, and J. Zhu, “Fast algorithms for frequent itemset mining using FP-trees”, in *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 10, pp. 1347 – 1362, 2005.

[129] S. Archana, and D. Elangovan, “Survey of classifications techniques in Data Mining”, *International Journal of Computer Science and Mobile Applications*, vol. 2, 2014, pp. 65 – 71.

[130] A. A. Freitas, *Data Mining and Knowledge Discovery with Evolutionary Algorithms*, Berlin, Germany: Springer-Verlag, 2002.

[131] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*, USA: Chapman & Hall, 1984.

[132] CSE5230 Tutorial: The ID3 Decision Tree Algorithm, Monash university. 2019. [Online]. Available: [http://web.info.uvt.ro/~dzaharie/dm2017/EN/projects/Algorithms/DecisionTrees/DecisionTrees\\_ID3Tutorial.pdf](http://web.info.uvt.ro/~dzaharie/dm2017/EN/projects/Algorithms/DecisionTrees/DecisionTrees_ID3Tutorial.pdf).

[133] ID3 ALGORITHM. 2019. [Online]. Available: <https://www.slideshare.net/HARDIKSINGH7/id3-algorithm-56632554>

[134] G. A. Wang, “Survey on Training Algorithms for Support Vector Machine Classifiers”, *Fourth International Conference on Networked Computing and Advanced Information Management*, Gyeongju, 2008.

[135] L. Auria, and R. A. Moro, “Vector Machines (SVM) as a Technique for Solvency Analysis”, *SSRN Electronic Journal*, vol. 1, no. 1, pp. 1 – 8, 2008.

[136] Классификатор kNN. 2019. [Электронный ресурс]. Доступно: <https://habrahabr.ru/post/149693/>.



- [137] K-nearest neighbours. 2019. [Online]. Available: <http://www.cs.upc.edu/~bejar/apren/docum/trans/03d-algind-knn-eng.pdf>.
- [138] P. Bhargavi, and Dr. S. Jyothi Soil, Classification Using Data Mining Techniques: A Comparative Study, in *International Journal of Engineering Trends and Technology*, 2011.
- [139] С. В. Субботин, и Д. Ю. Большаков, “Применение байесовского классификатора для распознавания классов целей”, *Журнал радиоэлектроники*, №4, 2006.
- [140] Naive Bayes Classifier 2019. [Online]. Available: <http://responsive.media.mit.edu/wp-content/uploads/sites/5/2014/01/Class-4-Naive-Bayes.pdf>
- [141] C. Maffort. Mining Architectural Patterns Using Association Rules. *International Conference on Software Engineering and Knowledge Engineering*, Boston, 2013.
- [142] M. Azzeh. Software Stage-Effort Estimation Using Association Rule Mining and Fuzzy Set Theory. 2019. [Online]. Available: <https://arxiv.org/ftp/arxiv/papers/1703/1703.04539.pdf>.
- [143] V. Dubovoy, I. Pylypenko, and R. Stets, *Markov model application for analyzing cyclicity influence on the branched technological process management*, SWVNTU, no. 4, 2015.
- [144] В. И. Тихонов, и М. А. Миронов, *Марковские процессы*, Москва, Россия : Советское радио, 1977.
- [145] Великий тлумачний словник сучасної української мови: 250000 / уклад. та голов. ред. В. Т. Бусел, Київ, Україна: Перун, 2005.
- [146] Е. Б. Дынкин, Марковские процессы и связанные с ними задачи анализа, *УМН*, № 2, 1960.
- [147] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern Oriented Software Architecture*, New Jersey, USA, 2002.
- [148] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, Boston, USA: Wesley, 2003.

- [149] Э. Дж. Брауде, *Технология разработки программного обеспечения*, Санкт-Петербург, Россия: Питер, 2004.
- [150] MVC херох парс. 2019. [Online]. Available: <https://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>
- [151] С. Рогачев, *Обобщенный Model-View-Controller. Каркас на основе шаблона проектирования MVC*, 2019. [Электронный ресурс]. Доступно: <https://rsdn.org/article/patterns/generic-mvc.xml>.
- [152] MVC Patterns 2019. [Online]. Available: <http://aalmiray.github.io/griffon-patterns/>
- [153] Д. Чедвик, Т. Снайдер, и Х. Панда. *ASP.NET MVC 4. Разработка реальных веб-приложений с помощью ASP.NET MVC*. Farnham, Surrey: O'reilly, 2013.
- [154] J. Rumbaugh, I. Jacobson, and G. Booch Addison. *The unified modeling language reference manual*. Boston, USA: Addison-Wesley, 1999.
- [155] О. М. Бевз, В. М. Папінов, та Ю. А. Скидан. *Проектування програмних засобів систем управління*. Вінниця, Україна: ВНТУ, 2010.
- [156] Е. Гемма, Р. Хелм, *Приемы объектно-ориентированного проектирования. Паттерны проектирования*, СПб, Россия: Питер, 2001.
- [157] Отношения классов – от UML к коду, 2019. [Электронный ресурс]. Доступно: <https://habr.com/ru/post/150041/>.
- [158] SQL или NoSQL – вот в чём вопрос. 2019. [Электронный ресурс]. Доступно: <https://habr.com/company/ruvds/blog/324936/>.
- [159] О. Н. Романюк, та Т. О. Савчук, *Організація баз даних і знань: навчальний посібник*. Вінниця, Україна: УНІВЕРСУМ-Вінниця, 2003.
- [160] J. MacLennan, Z. Tang, and B. Crivat, *Data Mining with Microsoft SQL Server 2005*. Indianapolis, USA: Wiley Publishing, 2008.
- [161] Д. Боуман, С. Эмерсон, и М. Дарновски, *Практическое руководство по SQL*. New-York, USA: Williams, 2002.