

## АНОТАЦІЯ

Жеваго О. О. Моделювання і аналіз процесів розробки та налагодження комп'ютерних програм. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки». – Дніпровський національний університет залізничного транспорту імені академіка В. Лазаряна, Дніпро, 2021.

### Зміст анотації

Дисертаційна робота присвячена дослідженню і вирішенню актуальної науково-практичної задачі аналізу процесів розробки та налагодження комп'ютерних програм, для підвищення якості навчання студентів програмуванню. Розуміння того, як студенти розробляють та налагоджують програмне забезпечення, і проблем, з якими вони стикаються, має великий потенціал для підвищення якості навчання програмування. А, як відомо, якість програмного забезпечення безпосередньо пов'язана з якістю відповідного процесу розробки. Стандартний підхід до оцінювання, який враховує тільки результат, не відповідає сучасним вимогам до навчання програмуванню.

В основі дослідження лежить припущення про те, що інструменти, які збирають дані про використання середовища розробки, дають більш детальне розуміння роботи розробників. Аналізуючи, як програмісти використовують середовище розробки, можна виявити закономірності поведінки та визначити проблеми, з якими вони стикаються.

У першому розділі виконано огляд та аналіз наявних підходів до отримання інформації щодо процесів розробки та налагодження програмного забезпечення. Показано, що багатогранні дослідження не ґрунтуються на моделях процесів і не дають можливості вивчення процесу конкретного розробника.

Встановлено на основі існуючих досліджень, що постійний моніторинг призводить до значного поліпшення успішності студентів.

Продемонстровано, що високий рівень практичної підготовки з програмування передбачає, що розробник володіє ефективним стилем розробки програм, що включає:

- застосування методу покрокової деталізації;
- навички декомпозиції;
- відповідність тексту програми стандартам програмування.

Дослідження в області навчання показали, що індивідуальний зворотний зв'язок допомагає студентам вчитися. Однак дати такий зворотний зв'язок студентам, які навчаються програмування, досить складно.

З'ясовано, що дослідження в області розробки програмного забезпечення зазвичай ґрунтуються на даних, отриманих з систем контролю версій, але не з середовища розробки. Система контролю версій не може оцінити внесок студента в кінцевий результат, оскільки не надає інформацію про процес розробки. Вони лише фіксують історію змін між комітами та не відображають реальну еволюцію коду.

На відміну від цього в дисертації інформація щодо процесів написання та налагодження програм отримується безпосередньо з середовища розробки.

У другому розділі за допомогою інструментів конструктивно-продукційного моделювання формалізовано процеси розробки та налагодження програмного забезпечення.

Показано універсальність і перспективність використання конструктивно-продукційного моделювання для вирішення завдань різних предметних областей.

Розроблено конструктори для:

- формування історії розробки програмного забезпечення при використанні середовища розробки для подальшого автоматичного аналізу;
- класифікації історії написання тексту програми по типах змін і побудови файлу журналу подій з метою вивчення процесу розробки студентами комп'ютерних програм;
- формування журналу подій при налагодженні програм в середовищі розробки.

Визначено зв'язок конструктивно-продукційного підходу та використання методів Process Mining з об'єктно-орієнтованими моделями.

На основі конструктивних моделей створено доповнення до середовища розробки Visual Studio для реалізації можливості відслідковування роботи кожного студента як на лабораторних заняттях, так і під час самостійної роботи, в якому всі дії по розробці та налагодженню фіксуються в журналах подій.

Використано методи Process Mining для аналізу і побудови моделей процесів розробки та налагодження програм. Метод Discovery – для формування моделей процесів на основі файлів журналів подій. Метод Conformance Checking – для порівняння різних виконань процесів і виявлення поведінкових схожостей та відмінностей.

У третьому розділі приведені результати експериментальних досліджень. Переверіено та підтверджено можливість застосування сформованих конструктивно-продукційних моделей та розроблених на їх основі інструментальних засобів для відстеження процесів розробки та налагодження програмного забезпечення.

Виконано експеримент у вигляді олімпіади з налагодження, під час якого за допомогою розроблених інструментів відстежувалися та фіксувалися дії учасників в середовищі розробки.

Для експерименту розроблені програми з засіяними типовими логічними помилками, які визначені на основі досліджень про найбільш поширені помилки початківців. Кожного разу логічна помилка засіяна у одному рядку коду програми. Всі програми синтаксично коректні, але виконання кожної з них призводить до некоректного результату. Щоб допомогти виявити всі помилки, студентам надавався тест з вхідними та вихідними даними, що відповідають специфікації.

У експерименті взяв участь 41 студент з першого по п'ятий курси. Всі вони мали попередній досвід розробки на C-подібних мовах програмування і використання середовища розробки Visual Studio. Всього студентам треба було виконати п'ятнадцять завдань за 4 години.

Під час експерименту сформовано 487 файлів журналів подій для 41 учасника. Загалом журнали подій складаються з 2415 сесій налагодження та 16536 подій.

По результатах експерименту проведено класифікацію учасників на основі кількості правильно виконаних завдань як "High", "Middle", "Low". Студенти відносяться до класу "Low", якщо вони виконали менше половини завдань, таких – 9. Студенти з "Middle" класу виконали більше половини завдань, але не всі, таких – 23. Студенти з "High" класу виконали всі завдання, таких – 9.

Результати показали навички студентів по налагодженню програм з типовими логічними помилками. Вдалося виявити проблеми, які є більш складними для локалізації та виправлення.

Для розуміння поведінки учасників з кожного класу засобами Process Mining сформовано відповідні моделі процесів на основі журналів подій.

На основі отриманих моделей процесів налагодження та найбільш часто повторювальних сесій вдалося виявити 4 шаблони поведінки учасників експерименту.

За допомогою методу Conformance Checking засобів Process Mining проведено порівняння процесів різних учасників з метою виявлення особливостей поведінки.

У четвертому розділі продемонстровано інструменти для автоматичного моніторингу та візуалізації процесів написання та налагодження програмного забезпечення.

Розроблені інструменти дозволяють:

- фіксувати взаємодію розробників із середовищем розробки;
- формувати журнали подій про процеси розробки та налагодження;
- візуалізувати процес розробки у вигляді 3D зображення з можливістю масштабування та обертання.

Для розширення середовища розробки Visual Studio можливістю відстежувати увесь процес розробки та налагодження програм розроблено VSPackage за допомогою Microsoft Visual Studio SDK.

На рівні процесу фіксуються події з базовою інформацією, такою як тип події (наприклад, зміна або збереження коду) та час. На контекстному рівні зберігаються конкретні дані залежно від типу події. Наприклад, події збереження зберігають ім'я файлу, який було збережено, та спосіб, за допомогою якого було введено дію збереження (наприклад, за допомогою гарячих клавіш чи вибору пункту меню).

Для збереження та аналізу даних, щодо процесів розробки та налагодження розроблено об'єктно-орієнтовану модель, яка відповідає терміналам з їх атрибутами і базується на моделі подій в середовищі розробки.

Наукова новизна отриманих результатів полягає в тому що в роботі вперше:

- виконано формалізацію процесів розробки та налагодження програмного забезпечення засобами конструктивно-продукційного моделювання, що на відміну від існуючих дозволяє розглядати ці процеси як послідовність елементарних дій за продукційними правилами, формалізувати процеси формування журналів подій та візуалізувати ці процеси;

- розроблені моделі процесів розробки та налагодження програм засобами Process Mining, що на відміну від існуючих дають можливість автоматизувати аналіз цих процесів;

- розроблені інструменти для збору даних щодо дій програміста у процесі розробки та налагодження програмного забезпечення з середовища розробки Visual Studio.

Дисертація є частиною науково-дослідної роботи «Конструктивно-продукційне моделювання в задачах розробки програмного забезпечення» (2021 р. № держреєстрації 0121U109167), яка виконана на кафедрі «Комп'ютерних інформаційних технологій» Дніпровського національного університету залізничного транспорту імені академіка В. Лазаряна.

**Ключові слова:** конструктивно-продукційне моделювання, конструктори, Process Mining, розробка програмного забезпечення, налагодження, огляд коду, навчання, інженерія програмного забезпечення.

## ABSTRACT

Zhevaho O. O. Modeling and analysis of the process of developing and debugging computer programs. – Qualifying scientific work on the rights of the manuscript.

Thesis for obtaining the Ph. D. degree in specialty 122 – Computer Science. – Dnipro National University of Railway Transport named after Academician V. Lazaryan, Dnipro, 2021.

### Abstract content

The dissertation is devoted to research and solving the urgent scientific and practical task of analyzing the processes of developing and debugging computer programs, in order to improve the quality of students' programming education. Understanding how students develop and debug programs and what problems they encounter has great potential to improve the quality of programming instruction. And, as we know, the quality of the software is directly related to the quality of the corresponding development process. The standard approach to assessment, which considers only the result, does not meet modern requirements for teaching programming.

The study is based on the assumption that tools that collect data on development environment usage provide a more detailed understanding of developers' work. By analyzing how programmers use the development environment, patterns of behavior can be identified and the problems they encounter can be identified.

The first chapter reviews and analyzes existing approaches to obtaining information about software development and debugging processes. It is shown that multifaceted studies are not based on process models and do not provide an opportunity to study the process of a particular developer.

Based on existing research, it has been found that continuous monitoring leads to significant improvements in student achievement.

It is shown that a high level of practical training in the field of programming implies that the developer has an effective style of program development, including:

- application of the step-by-step method;
- decomposition skills;
- compliance of the program text with programming standards.

Research in the field of learning has shown that individualized feedback helps students learn. However, giving such feedback to programming students is quite difficult.

It has been found that research in software development is usually based on data from version control systems but not from the development environment. Version control systems cannot assess the student's contribution to the final result because they do not provide information about the development process. They only record the history of changes between commits and do not reflect the actual evolution of the code.

In contrast, in this dissertation, information about the processes of writing and debugging programs is obtained directly from the integrated development environment.

The second chapter formalizes the processes of software development and debugging by using constructive-synthesizing modeling tools.

The versatility and perspectivity of the use of constructive-synthesizing modeling to solve problems in various subject areas is shown.

Constructors have been developed for:

- forming the history of software development using the development environment for further automatic analysis;
- classifying the history of writing program text by type of change and constructing a log file in order to study the process of computer program development by students;
- formation of a log of events during the debugging of programs in the development environment.

The connection between the constructive-synthesizing modeling and the use of Process Mining methods with object-oriented models is determined.

Based on constructive models, extension to the Visual Studio development environment were created to implement the ability to track each student's work both in labs and during independent work, where all development and debugging activities are recorded in event logs.

Process Mining methods are used to analyze and build process models for software development and debugging. The Discovery method is used to build process models based on event log files. The Conformance Checking method is used for comparing different processes and identifying behavioral similarities and differences.

The third chapter presents the results of experimental research. The possibility of using the generated structural models and the tools developed on their basis for tracking the software development and customization processes is tested and confirmed.

An experiment in the form of a debugging olympiad was conducted in which participants' actions in the development environment were tracked and recorded using the developed tools.

For the experiment, programs were designed with seeded typical logical errors, which were determined based on a study of the most common beginner's errors. Each time a logical error is seeded in one line of program code. All of the programs are syntactically correct, but the execution of each leads to the wrong result. To help identify all errors, students were given a test with inputs and outputs that matched the specification.

The experiment involved 41 students from the first to the fifth year. All of them had experience in C-like programming languages and using Visual Studio development environment. In total, students had to complete fifteen tasks in four hours.

During the experiment, 487 event log files were generated for 41 participants. In total, the event logs consist of 2415 debugging sessions and 16536 events.

According to the results of the experiment, the participants were classified on the basis of the number of correctly performed tasks as "High", "Middle", "Low". Students belong to the class "Low" if they have completed less than half of the tasks, such – 9. Students from the "Middle" class have completed more than half of the tasks, but not all, such – 23. Students from the "High" class have completed all tasks, such – 9.



The results showed the skills of students to debug programs with typical logical errors. We were able to identify problems that are more difficult to localize and fix.

To understand the behavior of participants from each class by means of Process Mining, appropriate process models are formed on the basis of event logs.

Based on the obtained models of debugging processes and the most frequently repeated sessions, it was possible to identify 4 patterns of behavior of the participants of the experiment.

Using the Conformance Checking method of Process Mining, the processes of different participants were compared in order to identify the characteristics of behavior.

The fourth chapter demonstrates tools for automatic monitoring and visualization of software writing and debugging processes.

Developed tools allow:

- to record the interaction of developers with the development environment;
- to form event logs about development and debugging processes;
- visualize the development process in the form of a 3D image with the ability to scale and rotate.

To extend the Visual Studio development environment with the ability to track the entire process of program development and debugging, VSPackage was developed using the Microsoft Visual Studio SDK.

At the process level, events are logged with basic information such as event type (e.g., code change or save) and time. At the context level, specific data is stored depending on the type of event. For example, save events store the name of the file that was saved and the way in which the save action was entered (e.g., by using hotkeys or selecting a menu item).

To store and analyze data about the development and debugging processes, an object-oriented model was developed that corresponds to terminals with their attributes and is based on the development environment event model.

The scientific novelty of the results is that for the first time in the work:

- formalization of software development and debugging processes by means of constructive-synthesizing modeling, which, unlike the existing ones, allows to consider

these processes as a sequence of elementary actions according to production rules, to formalize the processes of event logs formation and visualize these processes;

- developed models of processes of development and debugging of programs by means of Process Mining that unlike existing give the chance to automate the analysis of these processes;

- developed tools for collecting data on the actions of the programmer in the process of developing and debugging software from the Visual Studio development environment.

The dissertation is part of the research work «Modeling in software development tasks» (2021 year, registration number 0121U109167), which performed at the «Department of Computer Information Technologies», Dnipro National University of Railway Transport named after Academician V. Lazaryan.

**Keywords:** constructive-synthesizing modeling, constructors, Process Mining, software development process, debugging, code review, education, software engineering.

## СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

**Праці включені до міжнародних наукометричних баз (МНБД) Scopus та Web of Science:**

1. Shynkarenko, V., Zhevago, O. Development of a toolkit for analyzing software debugging processes using the constructive approach // Eastern-European Journal of Enterprise Technologies. – 2020. – pp. 29–38. <https://doi.org/10.15587/1729-4061.2020.215090>. *Стаття у фаховому журналі, що індексується у МНБД Scopus та має Q3 за даними Scimago Journal & Country Rank, зараховується як дві публікації.*

2. Shinkarenko V. I., Zhevago O. O. Generating university course timetable using constructive modeling // Radio Electronics, Computer Science, Control. – 2019. – pp. 152–162. <https://doi.org/10.15588/1607-3274-2019-3-17>. *Стаття у фаховому журналі, що індексується у МНБД Web of Science.*

3. Shynkarenko V., Zhevago O. Constructive modeling of the software development process for modern code review // IEEE 15th International Conference on Computer Sciences and Information Technologies. – 2020. – pp. 392–395. <https://doi.org/10.1109/CSIT49958.2020.9322002>. *Матеріали конференції, індексується у МНБД Scopus.*

4. Shynkarenko, V., Zhevago, O. Visualization of program development process // IEEE 14th International Conference on Computer Sciences and Information Technologies. – 2019. – pp. 142–145. <https://doi.org/10.1109/stc-csit.2019.8929774>. *Матеріали конференції, індексується у МНБД Scopus.*

**Праці у фахових виданнях затверджених МОН України:**

5. Zhevago O. O. An overview of tools for collecting data on software development and debugging processes from integrated development environments // Наука та прогрес транспорту. Вісник Дніпропетровського національного університету залізничного транспорту – 2021 – С. 24–37. <https://doi.org/10.15802/stp2021/242042>.

6. Шинкаренко В. І., Жеваго О. О. Експериментальні дослідження процесів налагодження комп'ютерних програм студентами з використанням Process Mining // Вісник Херсонського національного технічного університету – 2021 – С. 83–98. <https://doi.org/10.35546/kntu2078-4481.2021.3.10>.

### **Матеріали наукових конференцій:**

7. Жеваго О. О., Шинкаренко В.І. Дослідження методів відстеження процесів розробки та відлагодження програм // Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті: тези XI Міжнародної науково-практичної конференції, м. Дніпро. – 2017. – С. 103.

8. Жеваго О. О., Шинкаренко В.І. Дослідження стилю відлагодження програм // Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті: тези XIII Міжнародної науково-практичної конференції, м. Дніпро. – 2019. – С. 73.

9. Жеваго О. О., Шинкаренко В.І. Конструктивне моделювання та аналіз процесів розробки і відлагодження програм // Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті: тези XIV Міжнародної науково-практичної конференції, м. Дніпро. – 2020. – С. 129.

10. Жеваго О. О., Шинкаренко В.І. Аналіз та вдосконалення навчального процесу з програмної інженерії з використанням методів Process Mining // Сучасні тенденції розвитку інформаційних систем і телекомунікаційних технологій: тези III Міжнародної науково-практичної конференції, м. Київ. – 2021. – С. 73–74.

11. Жеваго О. О., Шинкаренко В. І. Виявлення навичок відлагодження програм методом підсіву помилок // Інформаційні Технології в Металургії та Машинобудуванні: тези Міжнародної науково-технічної конференції ІТММ 2021, м. Дніпро. – 2021. – С. 339–340.

12. Жеваго О. О., Шинкаренко В. І. Составление расписания занятий на основе конструктивного моделирования // Проблемы математического моделирования: Матеріали Всеукраїнської науково-методичної конференції, м. Кам'янське. – 2018. – С. 59–63.