

УПРАВЛІННЯ РОЗРОБКОЮ ПРОЄКТІВ ПРОДУКТОВИХ ІТ-КОМПАНІЙ НА ПЛАТФОРМАХ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ

©2021 СОКОЛОВСЬКА З. М., КЛЕПІКОВА О. А., ЯЦЕНКО Н. В., МАРЧЕНКО А. В.

УДК 330.46
JEL: C63; M15; M21

Соколовська З. М., Клепікова О. А., Яценко Н. В., Марченко А. В. Управління розробкою проєктів продуктивних ІТ-компаній на платформах імітаційного моделювання

Метою статті є розкриття можливостей застосування апарату імітаційного моделювання в управлінні розробкою проєктів продуктивних ІТ-компаній. На основі аналізу ряду літературних джерел проведено огляд досліджень у галузі управління ІТ-проєктами та охарактеризовано наявний стан застосування інструментарію імітації в розв'язанні прикладних завдань. Обґрунтовується відповідність імітаційних технологій специфіці завдань, притаманних процесам розробки програмних продуктів. Згідно з життєвим циклом створення програмних продуктів наведено стислу характеристику типових етапів просування проєктів. Запропоновано імітаційну модель, що відтворює динаміку розгортання продуктивних проєктів ІТ-компаній. Модель побудована з використанням гібридного підходу – комбінації дискретно-подієвої та агентної методологій – на програмній платформі багатопідходного імітаційного моделювання AnyLogic. Дискретно-подієвий підхід забезпечує високий ступінь деталізації процесів, що моделюються. Агентний підхід дозволяє визначити проєкт як динамічну одиницю зі специфічними властивостями, що надає можливість урахування особливостей продуктів, замовлених споживачами – клієнтами ІТ-фірми. Робота моделі продемонстрована за результатами стандартних та оптимізаційних імітаційних експериментів, реалізованих на умовних даних усередненої продуктової ІТ-компанії. Наведені ситуації доводять можливість застосування моделі як платформи прийняття управлінських рішень керівником команди розробки продуктивних проєктів. Модель представлена як тренажер з модульною, відкритою архітектурою та параметричним налаштуванням на конкретні умови експериментів. Може бути рекомендована для впровадження в практику продуктивних ІТ-компаній для управління проєктами розробки програмного забезпечення.

Ключові слова: продуктова ІТ-компанія, управління, проєкт, імітаційне моделювання, дискретно-подієвий підхід, агентний підхід, імітаційний експеримент.

DOI: <https://doi.org/10.32983/2222-4459-2021-5-108-123>

Рис.: 6. Табл.: 1. Бібл.: 26.

Соколовська Зоя Миколаївна – доктор економічних наук, професор, завідувачка кафедри економічної кібернетики та інформаційних технологій, Державний університет «Одеська політехніка» (просп. Шевченка, 1, Одеса, 65044, Україна)

E-mail: nadin_zs@te.net.ua

ORCID: <http://orcid.org/0000-0001-5595-7692>

Researcher ID: AAT-5469-2020

Scopus Author ID: <https://www.scopus.com/authid/detail.uri?authorId=36081195900>

Клепікова Оксана Ананіївна – кандидат економічних наук, доцент, доцент кафедри економічної кібернетики та інформаційних технологій, Державний університет «Одеська політехніка» (просп. Шевченка, 1, Одеса, 65044, Україна)

E-mail: kleroa@ukr.net

ORCID: <http://orcid.org/0000-0003-0399-9035>

Researcher ID: AAE-7844-2021

Scopus Author ID: <https://www.scopus.com/authid/detail.uri?authorId=36080730500>

Яценко Наталія Володимирівна – старший викладач кафедри економічної кібернетики та інформаційних технологій, Державний університет «Одеська політехніка» (просп. Шевченка, 1, Одеса, 65044, Україна)

E-mail: natali.jazenko@gmail.com

ORCID: <http://orcid.org/0000-0002-0355-0871>

Researcher ID: <https://publons.com/researcher/1810997/natalia-yatsenko/>

Марченко Антон Вікторович – магістр кафедри економічної кібернетики та інформаційних технологій, Державний університет «Одеська політехніка» (просп. Шевченка, 1, Одеса, 65044, Україна)

E-mail: Marh_a@ukr.net

UDC 330.46
JEL: C63; M15; M21

Sokolovska Z. M., Klepikova O. A., Yatsenko N. V., Marchenko A. V. Management of Development of Projects of Product IT Companies on Simulation Platforms
The article is aimed at disclosing the possibilities of using the simulation modeling apparatus in the management of the development of projects of product IT companies. Based on the analysis of a number of literary sources, an overview of research in the field of IT project management is carried out and the existing status of application of the simulation instrumentarium in the solution of applied tasks is described. The compliance of simulation technologies with the special requirements of the tasks inherent in the processes of software products development is substantiated. According to the life cycle of software creation, a brief characterization of the typical stages of project promotion is provided. A simulation model is proposed that reproduces the dynamics of the deployment of product projects of IT companies. The model is built using a hybrid approach – a combination of discrete-event and agent-based methodologies – on the software platform for multi-platform simulation AnyLogic. The discrete-event approach provides a high degree of detail of the processes being modeled. The agent-based approach allows to define the project as a dynamic unit with specific properties, which makes it possible to take into account the peculiarities of products ordered by consumers – clients of an IT firm. The model's work is demonstrated by the results of standard and optimization simulation experiments implemented on conditional data of an average product IT company. These situations prove the possibility of using the model as a platform for making managerial decisions by the head of the product project development team. The model is presented as a simulator with the modular, open architecture and parametric adjustment for specific conditions of experiments. It may be recommended for product IT companies to practice management of the projects of software development.

Keywords: product IT company, management, project, simulation modeling, discrete-event approach, agent-based approach, simulation experiment.

Fig.: 6. **Tabl.:** 1. **Bibl.:** 26.

Sokolovska Zoia M. – D. Sc. (Economics), Professor, Head of the Department of Economic Cybernetics and Information Technologies, State University "Odesa Polytechnic" (1 Shevchenko Ave., Odesa, 65044, Ukraine)

E-mail: nadin_zs@te.net.ua

ORCID: <http://orcid.org/0000-0001-5595-7692>

Researcher ID: AAT-5469-2020

Scopus Author ID: <https://www.scopus.com/authid/detail.uri?authorId=36081195900>

Klepikova Oksana A. – PhD (Economics), Associate Professor, Associate Professor of the Department of Economic Cybernetics and Information Technologies, State University "Odesa Polytechnic" (1 Shevchenko Ave., Odesa, 65044, Ukraine)

E-mail: klepoa@ukr.net

ORCID: <http://orcid.org/0000-0003-0399-9035>

Researcher ID: AAE-7844-2021

Scopus Author ID: <https://www.scopus.com/authid/detail.uri?authorId=36080730500>

Yatsenko Natalia V. – Senior Lecturer of the Department of Economic Cybernetics and Information Technologies, State University "Odesa Polytechnic" (1 Shevchenko Ave., Odesa, 65044, Ukraine)

E-mail: natali.jazenko@gmail.com

ORCID: <http://orcid.org/0000-0002-0355-0871>

Researcher ID: <https://publons.com/researcher/1810997/natalia-yatsenko/>

Marchenko Anton V. – Master, Department of Economic Cybernetics and Information Technologies, State University "Odesa Polytechnic" (1 Shevchenko Ave., Odesa, 65044, Ukraine)

E-mail: Marh_a@ukr.net

Становлення вітчизняної ринкової економіки тісно пов'язано з виникненням і розвитком усіх видів підприємництва. Спектр послуг, що надають різноманітні суб'єкти підприємницької діяльності, постійно розширюється та стосується різних галузей виробництва.

Одним із сучасних секторів підприємництва є надання ІТ-послуг. ІТ-галузь демонструє стійке зростання, є прибутковою та сприяє залученню інвестицій, незважаючи на загальний кризовий стан економіки [1]. Це, своєю чергою, позитивно впливає на вміст державного бюджету та загальне економічне становище країни.

Велика кількість ІТ-компаній, що функціонують у сфері надання відповідних послуг, приводить до необхідності дослідження їх діяльності з метою подальшого підвищення ефективності. Цьому сприяє високий рівень конкуренції в ІТ-галузі; експансія зарубіжних компаній, зростання вимог потенційних користувачів до складу та якості інформаційного, технологічного та програмного забезпечення. Використання сучасних технологій в організації розробки програмних продуктів особливо актуально для продуктивних ІТ-фірм. У колі питань реалізації проектів – кінцевих програмних продуктів – важливе місце належить налагодженню взаємодії між учасниками процесу розробки та забезпеченню оптимального співвідношення ресурсів.

Розв'язання проблем, пов'язаних з процесами розробки програмного забезпечення, лежить в руслі дослідження прикладних аспектів реалізації й управління проектами, притаманними продуктивним ІТ-компаніям. Ефективним засобом дослідження виступає створення економіко-математичних та імітаційних моделей.

Різноманітним аспектам функціонування ІТ-фірм, зокрема спрямованих на розробку власних програмних продуктів, присвячено значне коло літературних джерел [1–13]. Проте постійне зростання вимог кінцевих користувачів та ускладнення процесів створення, реалізації та моніторингу програмного забезпечення приводить до необхідності використання гнучких інструментів управління проектами продуктивних ІТ-фірм на всіх етапах їх життєвого циклу. На це звертають увагу дослідники та фахівці-практики в галузі управління ІТ-проектами [4; 5; 11; 12].

Досягненню конкретних результатів сприяє відтворення відповідних процесів у динаміці з урахуванням впливів різноманітних збурюючих факторів, що приводять до відхилень від запланованого розгортання проектів. Існуючі аналітичні методи недостатньо відповідають специфіці поставленої задачі. Ефективним інструментом повсякденного використання для менеджменту ІТ-компанії можуть служити моделі-тренажери з параметричним налаштуванням на конкретні ситуації та зміни вимог клієнтів-замовників і розробників фірми. Побудова таких моделей можлива на базі імітаційних технологій моделювання, що дістали в останні роки значного розповсюдження в різних прикладних сферах діяльності [14–16].

Існують відповідні напрацювання і стосовно ІТ-галузі. Наприклад, у [17] пропонується промислова імітаційна модель компанії – розробника програмного забезпечення. Модель представлена як достатньо типова з рекомендаціями налаштування на специфіку конкретної компанії. Можливості вдосконалення управління продуктивними та аутсорсинговими ІТ-фірмами на базі використання моделей-тренажерів, побудованих на програмній платформі

AnyLogic, наведено також у [18]. При побудові моделей використана гібридна парадигма імітації. Робота моделей проілюстрована фрагментами імітаційних експериментів на базі двох типів фірм – продуктової та аутсорсингової. Здобутки у сфері управління окремими ІТ-проектами та портфоліо проектів представлено в роботах [19; 20]. Результати наукового дослідження [19] стосуються побудови моделей розподілу трудових ресурсів у процесі управління портфоліо ІТ-проектів.

Рекомендації щодо ефективного моделювання проектів і приклади промислового впровадження наведено у [20]. Моделюванню бізнес-процесів в умовах обмежених ресурсів присвячена робота [21]. Використання імітаційних методологій для створення синтезованого методу проведення досліджень у сфері моделювання бізнес-процесів наведено в роботі [22].

Інструментарій імітаційних досліджень динамічно розвивається. Тому окремої уваги заслуговують наукові праці, присвячені методології проведення імітаційних експериментів на моделях складних систем і процесів. Теорія планування експериментів привертає увагу дослідників з точки зору як теоретичної постановки, так і в аспекті різноманітних прикладних реалізацій. З цього приводу треба відмітити роботи [23] і [24]. Поступово розвиваються й основні парадигми імітаційного моделювання – дискретно-подієва, агентна та системна динаміка – база багатопідходної методології. З огляду на використання у представлених у даній роботі розробках дискретно-подієвого підходу, треба звернути увагу на матеріали [25], що розкривають ряд аспектів добре відомої методології в новому ракурсі.

Однак, незважаючи на значні здобутки у сфері імітаційного моделювання, ІТ-галузь належить до однієї зі сфер, що має порівняно невелику кількість промислових модельних додатків. Водночас існує достатньо задач, зокрема у сфері управління ІТ-проектами різної природи, у дослідженні яких цей математичний апарат може стати в нагоді.

Мета статті – розкриття можливостей застосування апарату імітаційного моделювання в управлінні розробкою проектів продуктових ІТ-компаній.

У межах діяльності продуктової ІТ-компанії можна відокремити такі підсистеми:

- ✦ планування нових функцій;
- ✦ дизайну;
- ✦ розробки та впровадження;
- ✦ тестування та підтримки системи.

Процес обстеження технології реалізації наведених підсистем пов'язаний з проведенням комплексного аналізу й обробкою великих масивів даних, що ефективно здійснювати на базі побудови моделі проектів створення програмних продуктів.

Розробка програмного забезпечення відбувається за певним життєвим циклом (*SDLC – software devel-*

opment life cycle), який є чітко визначеною, структурованою послідовністю етапів. Кожна фаза дає результати, необхідні для наступного етапу. Розглянемо типові етапи процесу розробки програмних продуктів.

Етап 1. Збір та аналіз вимог замовника до кінцевого програмного продукту: функціональний і фінансовий аналіз доцільності втілення нових завдань; визначення та фіксація вимог від зацікавлених сторін за допомогою інтерв'ю, опитувань, дискусій тощо; чітке документування вимог; створення прототипів.

Досліджується функціонал, що підлягає розробці; прогнозуються ринкові потреби на стратегічну перспективу; аналізуються майбутні можливості ринку апаратних і програмних засобів. На основі ринкового опису продукту продакт-менеджери розробляють бізнес-вимоги, тобто конкретизують функціональну складову продукту. Бізнес-вимоги надходять технічним аналітикам, які оцінюють їх достовірність і визначають терміни розробки функціональних складових у людино-днях теоретичного програміста. Крім того, виявляється, які функціональні складові вимагають додаткових досліджень для встановлення можливості реалізації. Усе це дозволяє скласти технічне завдання на розробку продукту, що розподіляється по існуючих ресурсах; встановити реальний термін розробки. Результат цієї роботи – план-графік виконання проекту з термінами розробки та ресурсами, що надходить до продакт-менеджера, який звіряє функціональність, реалізовану в плані-графіку, з функціональністю, зазначеною в бізнес-вимогах. На базі цього менеджер приймає рішення про початок розробки проекту та формує так званий роад-меп (документ, який показує термін і склад релізу продукту), де вказуються терміни технічного релізу продукту (терміни розробки та тестування), бета-релізу (розробка, тестування, зовнішнє тестування) та комерційного релізу (запуску масових продажів).

Етап 2. Проектування: розробка високорівневого дизайну системи. Визначаються технології, які будуть використовуватися; завантаження команди; обмеження; часові рамки та бюджет. Дизайн закріплюється окремим документом – дизайн-специфікацією (*Design Specification Document*). У нотаціях закріплюються блок-схеми; ER-діаграми; UML-діаграми; макети (наприклад, створений у Photoshop прототип).

Етап 3. Впровадження або написання програмного коду. Найважливішу роль на цьому етапі відіграє професійна кваліфікація керівника групи розробки. У його функції входять перевірка обґрунтованості планів, прийняття рішення про забезпечення та розподіл ресурсів, визначення стратегії розробки, управління ресурсами та їх ефективністю і т. д. Результатом етапу розробки є вихід продукту в строк із функціональністю, відповідною бізнес-вимогам. Системні адміністратори налаштовують програмне середовище; фронтенд-програмісти розробляють користувацький інтерфейс програми та логіку її

взаємодії із сервером. Контроль коректності роботи модулів системи здійснюється за допомогою створення юніт-тестів. Готове програмне забезпечення розгортається у програмному середовищі. Цей цикл повторюється до тих пір, поки всі вимоги не будуть реалізовані. Даний етап – найдовший у життєвому циклі розробки програмного забезпечення.

Етап 4. Тестування. Тестування включає: функціональне тестування (визначення працездатності заявленої функціональності); кейс-тестування (перевірка роботи продукту в різних модельованих ситуаціях); загальне регресійне тестування (повне дослідження працездатності всіх складових частин продукту в будь-яких ситуаціях). Іноді регресійного тестування можна уникнути, якщо нова версія продукту не сильно відрізняється від попередньої або архітектура продукту побудована так, що додана функціональність не зачіпає основної гілки продукту (у цьому випадку тестується тільки ця функціональність). Результатом тестування є технічний реліз продукту – передача готового продукту продакт-менеджеру для приймання. На цьому кроці передбачається перевірка роботи продукту всіма зацікавленими департаментами: технічною підтримкою, департаментом впровадження тощо. До того ж, на етапі тестування виявляються пропущені при розробці помилки системи (баги). При виявленні дефекту тестувальник складає звіт про помилку, який передає розробникам. Останні її виправляють, після чого тестування повторюється – цього разу для того, щоб переконатися в повному виправленні програмних дефектів. Цикл тестування продовжується до повної відповідності встановленим критеріям оцінки. Після цього продукт передається розробникам для усунення помилок, знайдених під час приймання, і оголошується бета-реліз. Під час бета-релізу продукт тестується замовниками або викладається у відкритий доступ, якщо він призначений для масових користувачів.

Етап 5. Розгортання та технічна підтримка. По закінченні фази тестування програмного продукту приходить час релізу (розгортання) та передачі його замовникам. Після закінчення бета-тестування оголошується комерційний реліз продукту, і він надходить у продаж. Одночасно починається маркетингова діяльність за поданням нового продукту на ринку. Відділ технічної підтримки забезпечує зворотний зв'язок з користувачами, їх консультування та підтримку.

Відповідно до етапів циклу розробки програмного забезпечення побудовано імітаційну модель, що відтворює специфіку просування проекту в динаміці. У модель не увійшов етап підтримки, тому що він є безперервним і виконується розробниками програмного забезпечення. Етап аналізу вимог увійшов до етапу планування.

Як програмна платформа використано систему багатопідходного імітаційного моделювання

AnyLogic [26]. Модель реалізована з використанням дискретно-подієвого та агентного підходів. Обрання дискретно-подієвої методології обумовлене необхідністю значної деталізації процесів розробки програмного забезпечення. Агентна методологія забезпечує відтворення динамічного процесу просування програмного продукту/проекту по етапах розробки. Як динамічний елемент продукт/проект може мати індивідуальні властивості, що дає можливість урахувати специфічні особливості продуктів, що замовлені споживачами – клієнтами ІТ-фірми.

Далі модель наведено в узагальненому, спрощеному вигляді. Одиницею модельного часу обрано тиждень (оскільки ітерація в ідеології SCRUM складає 2 тижні). У процесах, втім, використовується період в 1 годину, оскільки їх тривалість можна задати саме в цих одиницях. Слід, проте, мати на увазі, що модельний час легко налаштовується під потреби конкретних користувачів.

Загальний вигляд моделі у графічному редакторі AnyLogic наведено на *рис. 1*. Змінні, що використані в імітаційній моделі, наведено на *рис. 2*. Призначення головних модельних елементів визначено в *табл. 1*.

Агентом у моделі визначається програмний продукт/проект (*Feature*).

Проект в моделі має три параметри:

- 1) складність (*Difficulty*). Тип параметра – int, значення за замовчуванням – 1. Складність проекту в процесі розробки впливає на час його упровадження;
- 2) пріоритет (*Priority*). Тип параметра – int, значення за замовчуванням – 1. Пріоритет проекту визначає проектний менеджер команди. Пріоритет впливає на те, скільки часу пройде від моменту створення проекту до взяття його в роботу та втілення;
- 3) тип (*isAbug*). Тип параметру – Boolean, значення за замовчуванням – false. Цей параметр визначає, чи пов'язані з проектом критичні баги (помилки системи) або нові (притаманні даному проекту) функції. Критичні помилки необхідно одразу ж ліквідувати, тому агенти з параметром *isAbug=true* беруться в роботу без проходження блоків дизайну та фронтенд-розробки.

Нові ж функції потребують планування, дизайну та розробки верстки: тому вони проходять багато етапів, перш ніж потрапити до блоку впровадження. Перш за все, здійснюється планування та розробка вимог.

На стадії планування визначається пріоритет проекту, за яким він буде просуватися далі в процесі розробки. Необхідність дизайну не є безперечною, тому ця фаза потребує додаткової перевірки згідно з вимогами користувача. Якщо дизайн потрібен, за-

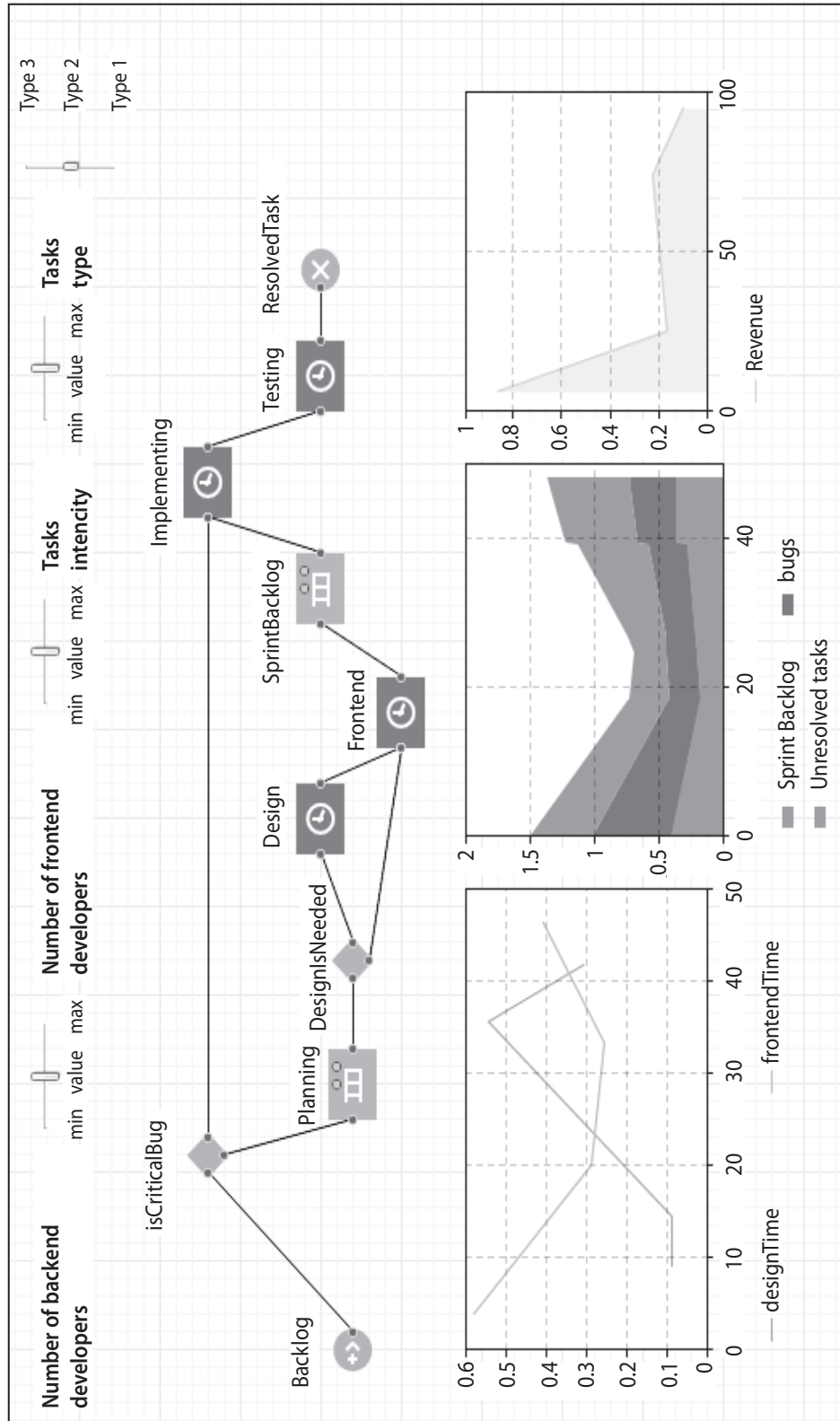


Рис. 1. Імітаційна модель процесу розробки програмного забезпечення

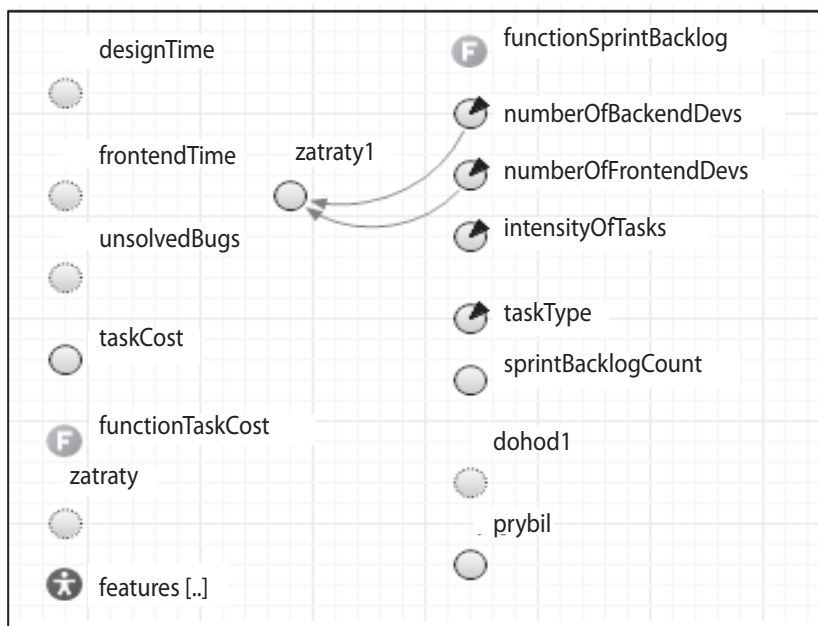


Рис. 2. Змінні, що використані в імітаційній моделі

Таблиця 1

Елементи моделі в AnyLogic

Найменування елемента	Тип елемента	Призначення елемента
1	2	3
Backlog	Блок типу Source	Генерує проекти (агенти Feature). Проекти прибувають 1 раз на день. Інтенсивність прибуття агентів задана за допомогою формули: $Agents\ per\ arrival = uniform_discr(2,3) * intensityOfTasks$
uniform_discr(2,3)	Генератор випадкових чисел	Значення у даному випадку – за добу генерується від 2 до 3 проектів
intensityOfTasks	Параметр	Інтенсивність виникнення проектів, яка регулюється за допомогою бігунка Intensity
taskType	Параметр	Тип складності проекту
isCriticalBug	Блок типу SelectOutput	Перевірка, чи пов'язані з проектом критичні помилки або нові функції. Направляє вхідних агентів в один із двох вихідних портів залежно від виконання заданої за допомогою ймовірностей умови. Код на виході True: $agent.isABug = true$; $unsolvedBugs = unsolvedBugs + 1$
unsolvedBugs	Динамічна змінна	Накопичувач кількості помилок
Planning	Блок типу Queue	Формування черги згідно з пріоритетом агента (проекту). Пріоритет визначається випадково: наприклад, $agent.Priority = uniform_discr(1,5)$
DesignIsNeeded	Блок типу SelectOutput	Розгалуження процесу обробки за заданою ймовірністю
Design	Блок типу Delay	Визначає часову затримку на стадії дизайну: $Delaytime = uniform_discr(5,8) * agent.Difficulty$ На вході визначається складність проекту для дизайнера: $agent.Difficulty = uniform_discr(1,5) * taskType$ На виході збирається статистика щодо середнього часу розробки дизайну: $designTime = Design.getElapsedTime(agent)$
designTime	Динамічна змінна	Збирає статистику стосовно середньої тривалості фази дизайну на кожному кроці моделювання

1	2	3
Frontend	Блок типу Delay	Визначає часову затримку на стадії фронтенд-розробки: Delaytime = uniform_discr(2,5)*agent.Difficulty На вході визначається складність проєкту для фронтенд-програміста: agent.Difficulty=uniform_discr(2,7) * taskType На виході збирається статистика щодо середнього часу фронтенд-розробки: frontendTime=Frontend.getElapsedTime(agent)
frontendTime	Динамічна змінна	Збирає статистику стосовно середньої тривалості фази фронтенд-розробки на кожному кроці моделювання
numberOfFrontendDevs	Параметр	Чисельність фронтенд-розробників компанії
SprintBacklog	Блок типу Queue	Формування черги до бекенд-розробників згідно з пріоритетом проєктів. Пріоритет визначається випадково: наприклад, agent.Priority=uniform_discr(1,5)
sprintBacklogCount	Динамічна змінна	Збір статистики щодо вмісту черги до бекенд-розробників. Визначається за викликом функції: sprintBacklogCount=functionSprintBacklog()
functionSprintBacklog	Функція	Тіло функції: intsprintBacklog=SprintBacklog.size(); returnsprintBacklog;
Implementing	Блок типу Delay	Визначає часову затримку на стадії бекенд-розробки: Delaytime=uniform_discr(6,10)*agent.Difficulty На вході визначається складність проєкту для бекенд-розробника: agent.Difficulty=uniform_discr(1,7) * taskType
numberOfBackendDevs	Параметр	Чисельність бекенд-розробників компанії
Testing	Блок типу Delay	Визначає часову затримку на етапі тестування: Delaytime=uniform_discr(1,4)*agent.Difficulty На вході визначається складність проєкту для тестування: agent.Difficulty=uniform_discr(1,3) * taskType На виході з блоку реалізується такий алгоритм (збір статистики щодо критичних помилок): if (agent.isAbug==true); {unsolvedBugs=unsolvedBugs-1;}
unsolvedBugs	Динамічна змінна	Лічильник критичних помилок
ResolvedTask	Блок типу Sink	Завершення процесу розробки проєкту із обчисленням основних кількісних і вартісних показників його реалізації

лежно від складності нового функціоналу дизайнеру необхідний деякий час для створення макета. У наведеному прикладі приймається ймовірний середній час від 5 до 8 годин (у моделі збирається та накопичується відповідна статистика).

Етап 6 – фронтенд-розробка: в моделі враховується складність проєкту для фронтенд-програміста. Залежно від складності нового функціонала фронтенд-розробникам необхідний час для створення верстки функціонала нового проєкту за дизайнерським макетом (у даному прикладі ймовірний час становить від 2 до 5 годин). Максимально на даній стадії може знаходитись одночасно стільки проєктів, скільки в компанії фронтенд-розробників. У моделі чисельність фронтенд-розробників регулюється динамічно.

Після розробки дизайну та створення верстки проєкти стають у чергу до бекенд-розробників, які будуть втілювати новий функціонал «у життя». У моделі збирається статистика стосовно вмісту черги

протягом періоду моделювання. Водночас на етапі бекенд-розробки ліквідуються критичні помилки, знайдені в коді. Залежно від складності розробники витрачають на цьому етапі від 6 до 10 годин (як приклад). Кількість проєктів, що одночасно знаходяться на стадії бекенд-розробки, залежить від чисельності наявних співробітників даного спрямування. У процесі експериментування цей параметр налаштовується в динаміці.

Останнім, **7 етапом**, є тестування проєкту, яке, за наведеними умовами, займає від 1 до 4 годин залежно від його складності. Чисельність тестувальників також може налаштовуватися протягом експериментів динамічно. Окремо в моделі розраховується статистика ліквідації критичних помилок системи. Близькість значення лічильника до 0 означає меншу кількість незадоволених клієнтів, які не можуть користуватися програмним забезпеченням при наявності критичних помилок у системі.

Після завершення тестування виконані проекти або ліквідовані критичні помилки виходять у публічний доступ (у моделі агент Feature знижується).

У моделі для завершених проектів розраховуються вартісні показники – заробітна плата розробників за всіма етапами, дохід від реалізації програмного продукту та прибуток фірми. Враховуються різні типи складності проектів. Загальні вартісні показники розраховуються також за вказані користувачем часові звітні періоди (місяць та ін.) та згідно з термінами моделювання. Гнучке налаштування може здійснюватися безпосередньо в процесі проведення експериментів залежно від конкретних потреб менеджерів фірми.

Роботу моделі проілюструємо фрагментами результатів імітаційних експериментів, проведених за даними типової продуктової ІТ-компанії (обрані умовні вхідні дані зі збереженням реальної тенденції розгортання процесів у часі). Результати базового експерименту наведено на рис. 3, де тут і надалі:

- ✦ графік а) – «Динаміка середнього часу на дизайн і фронтенд-розробку»;
- ✦ графік б) – «Динаміка кількості проектів у бекенд-черзі, невідрементованих критичних помилок і необроблених проектів»;
- ✦ графік в) – «Динаміка накопичення прибутку компанії».

Забезпечення ефективного проходження процесу розробки програмних продуктів передбачає оперативні дії від менеджерів проектів, яким, своєю чергою, необхідні дані стосовно динаміки конкретних параметрів. Одними з найбільш важливих є часові параметри знаходження проектів на трудомістких етапах процесу створення продукту – дизайн і фронтенд-розробка. У результаті здійснення стандартних імітаційних експериментів на моделі формується динаміка середнього часу перебування проекту в дизайнера та фронтенд-програміста. На графіку а) вісь ординат – час на розробку в тижнях, вісь абсцис – одиниці модельного часу впродовж терміну імітації (дані формуються у змінних designTime та frontendTime).

Отримана динаміка свідчить про більшу стабільність роботи дизайнерів, тоді як у фронтенд-розробників середній час коливається за значно більшою амплітудою. Деякий вплив на це робить ступінь складності конкретних проектів, але, зважаючи на усереднення часу, – менеджерам треба приділити більшу увагу чисельності та кваліфікації співробітників на даній стадії.

Важливими параметрами в процесі управління є також вміст черги до бекенд-розробників (кількість проектів у блоці SprintBacklog згідно з пріоритетом), кількість необроблених проектів (unresolvedtasks) і невідрементованих критичних помилок (unresolvedbugs). На графіку б) вісь ординат – кількість проектів відповідного типу, вісь абсцис – одиниці модельного часу.

Наведені результати свідчать про зростання черги до бекенд-розробки тільки наприкінці розглянутого періоду. Розширення періоду імітації може більше прояснити картину. З точки зору менеджменту фірми має викликати занепокоєння збільшення невиконаних проектів, особливо стрімке зростання якого спостерігається наприкінці досліджуваного періоду.

Динаміка накопичення прибутку ІТ-компанії (вміст змінної *Revenue*) представлена на графіку в), де вісь *y* – прибуток компанії в доларах США, вісь *x* – одиниці модельного часу.

Проілюструємо фрагментарно деякі можливості впливу менеджменту на підвищення ефективності управління проектами розробки програмного забезпечення.

Експеримент 1 (рис. 3).

Оцінка загальних ресурсів фірми згідно з інтенсивністю надходження проектів-замовлень. Регулюється інтенсивність надходження замовлень (модельний елемент управління – повзунк *Tasksintensity*) при незмінних інших параметрах.

Одиниця модельного часу – тиждень. Досліджуваний інтервал часу – 50 тижнів (приблизно 1 рік).

З 10-ї по 30-ту одиницю модельного часу при регулюванні повзунка в максимальне положення інтенсивність створення агентів (проектів) була у два рази вища, ніж за замовчуванням. На графіку можна спостерігати різке збільшення кількості невітлених проектів (змінна *Unresolvedtasks*) і навіть збільшення кількості критичних помилок (змінна *bugs*). Це не є бажаним для команди результатом. Але, з іншого боку, після зменшення інтенсивності створення агентів до мінімуму (після 30-го тижня) можна спостерігати, що розробники не тільки встигають з виконанням робіт, але й часто простоюють.

За отриманими даними, за рік було втілено близько 1450 проектів. Як можна спостерігати, інтенсивність створення проектів значною мірою впливає на те, чи встигає команда розробників справлятися з їх втіленням. Але важливим завданням менеджменту є визначення оптимального співвідношення фронтенд- і бекенд-розробників.

Експеримент 2 (рис. 4).

Кількість бекенд- і фронтенд-розробників регулюється за допомогою елементів управління – повзунків *Number of frontend devs* і *Number of backend devs*. Необхідно проаналізувати, яке оптимальне співвідношення двох типів розробників сприятиме тому, що кількість проектів у *Sprint Backlog* (проекти, що готові до реалізації) та нереалізованих проектів (*Unsolved tasks*) співвідноситься приблизно як 1 : 2 – 1 : 3. Таке співвідношення проектів буде найбільш сприятливим для команди розробки та забезпечуватиме повною зайнятістю як розробників, так і дизайнерів та тестувальників.

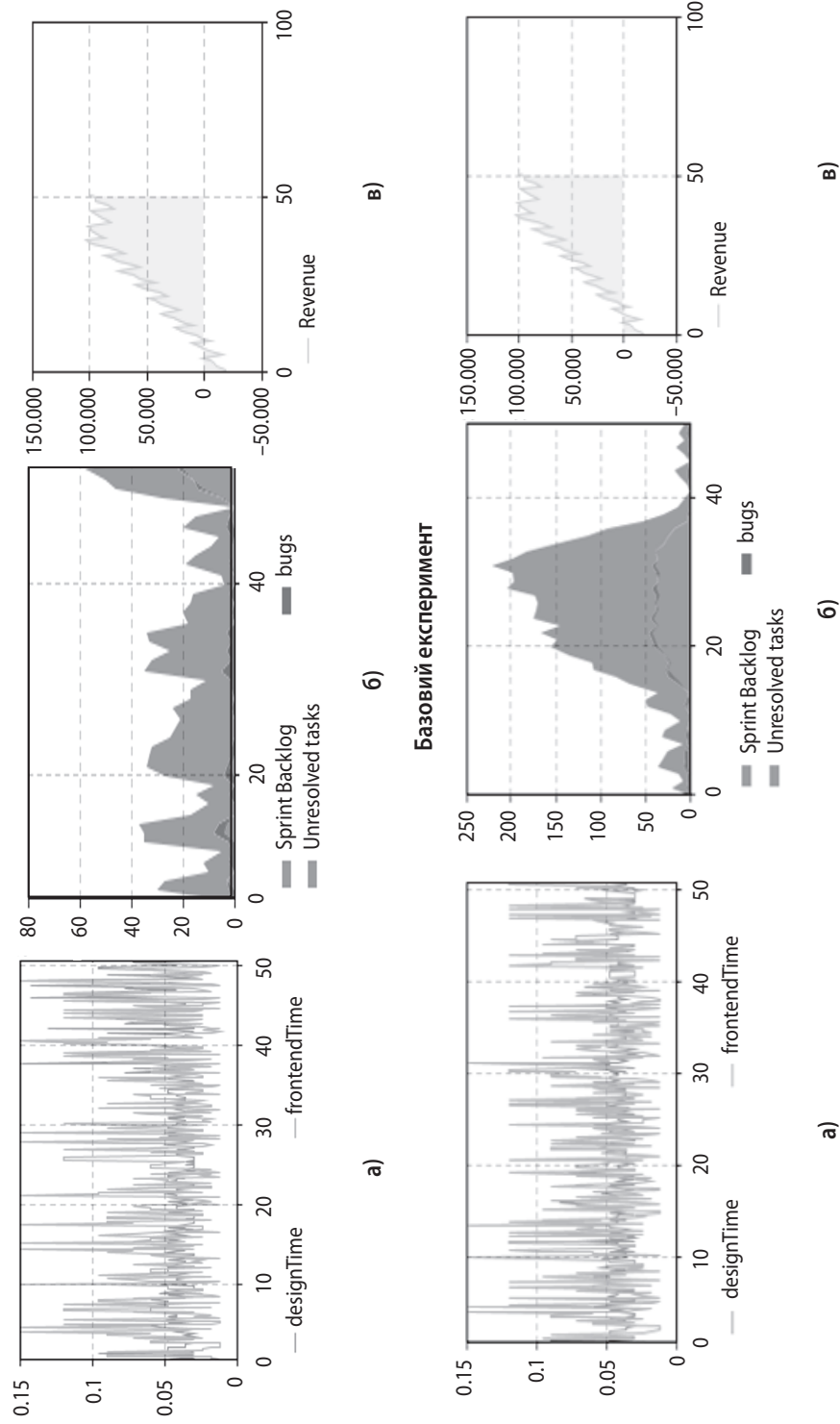


Рис. 3. Результати експериментів: базового та експерименту 1

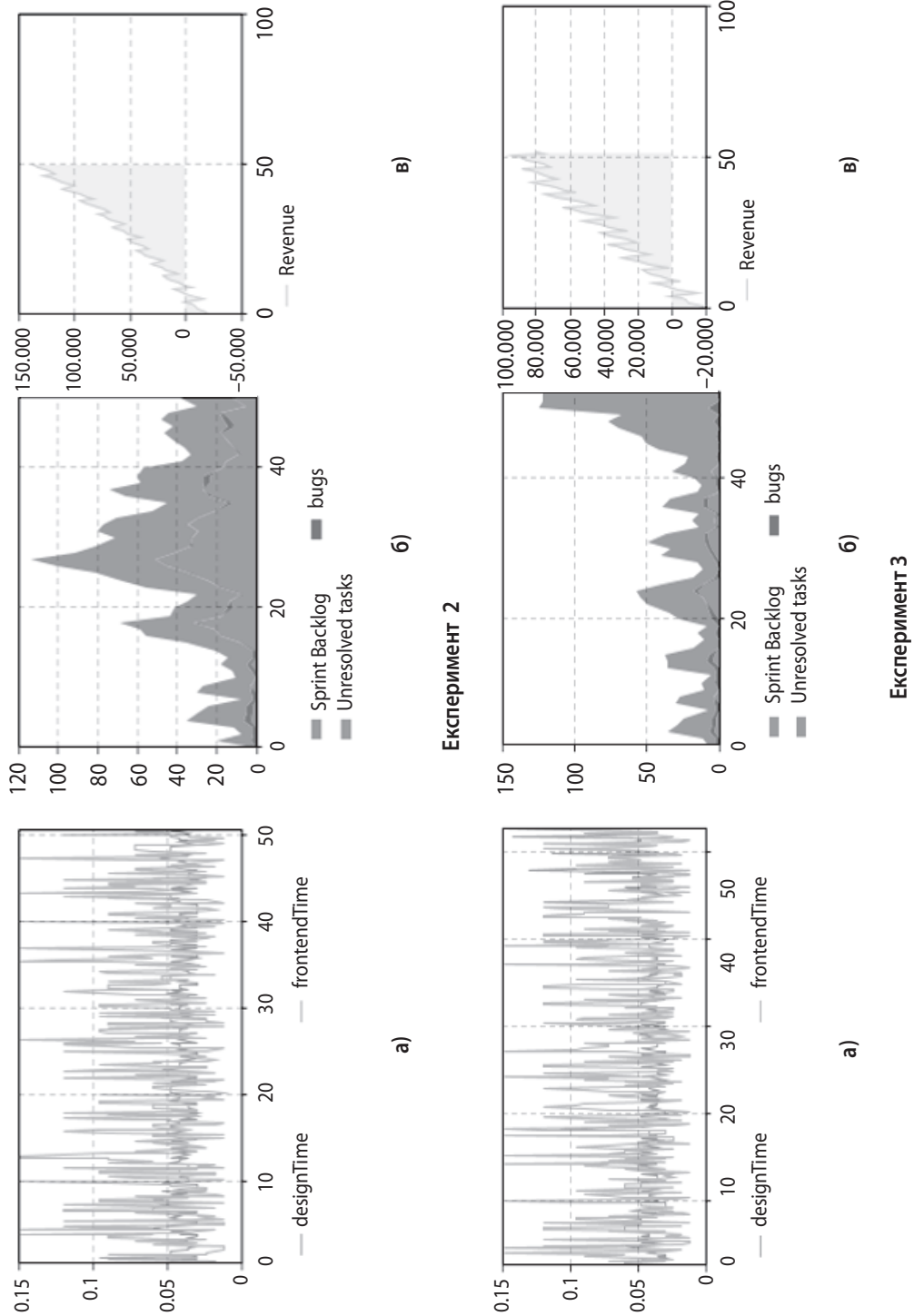


Рис. 4. Результати експериментів 2-3

За результатами експерименту можна спостерігати, що оптимальне співвідношення проектів досягається за наявності 3 фронтенд-розробників та 6 бекенд-розробників. Отже, менеджерам треба замислитися про збільшення штату фронтенд-розробників, тому що саме від них залежить кількість проектів, що потрапляє у SprintBacklog.

За наведеним сценарієм, за рік втілено приблизно 1400 проектів.

У наступному експерименті треба дослідити, місткість якого саме блоку – Frontend чи Backend – впливатиме на кількість проектів найсильніше.

Експеримент 3 (рис. 4).

Результати регулювання тільки місткості стадії фронтенд-розробки – бігунка Number of frontend developers – при незмінних інших параметрах. На проміжку 0–20 модельних одиниць часу кількість фронтенд-розробників дорівнювала 2, на проміжку 20–40 – 3, далі – 1. Як можемо спостерігати за графіком, збільшення фронтенд-програмістів спричиняє збільшення проектів у Sprint Backlog і позитивно впливає на кількість необроблених проектів. Однак при зменшенні фронтенд-розробників до 1 спостерігається різке зростання кількості незакінчених проектів, що не є бажаним результатом для фірми.

Експеримент 4 (рис. 5).

Вплив кількості бекенд-розробників на динаміку оброблених проектів за допомогою регулювання бігунка Number of backend developers при незмінних інших параметрах.

На проміжку 0–20 модельних одиниць часу кількість бекенд-розробників дорівнювала 10, на проміжку 20–40 – 15, далі – 5. На графіку можна спостерігати таку тенденцію: збільшення кількості бекенд-розробників майже зводить до нуля кількість проектів у Sprint Backlog, а зменшення діє на це у протилежний бік. Однак кардинальна зміна чисельності бекенд-програмістів не стала причиною такої ж кардинальної зміни кількості проектів порівняно з попередньо дослідженою кореляцією фронтенд-програмістів. До того ж, прийняття на роботу більшої кількості програмістів наведеного спрямування негативно впливає на прибутки організації, тому що зарплатня в них найбільша.

Отже, менеджмент ІТ-компанії може зробити відповідні висновки про необхідність прийняття на роботу кожного з двох типів розробників.

Змінна Task type впливає на складність розробки та на прибуток, який отримує компанія за втілення проекту цього типу.

Експеримент 5 (рис. 5).

Дослідимо вплив зміни складності проекту (динамічне налаштування елемента управління Task Type) на кількість втілених проектів та на прибуток

компанії. Значення бігунка на проміжку 0–20 складало 1, далі – 2 і 3 (складність збільшувалася). Тобто, після зміни типу проекту розробники почали встигати менше, але, з іншого боку, прибуток компанії продовжував зростати. Проте динаміка невиконаних проектів також була позитивною.

Керівник команди розробки може зробити висновок про те, що необхідно втілювати проекти більшої вартості, проте в жодному разі не можна забувати про кількість невиконаних проектів, яка невинно зростає, адже проекти третього типу дуже трудомісткі.

Завдяки тому, що в середовище системи AnyLogic вбудовано оптимізатор OptQuest, можливе проведення не тільки стандартних, але й оптимізаційних експериментів. Оптимізація складається з декількох послідовних прогонів моделі з різними значеннями параметрів. Комбінуючи евристичні, нейронні мережі та математичну оптимізацію, OptQuest дозволяє знаходити значення параметрів моделі, відповідні максимуму або мінімуму цільової функції, як в умовах невизначеності, так і за наявності обмежень.

На рис. 6 наведено результати двох типів експериментів:

1. Максимізація кількості розроблених ІТ-компанією програмних продуктів (проектів – Resolved tasks) при варіації параметрів чисельності фронтенд-розробників (Number Of Frontend Devs), бекенд-розробників (Number Of Backend Devs); типів складності проектів (task Type); інтенсивності появи замовлень, виникнення проектів (Intensity Of Tasks).
2. Оптимізація прибутку ІТ-компанії за варіацією наведених у пункті 1 параметрів.

Тривалість експериментів – 52 тижні.

Отримані результати першого експерименту доводять, що найбільшу кількість проектів можна реалізувати за наявності 14 бекенд- і 3 фронтенд-розробників. Водночас, якщо проекти першого (найменшого) типу складності будуть надходити з подвоєною інтенсивністю, буде реалізовано 1761 проектів на рік.

Однак менеджмент компанії повинен проаналізувати реальну доцільність такого організаційного рішення, зважаючи на найвищий існуючий рівень заробітної плати саме бекенд-розробників та порівнюючи динаміку цих виплат з можливим прибутком (для прогнозованої оптимальної кількості виконаних проектів). Можливо, результати стандартного експерименту, наведені вище, будуть більш бажаними для керівників, тому що 3 фронтенд-розробники та 6 бекенд-розробників реалізують приблизно 1400 проектів, але заробітної платні доведеться заплатити значно менше.

За результатами другого експерименту було визначено, що для отримання найбільшого прибутку компанії за рік при обмеженні невиконаних проектів у 150 штук необхідно 10 бекенд-програмістів, 2 фронтенд-

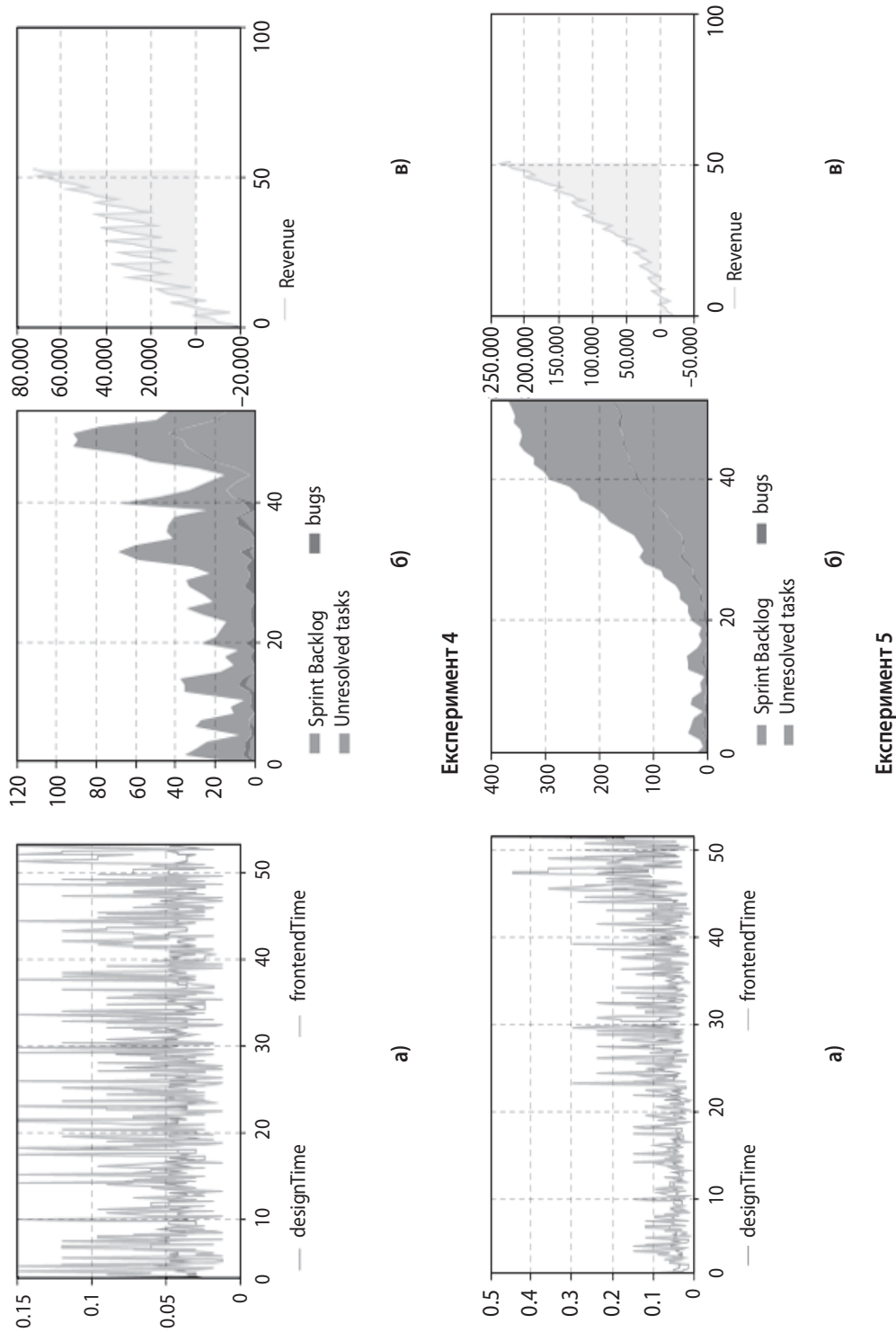
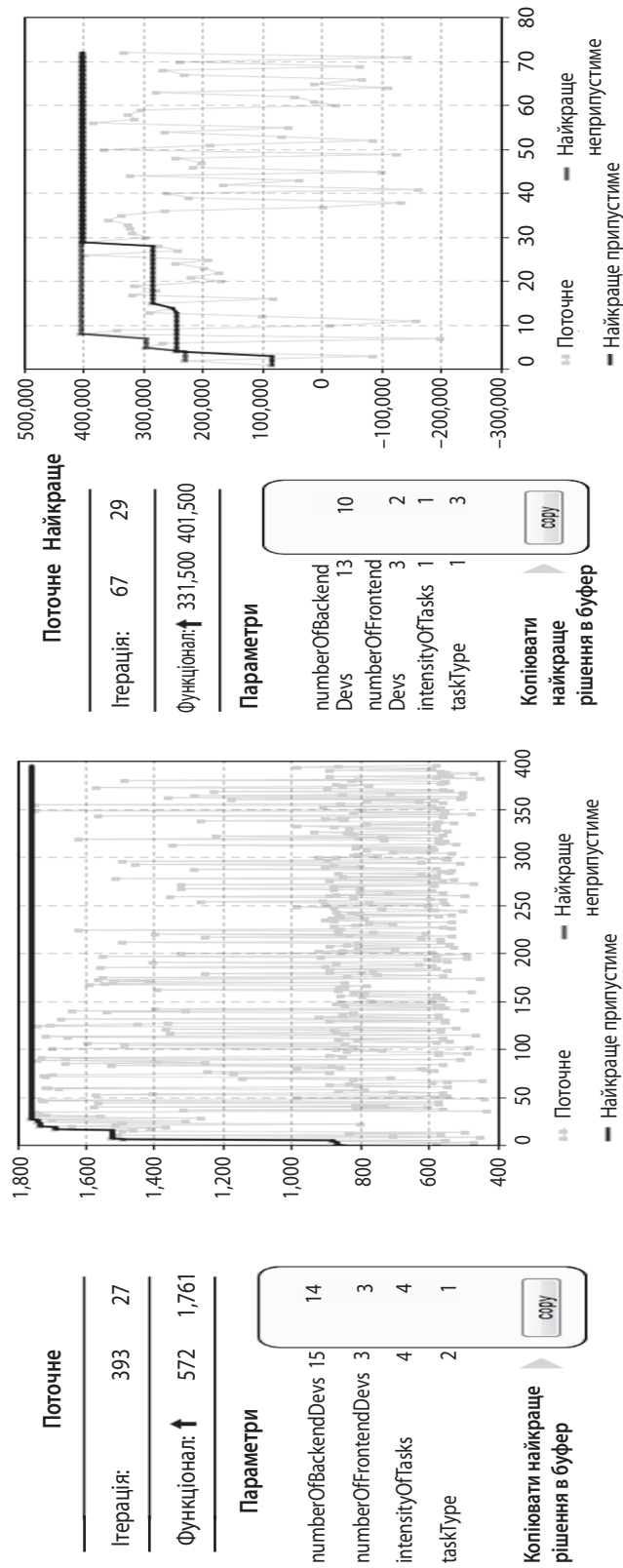


Рис. 5. Результати експериментів 4-5



Оптимізаційний експеримент 1 – максимізація кількості розроблених програмних продуктів

Оптимізаційний експеримент 2 – максимізація прибутку компанії

Рис. 6. Результати оптимізаційних експериментів

тенд-програмісти, інтенсивність надходження проєктів має складати 1, а виконувати необхідно проєкти третього типу, попри те, що вони дуже трудомісткі. При такому співвідношенні показників прибуток компанії за рік становитиме \$401500.

ВИСНОВКИ

Проведені стандартні й оптимізаційні експерименти спрямовані на забезпечення платформи прийняття управлінських рішень керівником команди розробки проєктів ІТ-компанії. Наведена модель є інструментом наочної демонстрації поведінки команди розробки при внесенні змін і дозволяє підвищити ефективність використання людських ресурсів компанії. Як ілюстрація, результати експериментів показали, що для отримання більшого прибутку доцільно виконувати проєкти найвищого типу складності, наймаючи навіть не максимальну чисельність програмістів.

Модель є достатньо типовим тренажером для управління розробкою проєктів продуктових ІТ-компаній завдяки модульності, відкритості та можливості гнучкого параметричного налаштування. Дружній інтерфейс користувача, легке пристосування до специфіки конкретних ІТ-фірм, налаштування на різноманітні типи експериментів (з великої лінійки експериментальної підтримки платформи AnyLogic) сприяють позитивним прогнозам прикладного впровадження запропонованого імітаційного модельного інструменту. ■

ЛІТЕРАТУРА

1. Software Development Outsourcing to Ukraine: 9 Reasons Why It's the Best Choice. URL: <https://www.aimprosoft.com/blog/it-outsourcing-ukraine/>
2. Барская И. С., Тесленко П. А., Денисенко В. Ю. Влияние заказчика на распределение трудовых ресурсов ИТ-проекта. *Вісник НТУ «ХПІ». Серія «Стратегічне управління, управління портфелями, програмами та проєктами»*. 2015. № 2. С. 56–60.
3. Галицин В. К., Суслов О. П., Самченко Н. К. Концепція моніторингу функціонування ІТ-фірми. *East European Scientific Journal*. 2016. Nr. 6 (2). S. 86–94.
4. Инструментальные средства моделирования систем в информационной экономике : монография / под ред. В. С. Пономаренко, Т. С. Клебановой. Харьков, 2019. 452 с.
5. Кравченко Т. К., Гоменюк К. С., Дружаев А. А., Липатников А. Ю. Планирование бизнес-анализа в компании с применением методологии Agile. *Прикладная информатика*. 2019. Т. 14. № 5. С. 5–17. DOI: 10.24411/1993-8314-2019-10032
6. Максимов К. В. Планирование деятельности ИТ-компании в условиях неопределенности с учетом использования облачных сервисов. *Прикладная информатика*. 2019. Т. 13. № 1. С. 25–31.
7. Моделирование процессов управления в информационной экономике : монография / под ред.

- В. С. Пономаренко, Т. С. Клебановой. Бердянск : Издатель Ткачук А. В., 2017. 420 с.
8. Саяпин О. В., Быстракова И. А., Тиханыхев О. В., Чискидов С. В. Разработка интерфейсов прикладных программ: макетирование или прототипирование. *Прикладная информатика*. 2020. Т. 18. № 1. С. 47–56. DOI: 10.24411/1993-8314-2020-10004
9. Сравнительный анализ теоретических моделей каскадных, итеративных и гибридных подходов к управлению жизненным циклом ИТ-проектов / Первухин Д. В., Исаев Е. А., Рытиков Г. О. и др. *Бизнес-информатика*. 2020. Т. 14. № 1. С. 32–40. DOI: 10.17323/2587-814X.2020.1.32.40
10. Яценко Р. М., Баликов О. Г. Комплекс моделей управления бизнес-процессами сервисной ИТ-компанії. *Бізнес Інформ*. 2017. № 10. С. 191–197. URL: https://www.business-inform.net/export_pdf/business-inform-2017-10_0-pages-191_197.pdf
11. Teslenko P., Bedrii D., Antoshchuk S., Lytvynchenko H. 3-level approach to the projects planning // 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018. Proceedings, 2018. P. 195–198.
12. Teslenko P., Polshakov I. Strategic management of evolving project-oriented organization. *Science and Education a New Dimension. Economics*. 2016. Vol. IV (2). Issue 94. P. 33–35. URL: https://seanewdim.com/uploads/3/4/5/1/34511564/p._teslenko_i._%D0%A0olshakov_d._bedrii_strategic_management_of_evolutionary_project-oriented_organization.pdf
13. Yatsenko R., Balykov O. Optimization of a service IT company's business processes functional structure. *Development Management*. 2019. Vol. 17. Is. 1. P. 35–50. DOI: 10.21511/dm.5(1).2019.04
14. Математичне та комп'ютерне моделювання економічних процесів : монографія / за заг. ред. З. М. Соколовської. Одеса : Астропринт, 2016. 272 с.
15. Митрошин С. Г. Многоподходные имитационные модели в производственных процессах информационно-технологических компаний : дис. ... канд. техн. наук : 05.13.17. Пенза, 2015. 192 с.
16. Соколовська З. М. Імітаційне моделювання бізнес-процесів складних економічних систем. *Праці Одеського політехнічного університету*. 2011. Вип. 3. С. 135–141.
17. Бугаевский Т. Имитационная модель бизнес-процессов компании-разработчика ПО по модели SCRUM. URL: http://zkinteractive.com/management/soft_co_bm/ru/
18. Соколовська З. М., Яценко Н. В., Хортюк М. В. Імітаційні моделі діяльності ІТ-фірм на платформі AnyLogic. *Бізнес Інформ*. 2019. № 6. С. 61–76. DOI: <https://doi.org/10.32983/2222-4459-2019-6-61-76>
19. Сотникова А. В. Разработка моделей и метода распределения трудовых ресурсов в управлении реализацией портфеля ИТ-проектов : дис. ... канд. экон. наук : 08.00.13. М., 2016. 142 с.
20. Sturrock D. T. Avoid Failures! Tested Success Tips for Simulation Project Excellence // Proceedings of the 2018 Winter Simulation Conference. P. 252–260. URL: <https://www.informs-sim.org/wsc18papers/includes/files/023.pdf>

21. Wagner G. Business Process Modeling and Simulation with DPMN: Resource-Constrained Activities // Proceedings of the 2020 Winter Simulation Conference. P. 45–59. URL: <https://informs-sim.org/wsc20papers/004.pdf>
 22. Martin N., Depaire B., Caris A. A Synthesized Method for Conducting a Business Process Simulation Study // Proceedings of the 2018 Winter Simulation Conference. P. 276–290. URL: <https://www.informs-sim.org/wsc18papers/includes/files/025.pdf>
 23. Suslov S., Katalevsky D. Modeling and Simulation Toolset // Evolving Toolbox for Complex Project Management. *Auerbach Publications*, 2019. Chapter 19. P. 411–440.
 24. Warnke T., Uhrmacher A. M. Complex Simulation Experiments Made Easy // Proceedings of the 2018 Winter Simulation Conference. P. 410–424. URL: <https://www.informs-sim.org/wsc18papers/includes/files/034.pdf>
 25. Allen N. B. S-FLOW: Methodology of Applying Discrete-Event Simulation // Proceedings of the 2018 Winter Simulation Conference. P. 4107–4108. URL: <https://www.informs-sim.org/wsc18papers/includes/files/375.pdf>
 26. AnyLogic (2020). URL: <http://www.anylogic.ru/>
- REFERENCES**
- Allen, N. B. "S-FLOW: Methodology of Applying Discrete-Event Simulation". *Proceedings of the 2018 Winter Simulation Conference*. <https://www.informs-sim.org/wsc18papers/includes/files/375.pdf>
- AnyLogic (2020). <http://www.anylogic.ru/>
- Barskaya, I. S., Teslenko, P. A., and Denisenko, V. Yu. "Vliyaniye zakazchika na raspredeleniye trudovykh resursov IT-proekta" [The Influence of the Customer on the Distribution of Labor Resources of the IT Project]. *Visnyk NTU «KhPI». Seriya «Stratehichne upravlinnia, upravlinnia portfeliamy, prohramamy ta proektamy»*, no. 2 (2015): 56-60.
- Bugayevskiy, T. "Imitatsionnaya model biznes-protsessov kompanii-razrabotchika PO po modeli SCRUM" [A Simulation Model of Business Processes of a Software Developer Based on the SCRUM Model]. http://zkinteractive.com/management/soft_co_bm/ru/
- Halitsyn, V. K., Suslov, O. P., and Samchenko, N. K. "Konseptsiia monitorynhu funktsionuvannia IT-firmy" [The Concept of Monitoring of IT Company]. *East European Scientific Journal*, no. 6(2) (2016): 86-94.
- Instrumentalnyye sredstva modelirovaniya sistem v informatsionnoy ekonomike* [Tools for Modeling Systems in the Information Economy]. Kharkiv, 2019.
- Kravchenko, T. K. et al. "Planirovaniye biznes-analiza v kompanii s primeneniyem metodologii Agile" [Planning Business Analysis in a Company Using Methodology Agile]. *Prikladnaya informatika*, vol. 14, no. 5 (2019): 5-17.
DOI: 10.24411/1993-8314-2019-10032
- Maksimov, K. V. "Planirovaniye deyatelnosti IT-kompanii v usloviyakh neopredelennosti s uchetom ispolzovaniya oblachnykh servisov" [Planning of Activities in the IT-company in Conditions of Uncertainty Taking into Account the Use of Cloud Services]. *Prikladnaya informatika*, vol. 13, no. 1 (2019): 25-31.
- Martin, N., Depaire, B., and Caris, A. "A Synthesized Method for Conducting a Business Process Simulation Study". *Proceedings of the 2018 Winter Simulation Conference*. <https://www.informs-sim.org/wsc18papers/includes/files/025.pdf>
- Matematychnye ta kompiuterne modeliuvannia ekonomichnykh protsesiv* [Mathematical and Computer Modeling of Economic Processes]. Odesa: Astroprint, 2016.
- Mitroshin, S. G. "Mnogopodkhodnyye imitatsionnyye modeli v proizvodstvennykh protsessakh informatcionno-tehnologicheskikh kompaniy" [Multi-Approach Simulation Models in Manufacturing Processes of Information Technology Companies]: *dis. .. kand. tekhn. nauk* : 05.13.17, 2015.
- Modelirovaniye protsessov upravleniya v informatsionnoy ekonomike* [Modeling Management Processes in the Information Economy]. Berdiansk: Izdatel Tkachuk A. V., 2017.
- Pervukhin, D. V. et al. "Srvnitelnyy analiz teoreticheskikh modeley kaskadnykh, iterativnykh i gibridnykh podkhodov k upravleniyu zhiznennym tsiklom IT-proektov" [Theoretical Comparative Analysis of Cascading, Iterative, and Hybrid Approaches to IT Project Life Cycle Management]. *Biznes-informatika*, vol. 14, no. 1 (2020): 32-40.
DOI: 10.17323/2587-814X.2020.1.32.40
- "Software Development Outsourcing to Ukraine: 9 Reasons Why It's the Best Choice". <https://www.aimprosoft.com/blog/it-outsourcing-ukraine/>
- Sayapin, O. V. et al. "Razrabotka interfeysov prikladnykh programm: maketirovaniye ili prototipirovaniye" [Development of Application Program Interfaces: Layout or Prototype]. *Prikladnaya informatika*, vol. 18, no. 1 (2020): 47-56.
DOI: 10.24411/1993-8314-2020-10004
- Sokolovska, Z. M. "Imitatsiine modeliuvannia biznes-protsesiv skladnykh ekonomichnykh system" [Simulation Modeling of Business Processes of Complex Economic Systems]. *Pratsi Odeskoho politekhnichnoho universytetu*, no. 3 (2011): 135-141.
- Sokolovska, Z. M., Yatsenko, N. V., and Khortiuk, M. V. "Imitatsiini modeli diialnosti IT-firm na platformi AnyLogic" [The Simulation Models of Activities of IT Firms on the Basis of AnyLogic Platform]. *Biznes Inform*, no. 6 (2019): 61-76.
DOI: <https://doi.org/10.32983/2222-4459-2019-6-61-76>
- Sotnikova, A. V. "Razrabotka modeley i metoda raspredeleniya trudovykh resursov v upravlenii realizatsiyyey portfelya IT-proektov" [Development of Models and Methods for Allocating Labor Resources in Managing the Implementation of a Portfolio of IT Projects]: *dis. ... kand. ekon. nauk* : 08.00.13, 2016.
- Sturrock, D. T. "Avoid Failures! Tested Success Tips for Simulation Project Excellence". *Proceedings of the 2018 Winter Simulation Conference*. <https://www.informs-sim.org/wsc18papers/includes/files/023.pdf>
- Suslov, S., and Katalevsky, D. "Modeling and Simulation Toolset". In *Evolving Toolbox for Complex Project Management*, chapter 19, 411-440. *Auerbach Publications*, 2019.
- Teslenko, P. et al. "3-level approach to the projects planning". *2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies*, 2018. 195-198.

Teslenko, P., and Polshakov, I. "Strategic management of evolving project-oriented organization". *Science and Education a New Dimension. Economics*. 2016. https://seanewdim.com/uploads/3/4/5/1/34511564/p._teslenko_i._%D0%A0olshakov_d._bedrii_strategic_management_of_evolutionary_project-oriented_organization.pdf

Wagner, G. "Business Process Modeling and Simulation with DPMN: Resource-Constrained Activities". *Proceedings of the 2020 Winter Simulation Conference*. <https://informs-sim.org/wsc20papers/004.pdf>

Warnke, T., and Uhrmacher, A. M. "Complex Simulation Experiments Made Easy". *Proceedings of the 2018 Winter*

Simulation Conference. <https://www.informs-sim.org/wsc18papers/includes/files/034.pdf>

Yatsenko, R. M., and Balykov, O. H. "Kompleks modelei upravlinnia biznes-protsesamy servisnoi IT-kompanii" [Complex of Business Process Management Models for a Service-Providing IT Company]. *Biznes Inform*, no. 10 (2017): 191-197. https://www.business-inform.net/export_pdf/business-inform-2017-10_0-pages-191_197.pdf

Yatsenko, R., and Balykov, O. "Optimization of a service IT company's business processes functional structure". *Development Management*, vol. 17, no. 1 (2019): 35-50. DOI: 10.21511/dm.5(1).2019.04

УДК 519:86:331.556
JEL: C63; F22; J61; O15

МОДЕЛЮВАННЯ ВПЛИВУ ІНТЕНСИФІКАЦІЇ МІГРАЦІЙНИХ ПРОЦЕСІВ НА СОЦІАЛЬНО-ЕКОНОМІЧНУ СТАБІЛЬНІСТЬ В УКРАЇНІ

©2021 ЛУК'ЯНЕНКО І. Г., НОВІК А. Ю.

УДК 519:86:331.556
JEL: C63; F22; J61; O15

Лук'яненко І. Г., Новік А. Ю. Моделювання впливу інтенсифікації міграційних процесів на соціально-економічну стабільність в Україні

Метою статті є розробка адекватного математичного інструментарію формування міграційних потоків для оцінювання їх впливу на соціально-економічну стабільність і детінізацію української економіки, а також визначення ефективних заходів державної політики, спрямованих на їх регулювання для запобігання ризикам соціально-економічної дестабілізації. У статті проведено емпіричний аналіз особливостей розвитку міграційних процесів в Україні та їх впливу на соціально-економічний стан у середньо- та довгостроковій перспективах. Розроблено систему взаємодоповнювальних динамічних макромоделей економіки України, що включає векторну авторегресійну модель та окремий комплекс імітаційних підмоделей системної динаміки, спрямованих на дослідження міграційних процесів та їх впливу на показники соціально-економічної стабільності. При цьому розроблені моделі системної динаміки можуть об'єднуватись як в єдину оригінальну імітаційну макромодель України, так і використовуватись самостійно залежно від цілей дослідження. Це дозволяє адекватно описати складні нелінійні процеси, системно визначити та дослідити механізми формування попиту та пропозиції на ринку праці з урахуванням тіньової складової та оцінити їх вплив на формування величини міграційних потоків, а також дослідити наслідки посилення інтенсивності міграційних процесів на соціально-економічну стабільність в Україні. Реалізація розробленого комплексу динамічних моделей дозволила, зокрема, розрахувати щорічне значення чистого міграційного потоку в Україні за допомогою ланцюга старіння населення; емпірично підтвердити, що міграційні потоки в Україні є недооцінені; виявити динамічність структури чистого сальдо міграції та помітне згасання амплітуди процесів міграції з часом, що свідчить про наявність балансувальної структури формування міграційних потоків.

Ключові слова: міграція, ринок праці, соціально-економічна стабільність, моделювання міграційних процесів, векторні авторегресійні моделі міграції, системна динаміка, тіньова економіка.

DOI: <https://doi.org/10.32983/2222-4459-2021-5-123-133>

Рис.: 7. Табл.: 1. Бібл.: 22.

Лук'яненко Ірина Григорівна – доктор економічних наук, професор, завідувачка кафедри фінансів, Національний університет «Кієво-Могилянська академія» (вул. Г. Сковороди, 2, Київ, 04655, Україна)

E-mail: iryna.lukianenko@ukma.edu.ua

ORCID: <http://orcid.org/0000-0002-4128-5909>

Researcher ID: <https://publons.com/researcher/1930228/iryna-lukianenko/>

Scopus Author ID: <https://www.scopus.com/authid/detail.uri?authorid=57189348551>

Новік Аліна Юріївна – кандидат економічних наук, старший викладач кафедри фінансів, Національний університет «Кієво-Могилянська академія» (вул. Г. Сковороди, 2, Київ, 04655, Україна)

E-mail: novikaj@ukma.edu.ua

ORCID: <https://orcid.org/0000-0001-7139-2214>

Researcher ID: <https://publons.com/researcher/3355714/alina-novik/>

UDC 519:86:331.556
JEL: C63; F22; J61; O15

Lukianenko I. G., Novik A. Yu. Modeling the Impact of Intensification of Migration Processes on the Socio-Economic Stability in Ukraine

The article is aimed at elaborating an adequate mathematical instrumentarium for the formation of migration flows to assess their impact on the socio-economic stability together with unshadowing the Ukrainian economy, as well as to identify efficient measures of the State-based policy directed towards regulating these flows to prevent risks of socio-economic destabilization. The article carries out an empirical analysis of the peculiarities of the development of migration processes in Ukraine and their impact on the socio-economic state in both the medium and the long term perspective. A system of complementary