

**Ю.И. Волков, Н.М. Войналович***Кировоградський державний педагогічний університет  
імені Володимира Винниченка***ПЕРРИН-ПОДОБНЫЕ РЕКУРРЕНТНЫЕ ПОСЛЕДОВАТЕЛЬНОСТИ**

Рассматривается множество последовательностей  $\{y_n\}$ , которые задаются рекуррентностями

$$y_{n+m} = ay_{n+r} + by_n, m \geq 2, (a, b, r) \in \{(1,1,1); (1,1, m-1); (-1,1,1); (-1,1, m-1)\}.$$

Мы получили эффективные формулы для  $y_n$  и их возможные применения в тестах простоты.

**Ключевые слова:** рекуррентности, последовательность Перрина, простые числа, производящая функция..

**Yrii Volkov, Nataliya Vojnalovich***The Kirovograd Volodymyr Vynychenko state pedagogical university***PERRIN-LIKE RECURRING SEQUENCES**

Consider the set of sequences  $\{y_n\}$  give be recurrences

$$y_{n+m} = ay_{n+r} + by_n, m \geq 2, (a, b, r) \in \{(1,1,1); (1,1, m-1); (-1,1,1); (-1,1, m-1)\}.$$

We get the effectie formulae for  $y_n$  and their possible use in primality tests.

**Keywords:** recurrences, Perrin sequences, prime numbers, generating function.

**ВІДОМОСТІ ПРО АВТОРІВ**

**Войналович Наталія Михайлівна** – доцент кафедри математики, доцент, кандидат педагогічних наук.

*Коло наукових інтересів:* методика навчання математики, дискретна математика.

**Волков Юрій Іванович** – професор кафедри математики, професор, доктор фізико-математичних наук.

*Коло наукових інтересів:* математичний аналіз, теорія ймовірностей і математична статистика, дискретна математика.

УДК 004.4+378.2

**В.В. Концедайло***Житомирський державний університет імені Івана Франка***ВИКОРИСТАННЯ СЕРЙОЗНИХ ІГОР ТА СИМУЛЯЦІЙ З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗВИТКУ НЕТЕХНІЧНИХ КОМПЕТЕНТНОСТЕЙ МАЙБУТНІХ ІНЖЕНЕРІВ-ПРОГРАМІСТІВ**

У статті розглянуто використання серйозних ігор та симуляцій з розробки програмного забезпечення для розвитку нетехнічних компетентностей майбутніх інженерів-програмістів. З'ясовано, що типовій освіті інженерів-програмістів бракує практичного опанування процесів розробки програмного забезпечення. Це пов'язано з тим, що розробку програмного забезпечення, а, особливо, великих систем програмного забезпечення, прийнято вважати суто технічним завданням, проте на практиці виявляється інакше: у більшості випадків розробка великих (і не тільки) систем є завданням, здебільшого, "нетехнічного" характеру. Виявлено, що єдиним можливим способом надання студентам досвіду участі у реальних процесах розробки програмного забезпечення (ПЗ) в академічному середовищі є використання ігрових симуляторів та симуляцій у поєднанні з лекціями і навчальними проектами. Доведено, що симулятори можуть принести в освіту інженерів-програмістів ту ж користь, яку вони принесли і у інші галузі (медицина, авіація та інші). Зокрема,

*Йдеться про те, що процес навчання та підготовки інженерів-програмістів може бути поліпшений та покращений за умови надання можливості студентам практикуватися за допомогою симуляторів в управлінні різними видами псевдо-реалістичних процесів розробки програмного забезпечення. Проведено літературний огляд робіт починаючи з 2000 року, опублікованих за темою навчання розробки ПЗ із використанням серйозних ігор та ігрових симуляцій. Розглянуто вище зазначені серйозні ігри та ігрові симуляції більш детально, за розподілом по формуванню нетехнічних компетентностей майбутніх інженерів-програмістів.*

**Ключові слова:** серйозні ігри; симуляції; інформаційно-комунікаційні технології (ІКТ); інженери-програмісти; нетехнічні компетентності.

**Вступ.** Десятки років тому відомий американський інженер-програміст та вчений-інформатик Фредерік Брукс (Frederick Brooks) у своєму класичному творі під назвою "Легендарний людино-місяць" (The Mythical Man-Month) написав: "Найважчою складовою процесу побудови програмної системи є прийняття однозначного рішення про те, що саме необхідно побудувати. Жодна з інших складових роботи над концепцією не представляє собою таку складність, як визначення детальних технічних вимог, включаючи всі аспекти зіткнення продукту з людьми, машинами та іншими програмними системами. Жодна інша складова цієї роботи не може завдати такої значної шкоди отриманій у результаті системі, якщо дана складова виконана неправильно. Саме цю складову процесу розробки найважче виправити на більш пізніх етапах" [7].

Розробку програмного забезпечення, а, особливо, великих систем програмного забезпечення, прийнято вважати суто технічним завданням. На практиці виявляється інакше: у більшості випадків розробка великих (і не тільки) систем є завданням, здебільшого, "нетехнічного" характеру. Тобто визначальними є аналіз та формулювання вимог до програмного забезпечення, управління розробкою програмного забезпечення (ПЗ), методології, а також моделі та методи розробки ПЗ. Це пов'язано з тим, що планування і аналіз мають надзвичайно важливе значення при розробці ПЗ. Ті, хто брав участь у реальних проектах погоджуються, що погане планування є найвизначнішою причиною провалів проектів [11]. Другим критичним фактором, що призводить до невдачі, є недостатнє розуміння потреб користувачів ПЗ [4]. Так, комунікація та управління є важливими у проектах будь-яких сфер, та особливого значення вони набувають у проектах саме з розробки ПЗ. Управління проектами програмного забезпечення означає: в першу чергу, збір усіх необхідних даних, тримаючи персонал добре поінформованими, а також координацію індивідуальних потреб і завдань із загальними цілями.

**Постановка проблеми.** Типовій освіті інженерів-програмістів бракує практичного опанування процесів розробки програмного забезпечення. Зазвичай, студентам представлені лише відповідні теорії процесів розробки ПЗ у лекціях, а можливості втілити в життя ці концепції на практиці у відповідних дисциплінах доволі обмежені [6].

**Аналіз досліджень і публікацій у досліджуваній області.** У той час, як більшість підходів до навчання майбутніх інженерів-програмістів спрямовані на додавання реалізму у практичні заняття в аудиторії, деякі автори (М. Баррос, А. Бейкер, С. Вернер, А. Дантас, Е. Наварро, А. Хук) стверджують, що єдиним можливим способом надання студентам досвіду участі у реальних процесах розробки ПЗ в академічному середовищі є використання ігрових симуляторів та симуляцій у поєднанні з лекціями і навчальними проектами. І хоча вище згадані підходи розрізняються з точки зору процесів, що вони імітують, і їх конкретних цілей, всі вони розроблені з метою дозволити студентам краще практикуватись і брати участь у процесах розробки ПЗ у більшому масштабі і більш швидким способом, ніж це

може бути досягнуто на основі фактичних проєктів [6].

**Метою нашої статті** є аналіз використання серйозних ігор та симуляцій з розробки програмного забезпечення для розвитку нетехнічних компетентностей майбутніх інженерів-програмістів.

**Виклад основного матеріалу.** Розглянемо детальніше ідею надання студентам досвіду участі у реальних процесах розробки ПЗ в академічному середовищі за допомогою використання ігрових симуляторів та симуляцій. Проте, по-перше, важливо дати визначення таким поняттям, як ігрові симулятори (або так звані серйозні ігри) та симуляції.

Відсутність чіткого визначення симуляції та ігор може привести до того, що деякі вчені називають "термінологічна неоднозначність" [8]. В останні роки багато керівників, педагогів та практиків звернули свою увагу на можливості використання симуляцій та ігор в освіті. Крім того, у наукових колах використання симуляцій та ігор в освіті є, на даний момент, досить поширеною темою.

Сучасні дослідження ігор та їх використання у навчальному процесі оперують наступними визначеннями: освітні ігри, серйозні ігри, навчання на основі електронних ігор або прикладні ігри.

Незважаючи на відсутність загальних визначень і термінології, автори, як правило, зосереджені на іграх не у контексті дозвілля, а у контексті здобуття серйозного досвіду, умінь та навичок. Визначення симуляцій та ігор сприяє термінологічній узгодженості і дозволяє уникнути двозначності [8].

Історично симуляції відносяться як до симуляцій в управлінні/бізнесі, так і до комп'ютерних симуляцій [8]. В цілому, симуляції – це моделі, що виражають складні реальні системи. Симуляції використовуються для аналізу конкретних систем, моделей розвитку учнів та студентів, а також для дослідження штучних (віртуальних) середовищ [12].

Згідно дослідження Л. Саува [10], гра, у її формальному визначенні, це цілеспрямована конкуруюча діяльність, що включає ту чи іншу форму конфлікту (конфлікт представляється у вигляді будь-якої перешкоди, що не дозволяє гравцю з легкістю досягти цілі гри) та проводиться у рамках певних узгоджених правил. У грі бере участь щонайменше одна окрема особа (гравець) або група осіб (гравці), яким у контексті даної гри необхідно приймати певні рішення задля досягнення поставленої цілі гри.

Різниця між іграми та симуляціями також полягає у їх меті: метою ігор та ігрових симуляторів є залучення та заохочення гравців за допомогою веселого і цікавого досвіду, тоді як мета симуляторів є підготовка і розвиток навичок своїх користувачів. Відповідно до дослідження В. Нараянасамі є два різних види симуляцій: тренувальні симуляції та моделювальні симуляції. Тренувальні симуляції імітують процеси реального світу шляхом відтворення певної системи або процесу з метою забезпечення максимальної ефективності та підвищення продуктивності користувача. Одним із прикладів є симулятор керування літаком "FlightGear Flight Simulator" (1997).

На відміну від тренувальних симуляцій, моделювальні симуляції – це симуляції, що моделюють певні системи з метою створення та/або випробування певної моделі: такі як, наприклад, симуляції погоди або симуляції автомобілей. Оскільки класифікації симуляцій та ігор частіш за все неоднозначні, а їх межі залежать від області застосування, варто підкреслити, що запропоновані класифікації для даних термінів слід розглядати як взаємопов'язанні категорії, а не як абсолютно окремі поняття [13].

Відповідно до дослідження Е. Наварро [6], симулятори являють собою надзвичайно

потужний освітній інструмент, який зазвичай використовується у навчальному процесі у випадках, коли реальна практика не є можливою або не є доступною. Таким чином, у роботах Е. Наварро висувається та обґрунтовується гіпотеза про те, що симулятори можуть принести у освіту інженерів-програмістів ту ж користь, яку вони принесли і у інші галузі (медицина, авіація та інші) [6]. Зокрема, йдеться про те, що процес навчання та підготовки інженерів-програмістів може бути поліпшений та покращений за умови надання можливості студентам практикуватися за допомогою симуляторів в управлінні різними видами псевдо-реалістичних процесів розробки програмного забезпечення [6].

І хоча будь-яка класифікація симуляцій та ігор є дискусійним питанням, їх визначення має важливе значення під час обговорення їх ефективності в академічній освіті. У даному дослідженні ми поєднуємо теорії від В. Нараянасамі [12], Дж. Ліна [9], Т. Епперлей [3], щоб класифікувати різні типи симуляцій та ігор [13]. Відповідно до зазначених робіт можна запропонувати наступну класифікацію, що представлена у таблиці 1.

Таблиця 1.

**Класифікація серйозних ігор, симуляційних ігор та симуляцій відповідно до робіт В. Нараянасамі [12], Дж. Ліна [9] та Т. Епперлей [3]**

Вид	Жанр	Короткий опис	Представники
Ігри	Рольові ігри	Наслідування персонажів	"The Sims"
	Стратегії	Прийняття стратегічних рішень	"Company of Heroes" (реального часу) "Civilization" (покрокова)
	Екшен	Активні виклики Гра на координацію	"Space Invader"
Гібриди	Симуляційні ігри	Цілеспрямоване відтворення реальних процесів	"Gazillionaire"
Симуляції	Тренувальні симуляції	Тренування для досягнення максимальної продуктивності у виконанні певних завдань	Авіа-симулятор
	Моделювальні симуляції	Моделювання певних процесів або об'єктів	Симуляція погодних умов Моделювання автомобілей

Відповідно до праць декількох авторів: С. Колфілд, С. Мей, Дж. Ся та Д. Віл [5], існує також декілька так званих "серйозних ігор" або ігрових симуляцій в області навчання та підготовки майбутніх інженерів-програмістів, що викладені у їх систематичному огляді літератури [5].

Серйозні ігри можуть допомагати в якості засобу для набуття досвіду, а також, враховуючи їх привабливий характер, і для мотивації студентів. Крім того, серйозні ігри, що базуються на симуляціях, дозволяють у процесі навчання брати участь у реальних сценаріях у середовищі без ризиків [1].

У свою чергу, у 2015 році Р. Атал і А. Сурека провели літературний огляд робіт дослідників з 2000 року до 2013 року, опублікованих за темою навчання розробки ПЗ із використанням концепції моделювання гри. Разом вище перераховані автори виділили наступний перелік робіт [1; 5]:

- "ANUKARNA" – гра-симулятор для підготовки студентів по передовій практиці експертної оцінки коду;
- "Ameise" – управління проектом розробки програмного забезпечення (з акцентом на якість ПЗ);
- "PRODEC" – управління програмними проектами;
- "DELIVER" – управління отриманою вартістю;
- "Simsoft" – програмне забезпечення для управління проектами у навчальній програмі;
- "ProMaSi", "SESAM" – управління проектами;
- "SimVBSE" – розробка ПЗ на основі цінності;
- "Problems and Programmers", "SimjavaSP", "SimSE" – процеси розробки ПЗ;
- "Incredible Manager" – емпіричне управління проектами.

Розглянемо вище зазначені серйозні ігри та ігрові симуляції більш детально, за розподілом по формуванню нетехнічних компетентностей майбутніх інженерів-програмістів.

**SESAM** (Software Engineering Simulation by Animated Models) створений А. Драппа (A. Drappa) та Дж. Людвіг (J. Ludewig) є одним із перших симуляторів, розроблених для освітніх цілей у 2000 році. Являє собою середовище моделювання процесу розробки програмного забезпечення, в якому студенти керують командою віртуальних співробітників для виконання віртуального проекту за графіком, в рамках бюджету, а також на рівні або вище необхідного рівня якості. Даний симулятор використовує дуже гнучку і виразну мову, але процес побудови моделі є трудомістким, вимагає певного навчання та написання коду у текстовому редакторі. **SESAM** – це перший приклад мови моделювання процесу програмного забезпечення, який є розпорядчим, прогнозованим та інтерактивним (але не графічним) [2].

У 2004 році Е. Наварро (E. Navarro) було розроблено **SimSE** – інтерактивний, графічний та гнучкий у налаштуванні симулятор процесу розробки програмного забезпечення для освітніх цілей. **SimSE** намагається вирішити проблему відсутності вивчення процесу розробки програмного забезпечення у типовому курсі програмної інженерії, надаючи студентам практичний досвід реалістичного процесу розробки програмного забезпечення в привабливій манері. Це дозволяє студентам практикувати "віртуальний" процес розробки програмного забезпечення у повністю графічному, інтерактивному та веселому середовищі, де прямий і зворотній зв'язок у графічному вигляді дозволяє їм вивчити складні причинно-наслідкові зв'язки, що лежать в основі процесів програмної інженерії.

У цьому ж році інший симулятор **The Incredible Manager** був введений Марсіо Барросом (Marcio Barros), Александре Дантасом (Alexandre Dantas) та Клаудією Вернер (Claudia Werner). Це гра-симулятор, розроблений спеціально для підготовки менеджерів проектів програмного забезпечення. Він більшою мірою зосереджений на управлінні проектами, ніж на процесі розробки програмного забезпечення. По суті, це промисловий симулятор з додатковим графічним користувацьким інтерфейсом, подібним до ігрового. **The**

**Incredible Manager** дозволяє налаштовувати й імітаційні моделі через свій текстовий інтерфейс.

**SimjavaSP** – це інтерактивний, веб-інтерфейсний, графічний симулятор процесу розробки програмного забезпечення, який з'явився у кінці 2005 року. Метою для студента у SimjavaSP є, діючи в якості менеджера проекту, розробити заданий проект програмного забезпечення протягом заданого часу та, використовуючи заданий бюджет, досягти при цьому відповідну заплановану якість. Користувацький інтерфейс для SimjavaSP являє собою поєднання графічного та текстового зворотного зв'язку.

**ProMaSi** (Project Management Simulator) розроблений Н. Петалідісом (N. Petalidis) дозволяє студентам активно перевіряти свої навички в управлінні розробкою програмного забезпечення за допомогою виконання відповідних керуючих дій у симуляторі. Варто зазначити, що одним з нових підходів, запропонованих у даному симуляторі, є поділ складових частин симулятора на архітектурні модулі, що дозволяє третім сторонам створювати та додавати свої власні доробки.

У 2013 році А. Болліном (A. Bollin) було розроблено **AMEISE** (A Media Education Initiative for Software Engineering) – гру-симулятор, де студенти беруть на себе роль технічного керівника проекту. У ході симуляції вони стикаються з труднощами, що виникають у процесі керування проектом відповідно до конкретної моделі завдання, обраної викладачем. Інструктор або викладач має можливість вибрати кількість випробувань (запусків моделювання) для вирішення поставлених завдань у рамках заданих обмежень. Студенти можуть вчитися на основі своїх попередніх запусків симулятора, змінювати свої стратегії та вимірювати свій власний успіх за допомогою функції самооцінки.

**PRODEC**: серйозна гра-симулятор для підготовки менеджерів розробки програмного забезпечення, розроблена Алехандро Кальдероном, Мерседес Руїсом. Автори зазначають, що хоча є деякі роботи, пов'язані із застосуванням серйозних ігор для навчання управління програмними проектами, на даний момент не вистачає інструментів, які поєднують у собі навчання і оцінювання в одному інструменті, а також які забезпечують для студентів умови експериментувати у прийнятті рішень в умовах, що близькі до реальних. Project Decision (PRODEC) – це серйозна гра-симулятор, створена з метою підготовки та оцінювання студентів у галузі розробки та управління програмними проектами. Основна мета полягає в тому, щоб, використовуючи привабливий характер гри, розмістити учнів у віртуальній організації, де вони можуть управляти програмними проектами і вирішувати реальні проблеми у середовищі без ризиків.

Для викладачів, PRODEC – це інструмент, що допомагає розглядати такі питання, як керівництво, менеджмент завдання та команди, моніторинг проектів, контроль та управління ризиками. Також даний симулятор допомагає викладачам оцінити навички та знання, які учні розвивають та здобувають, граючи у дану гру-симулятор. Після будь-якої спроби, PRODEC пропонує повний звіт, що представляє кожне рішення, яке гравці зробили, а також результат відповідно до застосованих критеріїв оцінки, заданих викладачем на початку гри.

**Simsoft** (Caulfield, Veal, & Maj, 2011b) – симулятор, розроблений з метою побачити, як серйозні ігри та симулятори можуть вплинути на якість підготовки майбутніх інженерів-програмістів і менеджерів проектів. Simsoft поставляється у двох частинах. По-перше, це роздруковане ігрове поле розміру А0, навколо якого гравці збираються разом, щоб обговорити поточний стан свого проекту і розглянути питання про їх наступний крок. Поле показує хід гри, в той час як пластикові лічильники використовуються для індикації

кількості співробітників проекту. Покерні фішки являють собою бюджет команди, за допомогою якого вона може придбати більше співробітників, а також звідки кошти можуть забиратися в залежності від рішень, прийнятих в ході гри. Існує також простий Java додаток, за допомогою якого гравці можуть побачити поточний і історичний стан проекту у серії звітів та повідомлень, а також де гравці можуть налаштувати параметри проекту. Програмною основою Simsoft є модель динаміки системи, яка втілює в собі невеликий набір фундаментальних причинно-наслідкових зв'язків доволі простих проектів розробки програмного забезпечення. В ході симуляції Simsoft команди студентів, практикуючих менеджерів проектів та інженери-програмісти керують гіпотетичним проектом розробки програмного забезпечення з метою завершення проекту в термін і в рамках бюджету. На основі вхідного сценарію гри, а також інформації, що надається під час гри, та власного досвіду гравці приймають рішення про те, як слід діяти: наймати більше співробітників або зменшити їхню кількість, скільки часу витратити на розробку і тд.

У процесі використання розглянутих вище стимуляторів у навчальному процесі, формуються наступні нетехнічні компетентності:

1. **Адаптивність** – у ході симуляції гравець повинен швидко адаптуватися до змін і розглядати нові підходи.
2. **Вирішення проблем** – гравцю необхідно виявляти проблеми, а також використовувати логіку, здоровий глузд та наявні дані для оцінки альтернативних рішень досягнення поставленої мети та результату.
3. **Збір та аналіз даних** – у ході симуляції гравцю необхідно шукати, збирати та синтезувати дані та знання, і у неупередженій манері приходити до висновків, мети або суджень, що служать базою для прийняття рішень і формування стратегії.
4. **Звітність** – гравець бере на себе відповідальність за успішне виконання поставлених цілей та досягнення результатів, встановлюючи високі стандарти роботи для себе та для віртуальних співробітників.
5. **Орієнтація на кінцевий результат** – гравцю необхідно фокусуватись на бажаному результаті, ставити складні стимулюючі цілі та досягати їх.
6. **Знання правил та процедур** – гравець повинен розуміти та застосовувати знання про положення, правила і процедури гри та процесу розробки програмного забезпечення.
7. **Зовнішня та організаційна поінформованість** – для успішного проходження симуляції гравець повинен визначати і розуміти, як внутрішні та зовнішні тенденції (наприклад, економічні, політичні, соціальні тощо) впливають на роботу організації.
8. **Планування та пріоритезація** – у ході симуляції гравець планує та організовує трудову діяльність, а також повинен бути здатний керувати декількома завданнями одночасно.
9. **Політична кмітливість** – гравець повинен проявляти впевненість та професійну дипломатію, і у той же час налагоджувати ефективні відносини із людьми на всіх рівнях, всередині та зовні організації.
10. **Прийняття рішень** – для успішного проходження симуляції гравцю необхідно здобувати необхідні знання, уміння та навички, визначати ключові питання та наслідки для прийняття обґрунтованих та об'єктивних рішень.
11. **Робота в команді** – гравцю необхідно працювати разом з іншими віртуальними співробітниками та сприяти досягненню спільних поставлених цілей.

12. **Співпраця** – у ході симуляції гравець працює спільно з іншими, всередині та за межами організації, для досягнення цілей, що ведуть до створення і підтримки взаємовигідних партнерських відносин, максимально ефективного використання знань та досягнення відповідних результатів.

13. **Стійкість** – для успішного проходження симуляції гравцю необхідно зберігати високу продуктивність та самоконтроль під тиском або під час негараздів.

14. **Увага до дрібниць** – у ході симуляції гравець повинен гарантувати повноту та точність даних і знань, а також слідкувати, щоб домовленості та зобов'язання були виконані.

15. **Управління змінами** – гравцю необхідно розуміти необхідність змін, планувати та пристосуватись до них настільки творчо і позитивно, наскільки це можливо.

16. **Якісний / кількісний аналіз** – для успішного проходження симуляції гравцю необхідно аналізувати та оцінювати наявні дані для керування та досягнення необхідних результатів.

У 2006 році **SimVBSE** симулятор був спеціально розроблений А. Джейном (A. Jain), щоб навчити студентів теорії вартості в основах програмної інженерії. SimVBSE має повністю графічний користувацький інтерфейс, що включає також анімацію та аудіо. Симулятор відтворює один конкретний приклад з реального світу та не включає в себе опції налаштування.

При використанні даного стимулятора у навчальному процесі, формуються аналогічно до попереднього випадку наступні компетентності: адаптивність; вирішення проблем; збір та аналіз даних; звітність; орієнтація на кінцевий результат; планування та пріоритизація; політична кмітливість; прийняття рішень; стійкість; увага до дрібниць; якісний / кількісний аналіз.

**ANUKARNA**: симулятор процесу розробки програмного забезпечення для практичного навчання основ прийняття рішень у процесі виконання експертної оцінки коду (Peer Code Review) розроблений Рітіка Аталом (Ritika Atal). Атал розробив веб-інтерактивну освітню гру-симулятор процесу розробки програмного забезпечення для пояснення та викладання переваг і передових практик процесу експертної оцінки коду (Peer Code Review). Атал описує основи та модель навчання, що засновані на "навчанні у процесі відкриття", навчанні на помилках, доказах та міркуваннях щодо викладання основних концепцій практики експертної оцінки коду. Також Атал оцінює запропоновані основи, модель та симулятор шляхом проведення експериментів та збору зворотного зв'язку від користувачів, а також представляє отримані результати.

При використанні даного стимулятора у навчальному процесі, формуються аналогічно наступні компетентності: вирішення проблем; збір та аналіз даних; знання правил та процедур; орієнтація на кінцевий результат; прийняття рішень; робота в команді; співпраця; стійкість; увага до дрібниць; управління змінами; якісний / кількісний аналіз.

**Висновки.** Даний огляд ігрових симуляторів та симуляцій, що використовуються у процесі підготовки майбутніх інженерів-програмістів для формування нетехнічних компетентностей, показує, що ігрові симулятори та симуляції, як педагогічні засоби та ІКТ, стають все більш поширеними, особливо в Європі та Америці. Студентам у цілому подобається грати та навчатися у такий спосіб, так як вони отримують цінний досвід, що дуже близький до реального. Тим не менш, деякі з ігрових симуляторів та симуляцій були розроблені поза межами їх початкових реалізацій, що свідчить про те, що їх педагогічна цінність не була продемонстрована у достатній мірі. Виходячи з цих даних, є відповідні



наслідки для дослідників, викладачів та розробників ігрових симуляторів та симуляцій, а саме: необхідність більш детального дослідження ефективності ігрових симуляторів та симуляцій у формуванні нетехнічних компетентностей майбутніх інженерів-програмістів. Тим не менш, наведено достатньо доказів того, що педагоги повинні розглянути можливість використання ігрових симуляторів та симуляцій у рамках своїх курсів з програмної інженерії.

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Alejandro Calderón. Bringing real-life practice in software project management training through a simulation-based serious game / A. Calderón, M. Ruiz. – CSEDU, Proceedings of the 6th International Conference on Computer Supported Education. – 2014. – P. 117-124.
2. ANUKARNA : A Software Engineering Simulation Game for Teaching Practical Decision Making in Peer Code Review
3. Apperley, T. H. (2006). Genre and Game Studies: Toward a Critical Approach to Video Game Genres. *Simulation & Gaming*, Vol. 37:1 (2009 ) 6 – 23
4. Balci, O. Validation, Verification, and Testing Techniques throughout the Life Cycle of a Simulation Study. *Annals of Operations Research* 33, (1994), 121-173.
5. Craig Caulfield. A Systematic Survey of Games Used for Software Engineering Education / Caulfield, C., Xia, J. (Cecilia), Veal, D., & Maj, S. P. – *Modern Applied Science*, 5(6). – 2011. – P. 28-43
6. Emily Navarro. SimSE: A Software Engineering Simulation Environment for Software Process Education / Emily Navarro – Irvine, CA: University of California, Irvine. – 2006.
7. Frederick P. Brooks, *The Mythical Man-Month: Essays on Software Engineering*, Anniversary Edition (2nd Edition) by Frederick P. Brooks , ISBN: 0201835959
8. Klabbers, J. H. G.: Terminological Ambiguity Game and Simulation. *Simulation & Gaming* Vol. 40:4 (2009) 446 – 463
9. Lean, J. , Moizer, J., Towler, M., Abbey, C.: Simulations and Games . Use and Barriers in Higher Education. *Active Learning in Higher Education* , Vol. 7: 3 ( 2006) 227 – 242]
10. Louise Sauvé. Games and simulations - Theoretical underpinnings / L. Sauve, L. Renaud, D. Kaufman. – Proceedings of the Digital Games Research Association Conference, Vancouver, B.C. – 2005.]
11. Metzger, P.W., Boddie, J. *Managing a Programming Project* (3rd Ed.). Prentice Hall, Englewood Cliffs, New Jersey, 1996.
12. Narayanasamy, W., Wong, K. K., Fung, C. C., Rai, S.: Distinguishing Games and Simulation Games from Simulators . *Comput. Entertain.*, Vol. 4:2 (2006)
13. Stephanie de Smale. The Effect of Simulations and Games on Learning Objectives in Tertiary Education: A Systematic Review / Stephanie de Smale. Tom Overmans, Johan Jeuring, Liesbeth van de Grint. – GALA. – 2015. – P. 506-516

*Valerii V. Kotsedailo*

*Zhytomyr Ivan Franko State University*

### USING SERIOUS GAMES AND SIMULATIONS OF SOFTWARE DEVELOPMENT PROCESS TO DEVELOP NON-TECHNICAL COMPETENCIES OF FUTURE SOFTWARE ENGINEERS

*The article describes the use of serious games and software development simulations to develop non-technical competencies of future software engineers. It was found that the typical software engineering education lacks a practical treatment of the processes of software engineering. This is due to the fact that the development of software, especially of large software systems, is considered to be a purely technical problem, but in practice it turns out differently: in most cases, the development of large (and not only) software systems, is mainly a non-technical problem. Revealed that the only feasible way to provide students with the experience of realistic software engineering processes within the academic environment is through simulation, as used in conjunction with lectures and projects. Proved that simulation can bring to software engineering education the same kinds of benefits that it has brought to other domains (medicine, aviation etc). In particular, the software process education can be improved and upgraded by allowing students to practice, through a simulator, the activity of managing various kinds of pseudo-realistic software development processes. Conducted a literature review of studies since 2000, which were published on the topic of software development using serious games and game simulations. The above-mentioned serious*

games and game simulations were analyzed in more details, with the correlation to non-technical competencies that are required for future software engineers.

**Keywords:** serious games; simulations; information and communications technologies (ICT); software engineers; non-technical competencies.

**Концедайло Валерій Валерьевич**

*Житомирский государственный университет имени Ивана Франко*

**ИСПОЛЬЗОВАНИЕ СЕРЬЕЗНЫХ ИГР И СИМУЛЯЦИЙ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ РАЗВИТИЯ НЕТЕХНИЧЕСКИХ КОМПЕТЕНТНОСТЕЙ БУДУЩИХ ИНЖЕНЕРОВ-ПРОГРАММИСТОВ**

*В статье рассмотрено использование серьезных игр и симуляций разработки программного обеспечения для развития нетехнических компетенций будущих инженеров-программистов. Выяснено, что типичному образованию инженеров-программистов не хватает практического освоения процессов разработки программного обеспечения. Это связано с тем, что разработка программного обеспечения, а, особенно, крупных систем программного обеспечения, принято считать сугубо техническим заданием, однако на практике оказывается иначе: в большинстве случаев разработка крупных (и не только) систем является задачей, в основном, "нетехнического" характера. Выявлено, что единственным возможным способом предоставления студентам опыта участия в реальных процессах разработки программного обеспечения (ПО) в академической среде является использование игровых симуляторов и симуляций в сочетании с лекциями и учебными проектами. Доказано, что симуляторы могут принести в образование инженеров-программистов ту же пользу, которую они принесли и в другие отрасли (медицина, авиация и другие). В частности, речь идет о том, что процесс обучения и подготовки инженеров-программистов может быть улучшен при условии предоставления возможности студентам практиковаться с помощью симуляторов в управлении различными видами псевдо-реалистичных процессов разработки программного обеспечения. Проведен литературный обзор работ начиная с 2000 года, опубликованных по теме обучения разработке ПО с использованием серьезных игр и игровых симуляций. Рассмотрены вышеуказанные серьезные игры и игровые симуляции более подробно, по распределению по формированию нетехнических компетентностей будущих инженеров-программистов.*

**Ключевые слова:** серьезные игры; симуляции; информационно-коммуникационные технологии (ИКТ); инженеры-программисты; нетехнические компетентности.

**ВІДОМОСТІ ПРО АВТОРІВ**

**Концедайло Валерій Валерійович** – аспірант Житомирського державного університету імені Івана Франка.

*Коло наукових інтересів:* електронне навчання; розвиток нетехнічних компетентностей майбутніх інженерів-програмістів