

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Одеська національна академія харчових технологій  
Навчально-науковий інститут комп'ютерних систем і технологій  
"Індустрія 4.0" ім. П.М. Платонова  
Факультет Комп'ютерної інженерії, програмування та кіберзахисту

**XIX Всеукраїнська науково-технічна конференція  
молодих вчених, аспірантів та студентів**

**“СТАН, ДОСЯГНЕННЯ І ПЕРСПЕКТИВИ  
ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ”**

*Матеріали конференції. Частина 2*



Одеса  
22 квітня 2019 р.

**Стан, досягнення і перспективи інформаційних систем і технологій** / Матеріали XIX Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. Одеса, 22 квітня 2019 р. - Одеса, Видавництво ОНАХТ, 2019 р. - 68 с.

Збірник включає матеріали доповідей її учасників, які об'єднані по секціях кафедр: комп'ютерної інженерії (КІ), інформаційних технологій та кібербезпеки (ІТтаКБ).

## **ОРГАНІЗАЦІЙНИЙ КОМІТЕТ**

### **Організаційний комітет**

Голова – д.т.н., проф., **Єгоров Б.В.**, ректор ОНАХТ.

#### **Співголови:**

**Поварова Н.М.** – к.т.н., доц., проректор з наукової роботи ОНАХТ,  
**Котлик С.В.** – к.т.н., доц., директор ННІКСіТ "Індустрія 4.0" ОНАХТ,  
**Даріуш Долива**, д.математичн.наук, уповноважений декана факультету Інформатики УІтаПЗ, м. Лодзь, Польща,  
**Ковалюк Т.В.** - к.т.н., доц. кафедри АСОІтаУ НТУУ «Київський політехнічний інститут».

#### **Члени оргкомітету:**

**Плотніков В. М.** – д.т.н., проф., завідувач кафедри ІТтаКБ ОНАХТ,  
**Артеменко С.В.** – д.т.н., проф., завідувач кафедри КІ ОНАХТ,  
**Князєва Н.О.** – д.т.н., проф. кафедри КІ ОНАХТ,  
**Хобін В.А.** – д.т.н., проф., завідувач кафедри АТПтаРС ОНАХТ,  
**Тарасенко В.П.** – д.т.н., проф., завідувач кафедри СКС НТУУ «Київський політехнічний інститут»,  
**Невлюдов І.Ш.** – д.т.н., проф., завідувач кафедри КІТАМ ХНУРЕ,  
**Мельник А.О.** – д.т.н., проф., завідувач кафедри ЕОМ НУ “Львівська політехніка”,  
**Жуков І. А.** – д.т.н., проф., завідувач кафедри КСтаМ НАУ.

Матеріали подано українською, російською та англійською мовами.  
Редактор збірника Котлик С.В.

## **ІГРОВА-ІНДУСТРІЯ ЯК СКЛАДОВА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ: МЕТОДОЛОГІЯ ТА ІНСТРУМЕНТИ**

**Крамський С. О., к.т.н., докторант, доцент кафедри менеджменту та економіки  
Міжрегіональна Академія управління персоналом**

Вступ. Світ інформаційних технологій дуже динамічний. Щодня з'являються все нові і нові технології і продукти. Звичайно ж, не обходиться і без розважальної сфери. Інформаційні технології – узагальнена назва технологій, що відповідають за зберігання, передачу, обробку, захист та відтворення інформації з використанням комп'ютерів. Ігрові ІТ-проекти представляють не тільки комунікативно-розважальну функцію, а так само виробляють заробіток їх творцям. Кожен проект є окремим цілим зі своєю ідеологією і функціональною відмінністю. Відповідно, до управління такими ІТ-проектами слід підходити не ординарно. Ігри є лідером розважальної інфраструктури. Щомісяця в світі стає все більше і більше нових ігрових продуктів зі своєю унікальною ідеєю та реалізацією.

Більшість ігрових компаній намагаються розробляти нові режими для залучення більшого попиту до їх продукції. Ігри умовно можна поділити на 2 типу: одиночні і розраховані на багато користувачів. Одиночна гра представляє собою певний сюжет з прописаними системними гравцями (ботами). Тому, ігри розраховані на багато користувачів, містять певну кількість реальних гравців на одній з генерованій карті. Всі гравці системно підключаються до сервера, на якому і зберігається потрібна інформація. При організації роботи у більшості молодих компаній є певні помилки та недоліки, через які на перших стадіях розробки найчастіше ІТ-команда розпадається, а продукт так і не виходить до кінцевого релізу. Але, щоб налагодити усі процеси виробництва, продукту потрібно дотримуватися певних моделей і методологій.

Перш за все, ми розглянемо методологію, яка використовується найчастіше у розробці програмного продукту в ІТ-індустрії – це SCRUM. Scrum – це методологія управління ІТ-проектами, яка побудована на принципах тайм-менеджмента і ролевого підходу. Далі, перейдемо до ролей, які залучені до самого процесу. Є власник продукту, Scrum майстер або керівник проекту, та Scrum команда ІТ-проекту. Основою Scrum є Sprint, впродовж якого виконується робота над продуктом [1]. Sprint – відрізок часу, який береться для виконання певного (обмеженого) списку завдань. Рекомендується брати 2-4 тижні (тривалість визначається командою один раз). По закінченню Sprint має бути отримана нова робоча версія продукту проекту. Перед початком кожного Sprint здійснюється “Sprint Planning”, на якому проводиться оцінка вмісту “Product Backlog” і формування “Sprint Backlog”, який містить завдання що повинні бути виконані в поточному спринті Кожен спринт повинен мати мету, яка є мотивуючим фактором і досягається за допомогою виконання завдань з Sprint Backlog. Спочатку команда ІТ-проекту на плануванні складає задачі з Product Backlog до Sprint BackLog, далі, спринт запускається, і задачі, які були



заплановані у Sprint BackLog переходять до статусу виконання. Наочно представлено час виконання цього спринту, а саме 15-30 днів. Команда та Scrum Master не може розривати тимчасові рамки, тобто додавати ще день або два. Якщо Sprint не виконали у срок, то він закривається і переходить до статусу “Провалений”, після цього вже не можна створювати новий спринт та аналізувати, чому саме команда не встигла у встановлений термін. показано час, який виділяється на щоденні обговорення з командою, а саме – 24 години з 30 днів. До щоденних “мітингів” входять обговорення кожного члена команди, на якій стадії він знаходиться, а також якісь нагальні проблеми. І якщо спринт пройшов добре, на виході ми отримуємо готову частину продукту ІТ-проекту. Далі, перейдемо до методології Канбан. Канбан часто порівнюють зі Scrum і зараховують до Agile-методологій. Методику можна назвати гнучкою, якщо говорити про розробку ПО, але сама по собі Канбан лише частково дотримується принципів гнучких методологій. Якщо порівнювати зі «Scrum»: в Kanban відсутні спринти, ролі, призначені для користувача історії, необов'язкові. При цьому методологію часто вважають більш «гнучкою», так як робочий процес практично не керується, мало регламентується, і результат на 90 % залежить від команди і повідомлення всередині неї, а не від менеджера.

Виходячи з цього, робочий колектив може бути як істотним недоліком, так і великою перевагою Kanban. Весь робочий процес візуалізується, щоб команда завжди розуміла, які завдання можуть почекати, а над якими необхідно працювати прямо зараз. Найбільш прийнятний інструмент для цього: Канбан-дошка – це таблиця з кількома стовпцями. Усередині стовпців знаходяться стікери з завданнями. У процесі роботи над проектом кожен член команди може переносити стікери у відповідне місце. За кожним членом команди може бути закріплені певні стікери з завданнями. Виходячи з чисельності команди проекту, визначається, над скількома завданнями вони можуть працювати одночасно [2].

Висновки. У роботі наведено методології розробки ігрового програмного продукту. Перша стадія це планування самого продукту, на який і збирається сама команда, назначається керівник і починається сама робота. Потім, після повноцінної роботи над ІТ-продуктом все це переходить до стадії кінцевого результату технічної частини. Останнім у цьому циклі є тестування та маркетинг. Хоч і на етапі розробки продукт тестується, по завершенню все одно треба провести остаточне тестування та підготовку продукту до релізу.

### **Список використаних джерел**

1. A Guide to the Scrum Body of Knowledge (SBOK Guide) 3rd ed. SCRUMstudy™, a brand of VMEdU, ISR, Arizona, USA. - 136.
2. Бойченко К.В. Управління ІТ-проектами в ігровій індустрії / Мат. XIV міжнар. наук.-практ. конф. "Актуальні проблеми сучасного управління в соціально-економічних, гуманітарних та технічних системах": Присвяченої 30–

чю МАУП. Збірник мат. тез доповідей. - О.: ОІ МАУП, ТОВ «Лерадрук», 2018. – С.216-222.

## **МОДЕЛЮВАННЯ ТА ДОСЛІДЖЕННЯ ЕКВІВАЛЕНТНІСТНИХ МОДЕЛЕЙ ГЕТЕРОАСОЦІАТИВНОЇ ПАМ'ЯТІ ДЛЯ ВИВЧЕННЯ ДОПУСТИМИХ РІВНІВ УШКОДЖЕНЬ НАДІЙНО АСОЦІЙОВАНИХ ЗОБРАЖЕНЬ**

**В.Г. Красиленко, к.т.н., с.н.с., доцент; Д.В. Нікітович, магістр;  
Вінницький національний технічний університет, м. Вінниця**

**Вступ, аналіз публікацій та постановка проблеми.** Останнє десятиріччя ознаменувалось значними досягненнями в області штучного інтелекту та нейрокомп'ютингу, суттєвим розширенням сфер застосування різноманітних нових моделей нейронних мереж (НМ), особливо загорткових НМ та алгоритмів їх машинного навчання, моделей авто-асоціативної, гетеро-асоціативної пам'яті (ГАП). На їх основі реалізуються аналізатори даних, асоціатори, системи машинного бачення, класифікатори, робото-технічні та діагностичні прогнозуючі комплекси різного призначення зі своїми недоліками, перевагами та специфічними особливостями. А тому це визначає актуальність і необхідність подальшого дослідження і вдосконалення відомих моделей НМ та нових концептуальних підходів. Переваги еквівалентністних, включаючи просторово-інваріантні узагальнені, моделей НМ, ГАП і багато-портових ГАП показані і підтверджені результатами моделювання в роботах [1-5]. В [3, 4] на прикладі розпізнавання множини взаємно пов'язаних пар зображень з розмірностями  $128 \times 128$ ,  $64 \times 64$  пікселів були висвітлені результати моделювання еквівалентнісної моделі (ЕМ) ГАП, де було показано, що такі ЕМ мають значну ємність, що на порядок перевищує кількість нейронів, та розпізнають вектори з ушкодженими завадами компонентами [4]. У вдосконалених ЕМ адаптивно-кореляційне зважування кореляційними коефіцієнтами і усереднені до навчальної вибірки коефіцієнтними векторами вхідних образів та авто-еквівалентнісне нелінійне загострення відгуків нейронів прихованих прошарків сприяло більш надійному розпізнаванню при значно менших параметрах (коефіцієнтах) нелінійних перетворень. Але більш детальні дослідження допустимих рівнів ушкоджень зображень (вхідних образів) для забезпечення надійних їх асоціацій в ЕМ ГАП без їх ємності не виконувались. У зв'язку з цим **метою** даної роботи є дослідження шляхом моделювання в MathCad роботи таких ЕМ ГАП та визначення допустимих рівнів завад, що спотворюють вхідні зображення – образи для асоціативного розпізнавання.

**Результати моделювання.** У першому нашому експерименті для моделювання багато-портової ГАП (БГАП) в середовищі MathCad нами використовувались як вхідні образи зображення букв у вигляді матриці