

Cross-cultural dialogue: Lectures on the problems of interethnic and interconfessional interaction. M. Yu. Martynovoy, V.A. Tishkova, N.M. Lebedevoy (Ed.). Moskva: RUDN Publ, 406 p. [in Russian].

5. Vershinina V.N. (Ed.). (2004). *Polikulturnoe vospitanie detey srednego i starshego doshkolnogo vozrasta* [Multicultural education of the children of middle and senior preschool age]. Ulyanovsk: UIPKPRO Publ. [in Russian].

6. Solodka A. (2005). *Polikulturne vykhovannia ta osvita – krok do intehtratsii osvity Ukrainy v yedynyi zahalnokulturnyi prostir* [Multicultural education and general education as a step to integration of Ukrainian education into the common cultural space]. *Naukovyi visnyk Mykolaivskoho derzhavnoho universytetu. Zb. prats.* Mykolaiv: MDU, pp. 212 – 217. [in Ukrainian].

Стаття надійшла до редакції 16.03.2017

УДК 378.091.12:005.963.2

Володимир Ракович, аспірант кафедри інформатики і кібернетики
Мелітопольського державного педагогічного університету імені Б. Хмельницького

АНАЛІЗ МЕТОДОЛОГІЙ РОЗРОБКИ ІГРОВИХ ДОДАТКІВ ДЛЯ СОЦІАЛЬНИХ МЕРЕЖ

У статті висвітлені загальні методології розробки ігрових додатків для соціальних мереж. Обґрунтовані переваги та недоліки зазначених методологій. Висвітлені найбільш популярні методології розробки програмного забезпечення. Наведені основні визначення гнучких методологій. Вибрана гнучка методологія, як одна із найбільш ефективних методологій під час вивчення дисципліни “Розробка мобільних додатків”.

Ключові слова: програмне забезпечення, гнучкі методології, ігрові додатки, життєвий цикл.

Рис. 1. Літ. 6.

Владимир Ракович, аспирант кафедры информатики и кибернетики
Мелитопольского государственного педагогического университета имени Б. Хмельницкого

АНАЛИЗ МЕТОДОЛОГИЙ РАЗРАБОТКИ ИГРОВЫХ ПРИЛОЖЕНИЙ ДЛЯ СОЦИАЛЬНЫХ СЕТЕЙ

В статье освещены общие методологии разработки игровых приложений для социальных сетей. Обоснованные преимущества и недостатки указанных методологий. Освещены наиболее популярные методологии разработки программного обеспечения. Приведены основные определения гибких методологий. Выбрана гибкая методология, как одна из самых эффективных методологий при изучении дисциплины “Разработка мобильных приложений”.

Ключевые слова: программное обеспечение, гибкие методологии, игровые приложения, жизненный цикл.

Volodymyr Rakovych, Postgraduate Student of the Computer Science and Cybernetics Department
Melitopol Bohdan Khmelnytskyi State Pedagogical University

THE ANALYSIS OF DEVELOPMENT OF GAMING APPLICATIONS FOR SOCIAL NETWORKS

The article highlights the general methodology of development of gaming applications for social networks. The advantages and disadvantages of these methodologies are discussed. The author highlights the most popular methodology of software development. The article provides the basic definition of the agile methodologies. The flexible methodology is selected as one of the most effective methodologies in the study of subject “the Development of mobile applications”.

Keywords: software, the flexible methodology, the gaming applications, a life cycle.

Постановка проблеми у загальному вигляді. На сьогодні соціальні мережі міцно увійшли в наше повсякденне життя. Кількість користувачів по усьому світу вимірюється мільйонами. Соціальні мережі були побудовані як сервіси для спілкування людей, але на даний момент вони також надають користувачу різноманітні додатки як для розваг, так і для

бізнесу. Це обумовлено присутністю в мережах цільової аудиторії і можливістю тісно спілкуватися з живими людьми.

Додатки для соціальних мереж може розмістити будь-яка людина. Така можливість соціальних мереж добре вирішує проблему доставки програмного забезпечення до користувача.

Великою перевагою для розробників програмного забезпечення є те, що створений додаток буде використовуватись великою кількістю людей. Це також є джерелом проблем у роботі веб-додатку, так як велика кількість користувачів створює високе навантаження на сервер. Також проблемою стає забезпечення стабільної роботи додатку та можливість його масштабування.

Мета статті. Проаналізувати загальні методології розробки ігрових додатків для соціальних мереж при вивченні дисципліни “Теорія ігор”.

Виклад основного матеріалу дослідження. Навчальна дисципліна “Теорія ігор” вивчається на магістратурі під час підготовки студентів спеціальності 122 Комп’ютерні науки та інформаційні технології протягом одного семестру. Метою вивчення дисципліни “Теорія ігор” є засвоєння необхідних знань щодо технологій створення додатків, які базуються на сучасних мобільних платформах і техніки їх застосування у реалізації бізнес-комунікацій. На вивчення дисципліни відводиться 120 години: лекційних 22, лабораторних 20 та 78 год. відводиться на самостійне опрацювання, при цьому тижневе навантаження складає 1 година.

На виконання будь-якого навчального завдання при вивченні дисципліни “Теорія ігор”: вивчення лекційного матеріалу, виконання лабораторної роботи, вирішення проекту витрачається дуже велика кількість ресурсів. В зв’язку з цим дуже важливо правильно планувати та використовувати ефективну методологію вивчення дисципліни.

Методологія розробки інформаційних систем – сукупність процедур, методів, інструментів і документації, які допоможуть розробникам в їх зусиллях по впровадженню нової інформаційної системи [1].

Основним нормативним документом, що регламентує склад процесів життєвого циклу ПЗ, є міжнародний стандарт ISO/IEC 12207:2008 “Systems and software engineering – Software life cycle processes”. Він визначає структуру життєвого циклу, що містить процеси, дії і завдання, які повинні бути виконані під час створення програмного забезпечення. Ця методологія зробила великий вплив на розробку інформаційних систем як загального підходу.

Існують різні моделі життєвого циклу розробки програмного забезпечення. Їх різноманітність зумовлено великою кількістю видів продуктів – від простого сайту до складного медичного ПЗ, і навіть якщо ви берете за основу одну з нижче наведених моделей – її в будь-якому випадку необхідно підганяти під особливості продукту,

проекту і компанії. Нижче наведені найбільш популярні методології розробки програмного забезпечення.

Waterfall (Водоспад) – каскадна модель процесу розробки ПЗ, в якій процес розробки виглядає як потік, що послідовно проходить фази аналізу вимог, проектування, реалізації, тестування, впровадження та підтримки. Нещодавно дана методологія була найбільш поширеною в розробці ігор. Вона рекомендує починати розробку проекту тільки після того, як буде готова у повному обсязі проектна документація [2].

Методологія передбачає, що документація є фінальною і не буде змінюватися. Тому процес розробки розподіляється на певну кількість модулів, кожен з яких має запланований час початку та тривалість. При цьому деякі модулі залежать від виконання попередніх.

Перевагою методології водоспаду є те, що вона дозволяє розподіляти обов’язки між відділами і добре піддається контролю. Розклад може бути встановлений з зазначенням термінів для кожної стадії розвитку і продукт може протікати через етапи процесу розробки один за іншим. Розробка рухається від концепції, через проектування, впровадження, тестування, установку, усунення несправностей, і закінчується в експлуатації і технічному обслуговуванні. Кожна фаза розвитку протікає в строго визначеному порядку.

Недоліком є те, що вона не допускає перегляду проекту і внесення змін. Після того, як додаток знаходиться в стадії тестування, дуже важко повернутися назад і змінити щось, що було не дуже добре спроектовано. Також великим недоліком є те, що працездатне програмне забезпечення можна отримати лише на пізніх етапах розробки. Дана модель може бути дуже ризиковою, так як невеликі помилки проектування можуть стати великими проблемами наприкінці розробки.

Iterative Model (Ітеративна) – не передбачає повного обсягу вимог для початку робіт над продуктом. Розробка програми може починатися з вимог до частини функціоналу, які можуть згодом доповнюватися і змінюватися. Процес повторюється, забезпечуючи створення нової версії продукту для кожного циклу [3].

В ітеративній моделі всі вимоги розподіляють на ітерації. Під час кожної ітерації, робота над проектом проходить через стадії визначення вимог, проектування, реалізації та тестування. Кожна наступна ітерація додає нові функції до попередньої версії. Процес триває до повної готовності системи відповідно до поставлених вимог.

Перевагою цієї моделі є те, що наприкінці кожної стадії розробки є працездатний додаток,

що надає можливість знайти функціональні або конструктивні недоліки. Виявлення проблем на ранній стадії розробки дозволяє вжити заходів щодо виправлення в умовах обмеженого бюджету.

Недоліком є те, що при використанні цієї методології можуть виникнути проблеми проектування, тому що на початку проекту не були зібрані всі вимоги. Дана методологія не підходить для маленьких проектів, так як їх важко розбити на ітерації.

Spiral (Спіральна) – поєднує ітеративну і послідовну (водоспаду) методології розробки з дуже високим акцентом на аналізі ризиків. Вона складається з чотирьох фаз. Проект багаторазово проходить через ці етапи, які називають “спіралі” [3].

Етап визначення цілей починається зі збору бізнес-вимог у першому витку спіралі. У наступних спіралях по мірі дозрівання продукту, дається визначення вимог системи та її підсистем. Ця фаза також включає в себе визначення вимог до системи шляхом безперервного обміну даними між клієнтом і системним аналітиком. В кінці спіралі продукт розгортається на певному ринку.

Етап проектування починається з концептуального проектування в базовій спіралі, в інших спіралях включає в себе архітектурне проектування, логічне проектування модулів і остаточний дизайн.

Етап розробки та перевірки продукту відноситься до виробництва фактичного програмного продукту на кожній спіралі. У вихідній спіралі формується концепт-документ для отримання зворотного зв'язку з клієнтами. Результатом наступних спіралей є робоча модель програмного забезпечення з номером версії. Ці збірки надсилають замовнику для зворотного зв'язку.

Етап аналізу ризиків включає в себе ідентифікацію, оцінку, а також моніторинг технічної здійсненності та управління ризиками. В кінці першої ітерації, після тестування збірки замовник оцінює програмне забезпечення і забезпечує зворотний зв'язок. На основі оцінки клієнта, процес розробки програмного забезпечення входить в наступну ітерацію. Процес ітерацій по спіралі триває протягом усього терміну служби програмного забезпечення.

Перевагою спіральної методології є те, що вона дозволяє додавати додаткові елементи до продукту. Вони будуть включені у проект тоді, коли вони стають доступними або відомими. Ще одним позитивним аспектом є те, що спіральна методологія передбачає раннє залучення користувачів до розробки системи.

Недоліком є те, що для успішного завершення продукту необхідне суворе управління проектом, так як існує ризик запуску спіралі в нескінченний цикл.

V-Model (V-модель) – є розширенням моделі водоспаду і заснована на асоціації стадії тестування для кожної відповідної стадії розробки [4, 8]. Це дуже суворя модель і наступний етап починається тільки після завершення попередньої фази.

Перевагою цієї методології є те, що планування, тестування та верифікація системи проводиться на ранніх етапах.

Недоліком є те, що ця методологія не є досить гнучкою, а також те, що написання коду відбувається у середині процесу розробки.

Agile методологія являє собою поєднання ітеративної та інкрементної моделей з акцентом на адаптивність процесу і задоволеність клієнтів шляхом швидкої доставки робочого програмного продукту [5, 50].

Методологія передбачає розбиття продукту на невеликі поступальні ітерації. Кожна ітерація зазвичай триває приблизно від однієї до трьох тижнів. Кожна ітерація включає в себе роботу команд різних напрямів, таких як планування, аналіз вимог, проектування, кодування, модульне тестування, і приймальні випробування. В кінці ітерації робочий продукт відображається для клієнта й інших зацікавлених сторін.

Кожна збірка додає нові функціональні можливості; остаточна збірка містить всі функції, необхідні замовнику. Методологія почала ставати популярною завдяки своїй гнучкості та адаптивності.

Методологія використовує адаптивний підхід, де немає ніякого детального планування. Є ясність тільки щодо того, які функціональні особливості повинні бути розроблені. Продукт тестується дуже часто, зводячи до мінімуму ризик виникнення будь-яких серйозних збоїв в майбутньому.

Взаємодія з клієнтами є основою методології Agile, і відкрите спілкування з мінімальною документацією є її типовими особливостями. Agile групи працюють в тісній взаємодії один з одним і найчастіше розташовані в тому ж географічному розташуванні.

Головною перевагою методології є те, що вона добре підходить як для проектів з фіксованими вимогами, так і для проектів з мінливими вимогами. Надає працездатний продукт на ранніх стадіях. Також важливою перевагою є те, що методологія може бути використана без планування або з малою її часткою.

Недоліком є те, що важко оцінити строки та кошторис усього проекту.

Методологія прототипування полягає у створенні прототипів програмних додатків, які відображають деякі функціональні можливості

продукту, та можуть повністю відрізнятись від фінальної версії продукту.

Прототипування є дуже популярним, так як воно дозволяє зрозуміти вимоги замовника на ранній стадії розвитку. Це допомагає отримати цінний зворотний зв'язок від клієнта і допомагає дизайнерам і розробникам зрозуміти, що саме очікується від продукту. На рис. 1 зображена схема процесу розробки програмного забезпечення у рамках даної методології.

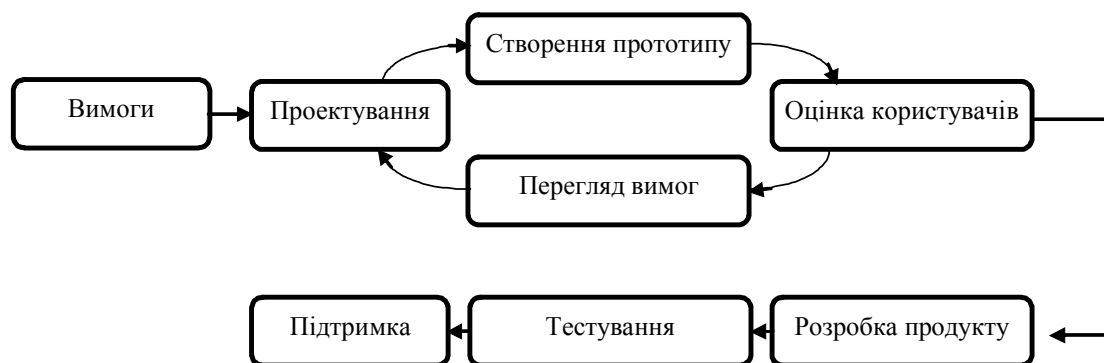


Рис. 1. Методологія прототипування

Розрізняють наступні типи прототипування: швидке, еволюційне, інкрементне, екстремальне [6, 83].

Швидке прототипування передбачає створення макету, який згодом не буде включений до кінцевого продукту. Вимагає мінімального аналізу вимог для побудови прототипу. Після того, як фактичні вимоги зрозумілі, прототип відкидається і фактична система розроблюється з набагато більш чітким розумінням потреб користувачів.

Еволюційний прототипування засновано на створенні реальних функціональних прототипів з мінімальною функціональністю на початку. Прототип стає серцем майбутніх прототипів, на базі якого буде побудована вся система. У прототип додаються лише добре вивчені вимоги, інші додаються тільки після їх зрозуміння. Перевагою такого підходу є те, що на кожному кроці ми маємо працюючу систему, хоч вона і не володіє всією необхідною функціональністю, але поліпшується з кожною ітерацією. При цьому, не витрачаються ресурси на код, який буде "викинуто".

Інкрементне прототипування передбачає створення декількох функціональних прототипів різних підсистем, а потім інтеграцію всіх наявних прототипів для формування повної системи.

Екстремальне прототипування найчастіше використовується в області веб-розробки. Вона складається з трьох послідовних етапів. По-перше, створення основного прототипу з усіма існуючими сторінками представлений в форматі

HTML. Потім додається прототип служби обробки даних, який симулює роботу реальних сервісів. Нарешті, сервіси реалізуються і інтегруються до остаточного прототипу.

Перевагами використання даного підходу є участь користувача у створенні продукту ще до початку його активної розробки, знижує витрати часу і коштів, оскільки дефекти можуть бути виявлені на ранніх стадіях, раннє виявлення відсутнього функціоналу. Також великою

перевагою є швидке розуміння здійсненності запланованого функціоналу.

Висновки з дослідження. Кожна з наведених методологій може бути використана при розробці ігрових додатків для соціальних мереж, але враховуючи швидкі зміни ринку програмного забезпечення більш доцільним є використання гнучкої методології розробки. Проте вона також не враховує специфіки розробки високонавантажених додатків для соціальних мереж.

Перспективи подальших розвідок. У подальшому науковому пошуку ми плануємо здійснити аналіз інших методик навчання розробки ігор з метою ґрунтовного дослідження цього питання.

Література

1. Avison D. E. *Information Systems Development: Methodologies, Techniques and Tools* / D.E. Avison, Guy Fitzgerald McGraw-Hill, 2003. – 592 с.
2. Цикл розробки програмного забезпечення [Електронний ресурс]. – Режим доступу: <http://batsihor.blogspot.com/2010/08/blog-post.html>
3. SDLC – Модель програмного забезпечення [Електронний ресурс]. – Режим доступу: http://www.uk.w3eacademy.com/sdlc/sdlc_software_prototyping.htm
4. Петрик М.Р. *Моделювання програмного забезпечення: науково-методичний посібник* / М.Р. Петрик, О.Ю. Петрик – Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2015. – 200 с.
5. Осадчий В.В. *Основи розробки веб-додатків: навч. посіб.* / В.В. Осадчий, В.С. Круглик. – Мелітополь: ТОВ "Вид. буд. ММД", 2012. – 540 с.
6. Якобсон. А. *Унифицированный процесс*

ОСОБЛИВОСТІ ФУНКЦІОНУВАННЯ КОРПОРАТИВНОЇ КУЛЬТУРИ ВИЩОГО НАВЧАЛЬНОГО ЗАКЛАДУ

разработки программного обеспечения / А. Якобсон, Г. Буч, Дж. Рамбо – СПб.: Питер, 2002. – 496 с.

www.uk.w3eacademy.com/sdlc/sdlc_software_prototyping.htm [in Ukrainian].

References

1. Avison D.E., Fitzgerald Guy (2003). *Information Systems Development: Methodologies, Techniques and Tools*. McGraw-Hill, 592 p. [in English].

2. Tsykl rozrobky prohramnoho zabezpechennia [The cycle of software development]. Retrieved from: <http://batsihor.blogspot.com/2010/08/blog-post.html> [in Ukrainian].

3. SDLC – Model' programnoho zabezpechennia [SDLC – Software Model]. Retrieved from: <http://>

4. Peter M.R., Peter O.Y. (2015). *Modelyuvannya programnogo zabezpechennia: naukovo-metodichniy posibnik [Simulation Software: Research Manual]*. Ternopil': Vid-vo TNTU imeni Ivana Pulyuya, 200 p. [in Ukrainian].

5. Osadchiy V.V., Kruglik V.S. (2012). *Osnovi rozrobki veb-dodatkov: navch. posib. [Basics of web application development: tutorial]*. Melitopol': TOV "Vid.bud.MMD", 540p. [in Ukrainian].

6. Jacobson A., Booch G., Rumbaugh James (2002). *Unifitsirovannyy protsess razrabotki programnogo obespecheniya [The unified software development process]*. SPb: Peter, 496 p. [in Russian].

Стаття надійшла до редакції 06.02.2017

УДК 378:005.35

Катерина Бровко, аспірант Київського університету імені Бориса Грінченка

ОСОБЛИВОСТІ ФУНКЦІОНУВАННЯ КОРПОРАТИВНОЇ КУЛЬТУРИ ВИЩОГО НАВЧАЛЬНОГО ЗАКЛАДУ

У статті розкрито особливості та закономірності функціонування корпоративної культури вищого навчального закладу. Визначено сутність понять “корпоративна культура”, “корпоративна культура вищого навчального закладу”. Виокремлено складові елементи корпоративної культури ВНЗ. Розкрито напрями формування корпоративної культури вищого навчального закладу. Подано інформацію щодо особливостей функціонування корпоративної культури Київського університету імені Бориса Грінченка.

Ключові слова: корпоративна культура, вищий навчальний заклад, університет, цінності, місія, Київський університет імені Бориса Грінченка.

Лит. 11.

Екатерина Бровко, аспирант Киевского университета имени Бориса Гринченко

ОСОБЕННОСТИ ФУНКЦИОНИРОВАНИЯ КОРПОРАТИВНОЙ КУЛЬТУРЫ ВЫСШЕГО УЧЕБНОГО ЗАВЕДЕНИЯ

В статье раскрыты особенности и закономерности функционирования корпоративной культуры высшего учебного заведения. Определена сущность понятий “корпоративная культура”, “корпоративная культура высшего учебного заведения”. Выделены составляющие элементы корпоративной культуры вуза. Раскрыты направления формирования корпоративной культуры высшего учебного заведения. Представлена информация об особенностях функционирования корпоративной культуры Киевского университета имени Бориса Гринченко.

Ключевые слова: корпоративная культура, высшее учебное заведение, университет, ценности, миссия, Киевский университет имени Бориса Гринченко.

Kateryna Brovko, Postgraduate Student, Kyiv Borys Hrinchenko University

FEATURES OF THE FUNCTIONING OF THE CORPORATE CULTURE OF HIGHER EDUCATIONAL INSTITUTIONS

The article deals with the features and principles of functioning of the corporate culture of the higher education institution. The essence of concepts “the corporate culture”, “the corporate culture of higher education institution” are determined. The constituent elements of the corporate culture of the university are obtained. The author discloses the directions of formation of the higher education institution's corporate culture. The information about the features of the functioning of the corporate culture of the Kyiv Borys Grinchenko University is presented.

Keywords: the corporate culture, a college, the university, values, a mission, Kyiv Borys Grinchenko University.

Постановка проблеми. Ефективність діяльності будь-якого вищого навчального закладу значною мірою залежить від того, наскільки його суб'єкти згуртовані щодо досягнення спільної мети та

цілей. Визначальним стратегічним інструментом, що дає змогу зорієнтувати всі структурні підрозділи вищого навчального закладу та його окремих осіб на спільну мету, активізувати ініціативність викладачів, студентів, а також