

АВТОМАТИЗАЦІЯ ПРОЦЕСІВ РОЗРОБКИ СИСТЕМ ЕЛЕКТРОННОЇ КОМЕРЦІЇ ЗА ДОПОМОГОЮ ПРЕДМЕТНО- ОРІЄНТОВАНОГО ПРОЕКТУВАННЯ

Вінницький національний технічний університет

Анотація

У цій роботі досліджується застосування предметно-орієнтованого проектування Domain-Driven Design (DDD) у розробці електронних комерційних систем, зосереджуючись на тому, як DDD спрощує складні системи, покращує співпрацю і узгоджує програмне забезпечення з бізнес-потребами. Розглядаються виклики імплементації, такі як крива навчання та інтеграція зі спадковими системами, та пропонуються рішення. Робота має на меті показати, як DDD може підвищити гнучкість, оперативність і відповідність бізнесу електронних комерційних платформ.

Ключові слова: предметно-орієнтоване проектування, розробка систем електронної комерції, автоматизація, методології agile.

Abstract

This paper examines Domain-Driven Design (DDD) in e-commerce system development, focusing on how DDD simplifies complex systems, improves collaboration, and aligns software with business needs. It discusses DDD's principles like bounded contexts and ubiquitous language, addresses implementation challenges such as the learning curve and legacy system integration, and suggests solutions. The work aims to show how DDD can enhance e-commerce platforms' agility, responsiveness, and business alignment.

Keywords: domain-driven design, e-commerce system development, automation, agile methodologies.

Вступ

Предметно-орієнтоване проектування (DDD, Domain-Driven Design) відіграє ключову роль у розробці систем електронної комерції, пропонуючи структуровану методологію, зосереджену на складних доменах у цій сфері. Цей підхід підкреслює важливість розуміння бізнес-домену для інформування проектування системи, забезпечуючи тісну відповідність програмних рішень бізнес-цілям і здатність вирішувати конкретні виклики, властиві електронній комерції[1][2][3].

DDD концентрується на предметній області, використовуючи обмежені контексти та спільну мову для моделювання програмного забезпечення. Цей підхід забезпечує узгодження розробки програмного забезпечення з бізнес-стратегіями, сприяючи спільному розумінню між розробниками та експертами області.

Оскільки електронна комерція розвивається, акцент DDD на специфічних для предметної області знаннях та колаборативних процесах проектування залишається незамінним для розробки систем, які не тільки задовольняють поточні ринкові вимоги, але й адаптовані до майбутніх змін [7].

Результати роботи

Впровадження предметно-орієнтованого проектування у розробці систем електронної комерції має глибокі наслідки для того, як ці системи концептуалізуються, проектуються та впроваджуються. Розглянемо трансформаційний вплив DDD на процеси розробки, та проаналізуємо його роль у спрощенні складних систем, сприянні співпраці та узгодженні розробки з бізнес-цілями.

Спрощення складних систем електронної комерції через стратегічне проектування. Системи електронної комерції є в своїй основі складними, інтегруючи різноманітні функціональності, такі як обробка платежів, управління запасами та обслуговування клієнтів. DDD вирішує цю складність шляхом декомпозиції систем на обмежені контексти, кожен з яких представляє окремий домен у більшій екосистемі електронної комерції. Це стратегічне проектування дозволяє командам зосереджуватися на конкретних областях системи, не будучи перевантаженими її загальною складністю. Дослідження показали, що застосування принципів DDD може призвести до більш керованих і модульних архітектур програмного забезпечення, покращуючи підтримку та масштабованість [3][4].

Покращення комунікації та співпраці в команді за допомогою спільної мови. Одним з основних принципів DDD є використання спільної мови, яку використовують як розробники, так і експерти предметної області. Ця спільна мова усуває непорозуміння і забезпечує, щоб усі інтереси мали чітке розуміння системи та її функціональності. У контексті електронної комерції, де узгодження між програмною системою та бізнес-цілями є критичним місцем, спільна мова сприяє більш зрозумілій комунікації та ефективнішому співробітництву[5][6].

Інкрементна та ітеративна розробка: Узгодження DDD з методологіями Agile. Фокусування DDD на основному домені та його ітеративний підхід до моделювання добре узгоджуються з методологіями Agile, які наголошують на інкрементній та ітеративній розробці та готовності до змін. Інтегруючи DDD з Agile практиками, команди розробників електронної комерції можуть швидше адаптуватися до змін на ринку, відгуків клієнтів та нових бізнес-можливостей. Ця синергія дозволяє швидко ітерувати програмні продукти, забезпечуючи еволюцію платформ електронної комерції у відповідь на динамічний онлайн-ринку. Поєднання DDD і методологій Agile сприяє культурі постійного вдосконалення та інновацій, що є важливим для підтримання конкурентної переваги в секторі електронної комерції [4][7].

Хоча предметно-орієнтоване проектування пропонує міцний каркас для вирішення складностей розробки систем електронної комерції, його впровадження не обходиться без викликів. Розглянемо загальні перешкоди, з якими зустрічаються під час застосування DDD у предметній області електронної комерції, і окреслемо практичні рішення та найкращі практики для подолання цих перешкод.

Навчання та експертиза. Одним з початкових викликів при реалізації DDD є крута крива навчання, пов'язана з його концепціями та методологіями. DDD вимагає глибокого розуміння бізнес-домену, а також здатності ефективно абстрагувати та моделювати його. Це може бути складно для команд, нових до DDD, або тих, у кого немає доступу до експертів предметної області. Для подолання цього виклику важливо інвестувати в навчання та наставництво для команди розробників. Заохочення співпраці між експертами предметної області та розробниками з самого початку також може полегшити передачу знань та допомогти командам швидше отримати необхідний досвід. Включення парного програмування та оглядів коду, зосереджених на принципах DDD, може додатково покращити навчання та дотримання найкращих практик DDD [3][4].

Інтеграція з успадкованими системами. Бізнеси електронної комерції часто покладаються на успадковані системи, які можуть не легко узгоджуватися зі стратегічними та тактичними принципами проектування DDD. Виклик полягає в інтеграції цих систем без компромісів щодо переваг, які DDD приносить. Застосування шаблону антикорупційного шару (Anticorruption Layer, ACL) є рекомендованим підходом для роботи зі спадковими системами. Шаблон ACL діє як проміжний рівень між спадковою системою та новою моделлю, запобігаючи забрудненню складностей спадкової системи системою, розробленою згідно з DDD. Поступовий рефакторинг спадкових систем, де це можливо, до більш предметно-орієнтованого проектування також може з часом знизити складності інтеграції [5].

Висновки

У цій роботі було розглянуто застосування предметно-орієнтованого проектування у розробці систем електронної комерції, підкреслюючи його значний вплив на спрощення складної бізнес-логіки, підвищення співпраці між мультидисциплінарними командами та тісне узгодження програмних рішень з

бізнес-цілями. Через дослідження різних викликів та рішень, вона продемонструвала практичність та ефективність DDD у вирішенні вроджених складностей платформ електронної комерції.

Предметно-орієнтоване проектування пропонує потужну парадигму для розробки систем електронної комерції, які не лише технічно досконалі, але й глибоко інтегровані з бізнес-стратегіями. Оскільки електронна комерція продовжує розвиватися, принципи та практики DDD безсумнівно відіграватимуть ключову роль у формуванні наступного покоління платформ електронної комерції, забезпечуючи їхню здатність задовольняти вимоги бізнесу та споживачів однаково.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Vernon V. Implementing Domain-Driven Design : Addison-Wesley Professional, 2013. 656 с. ISBN 978-0321834577.
2. Evans E. Domain-Driven Design: Tackling Complexity in the Heart of Software, 2003. 320 с. ISBN 9780321125217.
3. Domain-Driven Design in Software Development: A Systematic Literature Review on Implementation, Challenges, and Effectiveness. – URL: <https://arxiv.org/abs/2310.01905>
4. Patterns for Business-to-consumer E-Commerce Applications. – URL: <https://arxiv.org/abs/1108.3342>
5. Tackling Consistency-related Design Challenges of Distributed Data-Intensive Systems – An Action Research Study. – URL: <https://arxiv.org/abs/2108.03758>
6. Model-Driven Generation of Microservice Interfaces: From LEMMA Domain Models to Jolie APIs. – URL: <https://arxiv.org/abs/2202.11397>
7. A Reference Architecture for Blockchain-based Traceability Systems Using Domain-Driven Design and Microservices. – URL: <https://arxiv.org/abs/2302.06184>

Московко Сергій Геннадійович — факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: smoskovko@icloud.com.

Гуральник Фредерік Борисович — факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: frederikguralnik@gmail.com.

Moskovko Serhii G. — Department of intelligent information technologies and automation, Vinnytsia National Technical University, Vinnytsia, e-mail: smoskovko@icloud.com.

Guralnyk Frederik B. — Department of intelligent information technologies and automation, Vinnytsia National Technical University, Vinnytsia, e-mail: frederikguralnik@gmail.com.