



EUROPEAN CONFERENCE

Conference Proceedings

XXVII International Science Conference
«Current trends in the development of
modern educational technologies»

July 07-09, 2025
Munich, Germany

CURRENT TRENDS IN THE DEVELOPMENT OF MODERN EDUCATIONAL TECHNOLOGIES

Abstracts of XXVII International Scientific and Practical Conference

Munich, Germany
(July 07-09, 2025)

UDC 01.1

ISBN – 979-8-89940-585-3

The XXVII International scientific and practical conference «Current trends in the development of modern educational technologies», July 07-09, 2025, Munich, Germany, 153 p.

Text Copyright © 2025 by the European Conference (<https://eu-conf.com/>).

Illustrations © 2025 by the European Conference.

Cover design: European Conference (<https://eu-conf.com/>).

© Cover art: European Conference (<https://eu-conf.com/>).

© All rights reserved.

No part of this publication may be reproduced, distributed, or transmitted, in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher. The content and reliability of the articles are the responsibility of the authors. When using and borrowing materials reference to the publication is required. Collection of scientific articles published is the scientific and practical publication, which contains scientific articles of students, graduate students, Candidates and Doctors of Sciences, research workers and practitioners from Europe, Ukraine and from neighboring countries and beyond. The articles contain the study, reflecting the processes and changes in the structure of modern science. The collection of scientific articles is for students, postgraduate students, doctoral candidates, teachers, researchers, practitioners and people interested in the trends of modern science development.

The recommended citation for this publication is: Malyshevsky I.O., Kyfyak P.V., Myshkovsky Y.M. The study of the complex interplay of excessive body mass and abdominal neoplasms. Abstracts of XXVII International Scientific and Practical Conference. Munich, Germany. Pp. 92-94.

URL: <https://eu-conf.com/en/events/current-trends-in-the-development-of-modern-educational-technologies/>

ІННОВАЦІЙНІ ПІДХОДИ ДО ВИКЛАДАННЯ ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В КОНТЕКСТІ СУЧАСНИХ ОСВІТНІХ ТЕХНОЛОГІЙ

Калашник Валерій Юрійович

к.т.н., старший викладач

Київський Національний Університет Технологій та Дизайну

Анотація

У сучасному світі, де програмне забезпечення є рушійною силою інновацій, критично важливим стає якісна підготовка фахівців, здатних проектувати надійні, масштабовані та ефективні програмні системи. Дана робота присвячена аналізу та систематизації інноваційних підходів до викладання дисципліни "Проектування архітектури та забезпечення якості програмного забезпечення".

Акцент зроблено на інтеграції сучасних архітектурних патернів, практичних інструментів розробки (контейнеризація Docker, актуальні фреймворки) та комплексних стратегій тестування. Метою є формування у студентів не лише теоретичних знань, але й практичних навичок, необхідних для вирішення реальних інженерних завдань у сфері ІТ.

Вступ

Стрімкий розвиток інформаційних технологій та зростання складності програмних систем вимагають від освітніх програм постійної адаптації та впровадження новітніх підходів до навчання. Традиційні методи викладання архітектури та якості програмного забезпечення часто не встигають за динамікою індустрії, залишаючи прогалини між академічними знаннями та практичними потребами ринку праці. Для підготовки висококваліфікованих інженерів-програмістів, здатних працювати з розподіленими системами, мікросервісами та хмарними технологіями, необхідно переглянути зміст та методiku викладання відповідних дисциплін. Ця робота пропонує структуру та зміст навчального курсу, який інтегрує передові архітектурні практики та сучасні підходи до забезпечення якості, що відповідає вимогам XXVII Міжнародної науково-практичної конференції «Current trends in the development of modern educational technologies».

Основні положення та результати

Навчальний курс, розроблений на базі даного дослідження, охоплює ключові аспекти проектування архітектури та забезпечення якості ПЗ, з фокусом на практичну застосовність та відповідність сучасним індустріальним вимогам.

1. Проектування сучасних архітектурних рішень

Цей розділ курсу присвячений фундаментальним архітектурним патернам, які є основою для побудови будь-якої програмної системи. Розглядаються як класичні, так і новітні підходи:

- Багаторівнева архітектура (N-tier Architecture): Аналізується її структура (рівень представлення, бізнес-логіки, доступу до даних, бази даних), переваги у поділі відповідальності та модульності, а також недоліки, пов'язані з потенційною жорсткістю та зниженням продуктивності при великій кількості шарів. Практична реалізація демонструється на прикладі фреймворків, таких як ASP.NET Core (.NET) та Spring Boot (Java), які природно підтримують та заохочують такий поділ.

- Подієво-орієнтована архітектура (Event-Driven Architecture - EDA): Досліджується парадигма асинхронної взаємодії через події, що забезпечує слабку зв'язність, високу масштабованість та відмовостійкість. Обговорюються ключові елементи (видавці, споживачі, брокери повідомлень) та виклики (складність налагодження, забезпечення консистентності). Розглядаються такі брокери повідомлень, як Apache Kafka та RabbitMQ.

- Мікроядерна архітектура (Microkernel Architecture / Plugin Architecture): Вивчається концепція мінімального ядра та розширюваних плагінів, що забезпечує надзвичайну гнучкість та розширюваність системи, дозволяючи третім сторонам розробляти власні розширення. Аналізуються переваги та недоліки, зокрема складність розробки стабільного ядра.

- Мікросервіси: Детально розглядається підхід до побудови додатків як набору невеликих, автономних, незалежно розгортаємих сервісів. Акцентується увага на перевагах (масштабованість, відмовостійкість, гнучкість у виборі технологій, швидкість розробки) та недоліках (складність управління, розподілені транзакції).

- Клієнт-серверна архітектура: Розглядається як фундаментальний патерн, що лежить в основі більшості сучасних веб- та мобільних додатків, з акцентом на взаємодію "запит-відповідь" та розподіл ролей.

Особлива увага приділяється ролі Docker та контейнеризації як основи для розгортання та масштабування всіх цих архітектурних рішень, забезпечуючи ізоляцію середовища та спрощення CI/CD процесів. Також розглядаються сучасні фреймворки (NestJS для Node.js/TypeScript, ASP.NET Core для .NET/C#, Spring Boot для Java), які є ключовими інструментами для ефективної реалізації цих архітектур.

2. Комплексне тестування архітектурних рішень

Цей блок курсу зосереджений на забезпеченні якості програмного забезпечення через ретельне тестування на архітектурному рівні:

- Важливість тестування архітектури: Підкреслюється необхідність раннього виявлення архітектурних дефектів, підтвердження відповідності вимогам (як функціональним, так і нефункціональним), зниження ризиків та оптимізації продуктивності.

- Стратегії тестування інтеграції: Детально аналізуються підходи до перевірки взаємодії між компонентами та сервісами:

- Big Bang: Швидкий, але складний для локалізації помилок.

- Top-down (Низхідна): Раннє тестування основних функцій з

використанням заглушок.

- Bottom-up (Висхідна): Раннє виявлення дефектів у критичних низівних модулях з використанням драйверів.
- Sandwich (Гібридна): Комбінація обох підходів для паралельної інтеграції.
- Розглядаються інструменти для API-тестування (Postman, REST Assured), E2E-тестування (Cypress, Playwright) та використання Docker для створення тестових середовищ.
- Системне тестування: Фокусується на перевірці завершеної системи на відповідність нефункціональним вимогам (NFRs), що є критично важливим для архітектури:
 - Продуктивність (Performance): Час відгуку, пропускна здатність, використання ресурсів під навантаженням (JMeter, k6, Gatling).
 - Безпека (Security): Вразливості, захист даних, аутентифікація, авторизація (OWASP ZAP, Burp Suite).
 - Надійність (Reliability): Здатність працювати без збоїв, відновлення після збоїв.
 - Масштабованість (Scalability): Здатність справлятися зі збільшенням навантаження.
 - Відмовостійкість (Resilience): Здатність відновлюватися після збоїв компонентів.

3. Моделювання та візуалізація архітектури

Ефективна комунікація архітектурних рішень є запорукою успіху проекту. Цей розділ вивчає різні мови та підходи до моделювання:

- UML (Unified Modeling Language): Розглядаються ключові діаграми для архітектурного опису: діаграми компонентів (структура системи), розгортання (фізичне розміщення), пакетів (логічна організація), а також обмежено діаграми класів та послідовності для високорівневих абстракцій. Обговорюються обмеження UML для сучасних розподілених систем.
- C4 Model: Представляється як спрощений, ієрархічний підхід до візуалізації архітектури на рівнях Контекст, Контейнери, Компоненти, що забезпечує зрозумілість для широкої аудиторії. Надаються практичні поради щодо її використання.
- Archimate: Вивчається як мова моделювання підприємницької архітектури, що дозволяє відобразити зв'язок між бізнес-стратегією, організаційною структурою, інформаційними системами та технологічною інфраструктурою.
- Порівняльний аналіз: Проводиться глибоке порівняння UML, C4 та Archimate за критеріями типу інформації, сильних/слабких сторін, придатності для різних проектів та аудиторій, а також їхньої ролі на етапах життєвого циклу ПЗ. Особлива увага приділяється моделюванню складних архітектур (мікросервісів, хмарних рішень) та стратегіям підтримки узгодженості та еволюції моделей.

4. Оптимізація процесів тестування та забезпечення якості в експлуатації

Цей розділ курсу поглиблює знання про алгоритми тестування та управління якістю на етапі експлуатації:

- Алгоритми для оптимізації тестування:
 - Генерація тестових даних: Розглядаються алгоритми автоматичного створення тестових наборів на основі специфікацій (граничні значення, класи еквівалентності) та комбінаторних підходів.
 - Оптимізація тестових наборів: Вивчаються підходи до зменшення розміру тестових наборів, пріоритизації тестів та видалення надлишкових тестів.
 - Аналіз результатів тестування: Досліджуються алгоритми для виявлення дублікатів дефектів, кластеризації подібних проблем та прогнозування їхньої серйозності.
 - Оцінка ефективності алгоритмів: Визначаються критерії оцінки (покриття коду, час виконання, кількість виявлених дефектів) та методи порівняння алгоритмів.
 - Машинне навчання в тестуванні: Розглядається застосування ML для автоматизації генерації тестів, класифікації та прогнозування аномалій.
- Моніторинг та аналіз проблем:
 - Вивчаються цілі, завдання та типи метрик моніторингу (продуктивність, ресурси, доступність).
 - Представляються інструменти моніторингу (Prometheus, Grafana, ELK stack) та методи аналізу логів для проактивного виявлення проблем.
- Виявлення та усунення проблем. Реінжиніринг:
 - Розглядаються методи діагностики проблем (трасування, профілювання), класифікація дефектів та процес їх усунення.
 - Детально аналізуються цілі та стратегії реінжинірингу програмного забезпечення (рефакторинг, реструктуризація, зворотне проектування, міграція) як засіб покращення якості та адаптації існуючих систем.
 - Обговорюється роль DevOps та SRE у забезпеченні високої якості на етапі експлуатації, включаючи автоматизацію розгортання, моніторингу та оновлення ПЗ.

Висновки та практична цінність

Представлений підхід до викладання дисципліни "Проектування архітектури та забезпечення якості програмного забезпечення" дозволяє ефективно закривати ключові програмні результати навчання, формуючи у студентів глибокі концептуальні знання, практичні вміння та навички критичного мислення. Інтеграція актуальних індустріальних практик, таких як мікросервіси, контейнеризація, автоматизоване тестування, сучасні підходи до моделювання та управління якістю в експлуатації, забезпечує високу релевантність курсу та готує випускників до викликів сучасної ІТ-індустрії. Застосування цього курсу в освітніх програмах сприятиме підвищенню якості підготовки фахівців, здатних проектувати, розробляти та підтримувати складні програмні системи, що є

важливим трендом у розвитку сучасних освітніх технологій.

Список літератури

1. Richards, M., & Ford, N. (2020). *Fundamentals of Software Architecture*. O'Reilly Media.
2. Luksa, M. (2021). *Kubernetes in Action, Second Edition*. Manning Publications.
3. Newman, S. (2019). *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media.
4. Price, D., & Goetz, E. (2020). *ASP.NET Core in Action* (3rd ed.). Manning Publications.
5. Tilkov, S. (2019). *Event-Driven Architecture in Practice*. O'Reilly Media.
6. Byars, B. (2021). *Testing Microservices with Mountebank*. Manning Publications.
7. Jones, N. (2020). *Chaos Engineering: Building Confidence in System Behavior Through Experiments*. O'Reilly Media.
8. Bird, C., et al. (2021). *Machine Learning in Software Engineering*. Morgan & Claypool Publishers.
9. Сарновський, М. А. (2021). *Архітектура програмних систем*. К.: КПП ім. Ігоря Сікорського.
10. Кент, Б. (2020). *Чистий код. Створення, аналіз і рефакторинг*. К.: Наш Формат.
11. Григоренко, О. А. (2019). *Основи тестування програмного забезпечення*. К.: Центр учбової літератури.
12. Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice* (3rd ed.). Addison-Wesley.
13. Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley.
14. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
15. IEEE Std 1016-2009. (2009). *IEEE Standard for Information Technology – Systems and Software Engineering – Recommended Practice for Software Design Descriptions*. IEEE.

Scientific publications

MATERIALS

The XXVII International Scientific and Practical Conference
«Current trends in the development of modern educational technologies»

Munich, Germany
(July 07-09, 2025)