даних з оперативністю серверних сповіщень, що дозволяє нівелювати проблеми нестабільного з'єднання та ефективно вирішувати конфлікти версій.

Список використаних джерел

1. China C. R., Goodwin M. What Is Data synchronization? IBM. URL: https://www.ibm.com/think/topics/data-synchronization (дата звернення: 30.11.2025).
2. Kleppmann M. Designing data-intensive applications: the big ideas behind reliable, scalable, and maintainable systems. O'Reilly Media, Incorporated, 2017. 616 с.
3. Itani Z., Diab H., Artail H. Efficient pull based replication and synchronization for mobile databases. ICPS '05. international conference on pervasive services, 2005., м. Santorini, Greece. URL: https://doi.org/10.1109/perser.2005.1506554 (дата звернення: 30.11.2025).
4. Build an offline-first app. Android Developers. URL: https://developer.android.com/topic/architecture/data-layer/offline-first (дата звернення: 30.11.2025).
5. Chen M. What is cloud sync? Oracle. URL: https://blogs.oracle.com/cloud-infrastructure/what-is-cloud-sync (дата звернення: 30.11.2025).

МЕТОДИ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Лариков Данило
здобувач вищої освіти магістерського рівня
Кафедра інформаційно вимірювальних технологій
Захватова Тетяна
старший викладач
Кафедра фізичного виховання
Єгоров Андрій
к.т.н., професор
Кафедра інформаційно вимірювальних технологій
Харківський університет радіоелектроніки, Україна

Швидкий розвиток інформаційних технологій зумовлює підвищення вимог до якості програмного забезпечення. Користувачі очікують стабільної, безпечної та зручної роботи програмних продуктів, що можливо лише за умови ефективного контролю їхньої якості.

Одним із ключових етапів цього контролю є тестування — процес, який дає змогу не лише виявити помилки, а й оцінити надійність та придатність програмного забезпечення до практичного використання. Від ефективності тестування залежить не лише кінцева якість продукту, а й рівень довіри до нього з боку користувачів і замовників.

Відповідно до міжнародного стандарту IEEE 610.12-1990, тестування визначається як «процес оцінювання системи або її компонента з метою визначення, чи відповідають результати очікуванням».

Стандарт ISO/IEC 12207 визначає тестування як одну з центральних фаз життєвого циклу програмного забезпечення (ПЗ), що безпосередньо впливає на якість кінцевого продукту.

Тестування є важливим елементом системи забезпечення якості (Quality Assurance), яка охоплює всі дії, спрямовані на гарантування стабільності, функціональності та безпеки продукту.

Типовий процес тестування ПЗ включає такі основні фази:

1. Планування — визначення цілей, обсягів і критеріїв завершення тестування.
2. Проектування тестів — створення тест-планів, тест-кейсів і сценаріїв.
3. Підготовка середовища — налаштування інфраструктури, баз даних і тестових даних.
4. Виконання тестів — ручне або автоматизоване проведення тестових сценаріїв.
5. Аналіз результатів і звітність — документування виявлених дефектів, формування звітів.
6. Ретестування й регресія — перевірка виправлень і відсутності нових помилок.

Відповідно до стандартів IEEE 829 та ISO/IEC 29119, процес тестування організовується на кількох рівнях:

- Модульне (unit) — перевірка окремих функцій або класів;
- Інтеграційне — перевірка взаємодії між компонентами системи;
- Системне — тестування повного програмного продукту в робочому середовищі;
- Приймальне (acceptance) — перевірка готовності продукту до експлуатації замовником.

Тестування програмного забезпечення охоплює велику кількість напрямів, які різняться за метою, об’єктом перевірки, рівнем деталізації, ступенем автоматизації та способом взаємодії з програмою.

Кожен вид тестування має власне призначення, набір методів та інструментів, що дають змогу оцінити якість програмного продукту з різних точок зору: логічної правильності, стабільності, швидкодії, безпеки чи зручності користування. У загальному вигляді вид тестування — це сукупність дій, спрямованих на перевірку конкретних характеристик системи або її частини з урахуванням поставленої мети.

Узагальнена схема класифікації видів тестування програмного забезпечення подана на рисунку 1, який відображає основні напрями тестування за рівнем доступу до коду, способом виконання, ступенем автоматизації та метою перевірки.

Позначення на рисунку:

- Scalability Testing – визначення, чи зберігається продуктивність при збільшенні навантаження або обсягу даних.

- Stability (Endurance) Testing – тривале тестування під середнім навантаженням для виявлення витоків пам’яті та збоїв.
- Load Testing – перевірка реакції системи при поступовому зростанні навантаження.
- Stress Testing – оцінка поведінки програми при перевищенні критичних меж навантаження.

Рисунок 1 - Класифікація видів тестування програмного забезпечення

У загальному випадку класифікації видів і типів тестування будуються за наступними ознаками:

- Класифікація за виконанням коду,
- За способом проведення перевірки,
- Класифікація за доступом до коду,
- За рівнем знань тестувальника про внутрішню структуру програми,
- Класифікація за рівнем деталізації системи,
- За рівнем тестування,
- Класифікація за ступенем автоматизації,
- За способом виконання,
- Класифікація за принципами роботи з додатком,
- Класифікація за метою тестування.

Список використаних джерел

1. ISO/IEC 12207:2008 Systems and software engineering — Software life cycle processes.

2. ISO/IEC/IEEE 29119:2022 Software and systems engineering — Software testing.
3. IEEE 610.12-1990 Glossary of Software Engineering Terminology.
4. IEEE 829-2008 Software and System Test Documentation.

RESEARCH ON RISKS AND CYBER THREATS AT THE ENTERPRISE AND DEVELOPMENT OF A SYSTEM TO ENSURE ITS INFORMATION SECURITY

Shapovalova Olena
Ph.D., Associate Professor
Department of Cybersecurity and Information Technologies
Karnauchenko Andriy
Master's degree student
Simon Kuznets Kharkiv National University of Economics,
Kharkiv, Ukraine

Information crime is becoming increasingly relevant as a specific type of illegal activity in cyberspace [1]. In the context of rapid digitalization and geopolitical instability, cybersecurity is becoming a critical factor for business survival. IT companies with research and development (R&D) departments are particularly at risk. Unlike the financial sector, where funds are the main asset, the main asset of R&D companies is intellectual property: source code, architectural solutions, and customer data.

The relevance of the study is due to the growing number of targeted attacks on supply chains and ransomware, which can instantly paralyze development processes. For medium-sized enterprises, which often do not have the budgets of large corporations, it is critically important to create an effective but economically sound protection system [2].

The aim of this work is to develop and design a comprehensive information security system for the IT company NovaTech Initiatives Ukraine based on a detailed analysis of risks and modern technological solutions.

The design of the protection system is based on a hybrid methodological approach. Analysis of existing standards has shown that for comprehensive protection, it is advisable to combine the process-oriented approach of ISO/IEC 27001 (for building an Information Security Management System — ISMS) [3] with specific technical controls and NIST frameworks (in particular, NIST SP 800-53 and Cybersecurity Framework) [4].

A qualitative-quantitative matrix method based on the recommendations of ISO/IEC 27005 was chosen for risk assessment. This approach allows risks to be classified by level (from low to critical) by comparing the probability of a threat occurring and its potential impact on business processes, which is optimal for medium-sized enterprises [5].