УДК 004.4; 004.5
DOI: 10.37203/kibit.2023.49.03

Ігор ГУНЬКО
спеціаліст «Облік та аудит»,
бакалавр «Комп'ютерні науки»,
менеджер з тестування програмного забезпечення
ORCID ID: 0009-0001-7704-6352
twinigor@gmail.com
м. Київ, Україна

ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ У 2023 РОЦІ: НОВІ ТЕНДЕНЦІЇ ТА ПРОБЛЕМИ

У статті з'ясовано специфіку програмного забезпечення 2023 року, зокрема, розкрито особливості об'єднання адаптації та інновації, щоб задовольнити мінливе середовище розробки програмного забезпечення. У дослідженні висвітлено модерні тенденції у тестуванні програмного забезпечення (штучний інтелект і машинне навчання в автоматизації тестування, безперервне тестування та DevOps, інтеграція технології блокчейн у безпечне тестування програмного забезпечення); з'ясовано головні проблеми та ризики галузі (управління тестовими даними, питання тестового середовища й автоматизації тестування; ризики якості, надійності та безпеки, комплаєнс і регуляторні ризики). Доведено потребу нагального впровадження сучасних інновацій в українську царину тестування програмного забезпечення задля надання ефективних програмних рішень, котре охоплює всі аспекти життєдіяльності, гарантуючи надійність, безпеку і функціональність.

***Ключові слова**: тестування програмного забезпечення, тенденції, штучний інтелект, машинне навчання, безперервне тестування, блокчейн, етичне тестування.*

Ihor HUNKO
Specialist «Accounting and Audit»,
Bachelor «Computer Science»,
QA manager
ORCID ID: 0009-0001-7704-6352
twinigor@gmail.com
Kyiv, Ukraine

SOFTWARE TESTING IN 2023: NEW TRENDS AND CHALLENGES

The article clarifies the specifics of the software of 2023, in particular, reveals the peculiarities of combining adaptation and innovation to meet the changing environment of software development. Research in the field of software testing has delved into various aspects of a comprehensive problem, laying the foundation for further study and innovation in practice to increase the efficiency and accuracy of testing, create intelligent test scenarios, predictive analysis of defects, optimization of testing, which will reduce manual labor, accelerate test execution and detect defects at early stages of the development life cycle.

The work highlights modern trends in software testing (artificial intelligence and machine learning in test automation, implemented in the generation of test scenarios, predictive analytics and optimization of test execution; continuous testing and DevOps, actualized in automated, parallel, distributed pipeline testing and shift testing (left; integration of blockchain technology into secure software testing, including test data management, secure collaboration, and audit logs); clarifies the main problems and risks of the industry (quality and reliability risks include comprehensive and continuous testing and Shift-Left testing, security risks – rigorous security testing and ethical hacking, compliance and regulatory risks – documentation, auditing and compliance mechanisms, test environment – infrastructure automation and version control for environments, test automation – test automation strategy, regular maintenance and training and skill development).

Software testing in 2023 is a dynamic industry, constantly evolving at the forefront of technological innovation, playing a critical role in ensuring the quality, reliability and security of software. Constant adaptation, innovation, and commitment to ethical testing practices must be addressed, as software testing will continue to be a driving force in delivering high-quality, secure, and ethical software solutions that shape the digital landscape. The need for the urgent introduction of modern innovations into the Ukrainian field of software testing in order to provide effective software solutions that cover all aspects of life, guaranteeing reliability, safety and functionality, has been proved.

Keywords: *software testing, trends, artificial intelligence, machine learning, continuous testing, blockchain, ethical testing.*

Вступ

У 2023 р. тестування програмного забезпечення зіштовхнулось із багатогранною проблемою: складність програмних систем, що постійно зростає, у поєднанні з потребою швидкої розробки й розгортання вимагає переоцінки наявних методологій тестування. Тому потрібний баланс між ретельністю та ефективністю тестування. Традиційні підходи до тестування часто виявляються недостатніми, що збільшує ризик потрапляння дефектів програмного забезпечення у виробництво. Навпаки, надмірне тестування зумовлює затримки проєктів і збільшення витрат. Ця дилема окреслює гостру необхідність

інноваційних методів тестування, які оптимізують якість програмного забезпечення, одночасно дотримуючись обмеження його сучасної розробки. Звідси, актуальність роботи незаперечна, враховуючи зростаючу частоту збоїв програмного забезпечення у практичній реалізації сценаріїв із високими ставками.

Дослідження в галузі тестування програмного забезпечення поглибилися в різні аспекти всеосяжної проблеми, заклавши основу для подальших студіювань та інновацій у практичній діяльності, що презентовано в кількох аспектах.

Автоматизація та тестування з використанням штучного інтелекту (ШІ). Автоматизація значно підвищила ефективність, проте залишаються проблеми з автоматизацією складних сценаріїв тестування і забезпеченням надійності й адаптованості тестування на основі ШІ. Тож потрібні передові методології та середовища тестування програмного забезпечення.

Тестування зі зсувом вліво. Дії з тестування переносяться на більш ранні етапи життєвого циклу розробки програмного забезпечення, однак виникають проблеми з практичною реалізацією та інтеграцією з робочими процесами Agile і DevOps.

Безперервне тестування. З появою конвеєрів безперервної інтеграції та безперервної доставки (CI/CD) з'явилося поняття безперервного тестування (постійне тестуванні протягом усього процесу розробки програмного забезпечення). Необхідні оптимізовані стратегії безперервного тестування, що відповідають швидким циклам розробки.

Тестування стійкості та безпеки. Через кіберзагрози тестування стійкості та безпеки стає пріоритетним (кібератаки, превентивне тестування).

Метрики користувально-орієнтованого тестування. Дедалі частіше якість програмного забезпечення оцінюється на основі досвіду користувача. Однак послідовне визначення та вимірювання цих показників у різних галузях програмного забезпечення потребують ретельного вивчення.

Отже, у статті увагу зосереджено на новаторських ідеях, перспективі та глибшому розумінні середовища тестування програмного забезпечення, що розвивається у добу модерних цифрових технологій.

Мета дослідження – простудіювати інноваційні методології та парадигми тестування програмного забезпечення, проаналізувати складники модерних підходів, презентуючи новаторські факти, висновки та рекомендації, аби вплинути на практику (способи й перевірки) тестування програмного забезпечення у теперішньому і майбутньому галузі.

Методи та матеріали

Задля реалізації мети варто подолати розрив між традиційними підходами до тестування і складністю сучасних програмних систем, що зростає, щоб окреслити нові межі можливого в тестуванні програмного забезпечення, коли системи є безпечними та здатними протистояти викликам, що виникають за стрімкого розвитку цифрових технологій.

Насамперед доречно виокремити модерні тенденції у тестуванні програмного забезпечення.

1. ***Штучний інтелект і машинне навчання в автоматизації тестування.*** Штучний інтелект (ШІ) та машинне навчання (МН) охопили майже всі аспекти сучасних технологій. У 2023 р. потужний вплив зазначних феноменів на автоматизацію тестування здатний спричинити стрімкий прогрес: підвищити ефективність і точність тестування, створити інтелектуальні тестові сценарії, прогнозний аналіз дефектів, оптимізацію тестування, що зумовить скорочення ручної праці, прискорене виконання тестів і виявлення дефектів на ранніх етапах життєвого циклу розробки.

А. *Генерація тестових сценаріїв:* інструменти на основі ШІ аналізують код програми та користувацькі сценарії для інтелектуального створення тестових сценаріїв. Так, у фінансовому додатку інструмент ШІ може автоматично створювати тестові приклади різних типів транзакцій, забезпечуючи всебічне охоплення.

Б. *Предиктивна аналітика:* алгоритми МН аналізують історичні дані про дефекти, аби прогнозувати потенційні галузі високого ризику в програмному забезпеченні.

В. *Оптимізація виконання тестів:* інструменти виконання тестів на основі ШІ визначають пріоритетність тестових випадків, залежно від ймовірності виявлення дефектів, що гарантує першочергове виконання пріоритетних тестів.

2. Безперервне тестування та DevOps. Безперервне тестування – камінь сучасної розробки, пов'язаний з практиками DevOps. Мета – забезпечити інтеграцію тестування протягом усього життєвого циклу розробки програмного забезпечення, що дозволить швидше отримувати зворотний зв'язок і випускати версії, не знижуючи якість.

А. Автоматизоване тестування конвеєрів: у середовищі DevOps конвеєри безперервної інтеграції та безперервної доставки (CI/CD) автоматизовані. Безперервне тестування гарантує, що кожна фіксація коду запускає серію тестів, зокрема інтеграційні, модульні й регресійні, перед розгортанням коду.

Б. Тестування зі зсувом вліво: коли тестування починається на ранніх етапах процесу розробки, то розробники пишуть модульні тести разом із кодом, що дозволяє швидко виявляти й усувати дефекти у джерелі.

В. Паралельне та розподілене тестування: щоб задовольнити вимоги прискорених циклів випуску, під час безперервного тестування використовують стратегії паралельного і розподіленого тестування, що застосовують одночасно у різних середовищах, значно скорочуючи час виконання.

3. Інтеграція технології блокчейн у безпечне тестування програмного забезпечення. Технологію блокчейн, відому децентралізованою і стійкою до злому природою, застосовують у криптовалютах. При тестуванні програмного забезпечення блокчейн використовують для забезпечення цілісності та прозорості процесів тестування. Мета – підвищити цілісність і відстеження даних. Незмінність блокчейна гарантує, що дані, пов'язані з тестуванням, зокрема тестові приклади, журнали виконання і результати, не можуть бути

змінені або підроблені. Така прозорість підвищує достовірність результатів випробувань і процесів аудиту, гарантує захист тестових даних, надання доказів тестування, що перевіряються, та безпечну співпрацю в розподілених групах тестування.

А. Управління тестовими даними: Технологію блокчейна можна використовувати для безпечного зберігання і керування конфіденційними тестовими даними (наприклад, особиста інформація (PII)), доступ до яких контролюється за допомогою криптографічних ключів, тобто лише авторизованого персоналу.

Б. Безпечне співробітництво: блокчейн забезпечує безпечну співпрацю, підтримуючи прозорий реєстр тестових дій у групах, коли тестувальники з різних місць можуть отримати доступ до захищеної від несанкціонованого доступу історії виконання тестів, що забезпечує довіру і підзвітність.

В. Журнали аудиту: блокчейн створює незмінні журнали аудиту для тестування, що в регульованих галузях (наприклад, охорона здоров'я) містять докази, які перевіряються за стандартами тестування і захисту даних.

Окресливши основні модерні тенденції у тестуванні програмного забезпечення, доречно проаналізувати проблеми 2023 р. в галузі.

1. Проблеми управління тестовими даними лишаються сталими для тестування програмного забезпечення. Мета – гарантувати, що тестові дані є репрезентативними, різноманітними та постійно доступними для сценаріїв тестування. Проблеми виникають через складність сучасних програмних систем, необхідність забезпечення конфіденційності, відповідності даних і динамічний характер вимог до них. Однак наявні стратегії пом'якшення наслідків, як-от:

А. Маскування і анонімізація даних: конфіденційні дані можуть бути замасковані чи анонімізовані для захисту конфіденційності, зберігаючи при цьому структуру даних для тестування.

Б. Інструменти генерації даних можуть створювати різноманітні набори тестових даних, які охоплюють різні сценарії.

В. Механізми управління даними забезпечують якість, узгодженість і відповідність даним.

2. Проблеми тестового середовища. Мета – максимально точне копіювання виробничого середовища для забезпечення точного тестування, чому перешкоджають складність конфігурацій системи, залежності й обмежені ресурси. Тож варто вдатись до таких стратегій пом'якшення наслідків:

А. Віртуалізація середовища дозволяє створювати репліки тестового середовища, зменшуючи конкуренцію за ресурси.

Б. Контейнеризація: інструменти контейнеризації (наприклад, Docker) спрощують надання тестових середовищ та керування ними.

В. Інфраструктура як код (IaC) автоматизує налаштування середовища, забезпечуючи узгодженість і відтворюваність.

3. Питання автоматизації тестування постає наріжним каменем ефективного тестування, пов'язане з обслуговуванням сценаріїв та вибором

інструментів, забезпеченням всебічного покриття тестуванням. Тому слід озброїтись відповідними стратегіями пом'якшення наслідків.

А. Безперервне обслуговування тестування: регулярно оновлюйте тестові сценарії, враховуючи зміни у програмі.

Б. Надійні середовища тестування: вибирайте середовища автоматизації тестування, що підтримують модульність і можливість повторного використання.

В. Автоматизація тестування на основі штучного інтелекту: використовуйте ШІ та МН для створення і обслуговування інтелектуальних тестових сценаріїв.

Результати

Простудіювавши проблеми галузі, варто виокремити групи ризиків, пов'язані з неналежним тестуванням.

1. Ризики якості та надійності: неадекватне тестування може зумовити появу невиявлених дефектів та вразливостей, що негативно вплине на якість і надійність програмного забезпечення, оскільки провокуватиме збої, помилки та зниження задоволеності користувачів. Для того, аби помякшити наслідки, можна використати низку стратегій.

А. Комплексне тестування: вдайтесь до ретельного тестування (функціональне, нефункціональне і регресійне).

Б. Тестування за допомогою Shift-Left: почніть тестування на ранніх етапах процесу розробки, щоб виявити та усунути проблеми у джерелі.

В. Безперервне тестування: інтегруйте тестування в конвеєр CI/CD, аби гарантувати, що кожна зміна коду проходить тестування.

2. Ризики безпеки: неадекватне тестування безпеки може зробити програмне забезпечення вразливим для кібератак, витоку даних та несанкціонованого доступу, що зумовить компрометацію даних, фінансові й репутаційні втрати. Стратегіями пом'якшення наслідків є такі:

А. Ретельне тестування безпеки: тестування на проникнення, сканування вразливостей та огляди коду, щоб виявити й усунути слабкі місця безпеки.

Б. Етичний злом: Залучайте етичних хакерів для оцінки стану безпеки програмного забезпечення і виявлення вразливостей, аби уможливити втручання зловмисників.

3. Комплаєнс та регуляторні ризики: недотримання галузевих правил і стандартів відповідності спричинить юридичні наслідки, штрафи та зіпсовані ділові стосунки. Цьому можна запобігти, якщо вжити низку заходів.

А. Застосування актуальних для галузі *механізмів забезпечення відповідності* (HIPAA, GDPR або PCI DSS).

Б. Документація та аудит: ведення повної документації процесів та результатів тестування для демонстрації відповідності.

Крім окремих випадків, наявні загальні стратегії пом'якшення проблем та ризиків тестування.

4. Управління тестовими даними:

А. Політика конфіденційності даних: розробляйте і застосовуйте політику конфіденційності даних, яка регулює їх обробку.

Б. Віртуалізація даних: використовуйте інструменти віртуалізації даних для створення безпечних та ефективних рівнів доступу до них.

В. Автоматизація надання даних: автоматизуйте надання тестових даних, щоб скоротити ручні зусилля і забезпечити їх доступність.

5. Тестове середовище:

А. Автоматизація інфраструктури: інвестуйте в інструменти автоматизації інфраструктури, аби спростити налаштування і демонтаж середовища.

Б. Контроль версій для середовищ: введіть контроль версій для тестових середовищ, аби відстежувати зміни та забезпечувати узгодженість.

В. Моніторинг середовища: постійний моніторинг тестових середовищ допоможе розв'язувати проблеми продуктивності й ресурсів.

6. Автоматизація тестування:

А. Стратегія автоматизації тестування: побудуйте комплексну стратегію автоматизації тестування, що відповідає цілям і завданням проєкту.

Б. Регулярне обслуговування: регулярно оновлюйте сценарії, які мають відповідати змінам у програмі.

В. Навчання та розвиток навичок: інвестуйте у навчання і розвиток навичок інженерів з автоматизації тестування, щоб тримати їх у курсі нових технологій.

Висновки

У 2023 р. галузь тестування програмного забезпечення зазнала значних змін через багаторівневість складного технологічного середовища, оскільки потрібно опрацювати передові тенденції, подолати серйозні проблеми та взяти до уваги етичні міркування, дотримуючись чільної мети – забезпечити якість і надійність програмного забезпечення, беручи курс на постійну адаптацію та інновації, розуміння майбутніх напрямків тестування програмного забезпечення. Тому варто визначити пріоритетні напрямки у розвитку галузі.

1. Нові технології. Штучний інтелект (ШІ) та машинне навчання (МН) зроблять революцію в автоматизації тестування. Генерація тестових сценаріїв, що прогнозує аналіз дефектів, та оптимізоване виконання тестів стали невід'ємною частиною стратегій тестування.

2. Безперервне тестування, пов'язане з практиками DevOps, є нормою та невід'ємною частиною життєвого циклу розробки програмного забезпечення. Автоматизація тестування, тестування зі зсувом вліво й паралельне виконання забезпечують швидкий зворотний зв'язок і прискорення випуску версій.

3. Інтеграція технології блокчейн у безпечне тестування програмного забезпечення підвищила цілісність і прозорість даних (безпечна обробка, контрольні журнали та спільне тестування у групах).

4. Етичні міркування при тестуванні, особливо щодо ***конфіденційності*** та безпеки даних, спонукали тестувальників бути більш пильними щодо даних користувача, отриманні поінформованої згоди і дотриманні практики відповідального розкриття інформації.

5. Проблеми управління тестовими даними та середовища тестування можна розв'язати завдяки таким інноваціям, як віртуалізація даних, автоматизація інфраструктури й контейнеризація.

6. Автоматизація тестування стала свідком інновацій в обслуговуванні сценаріїв, виборі інструментів та комплексному покритті тестів. Надійні середовища тестування, автоматизація тестування на основі штучного інтелекту і регулярне обслуговування підвищують ефективність дій.

Тож тестування програмного забезпечення – це динамічна галузь, постійна актуальність якої залежить від адаптації та інновацій, що зумовлює важливість таких імперативів:

1. Гнучкість. Тестувальники й команди тестування повинні залишатися адаптованими й оперативно реагувати на мінливі технології та вимоги бізнесу, тобто вчитися і переучуватися, оскільки єдиною константою постають зміни.

2. Етичне тестування. Тестувальники повинні зберігати відданість етичним практикам тестування, гарантуючи, що конфіденційність, безпека й обробка даних відповідають найвищим етичним стандартам.

3. Освіта та розвиток навичок. Тестувальникам слід інвестувати у професійний розвиток, щоб опановувати нові тенденції, практики й інструменти.

4. Співпраця. Міжфункціональне співробітництво (наприклад, між командами тестування і розробниками) гарантує, що тестування відповідає бізнес-цілям і легко інтегрується у процес розробки.

5. Автоматизація та впровадження штучного інтелекту. Автоматизація тестування, заснована на ШІ й МН, буде розвиватися. Тестувальникам доцільно активно вивчати та впроваджувати інструменти і методи автоматизації для підвищення ефективності й підтримки конкурентоспроможності.

Майбутнє тестування програмного забезпечення відкриває захоплюючі можливості, а саме:

1. Тестування на основі штучного інтелекту: ШІ та МН формуватимуть майбутнє тестування. Автоматизація тестування стане більш інтелектуальною, а стандартними практиками будуть створення тестових сценаріїв на основі штучного інтелекту, прогнозна аналітика й тестові самовідновлювальні сценарії.

2. Тестування квантових обчислень: тестувальники розроблятимуть стратегії, що забезпечують квантову готовність і безпеку програмного забезпечення.

3. Тестування Інтернету речей: тестувальники зосередяться на тестуванні сумісності, безпеки та продуктивності екосистем IoT.

4. Тестування зі зрушенням праворуч забезпечуватиме моніторинг програмного забезпечення у виробництві для виявлення проблем, збору відгуків користувачів і постійного поліпшення якості.

5. Тестування на основі етики стане невід'ємною частиною методологій тестування. Програмне забезпечення має поважати конфіденційність користувачів, відповідати етичним принципам ШІ та суспільним цінностям.

6. ***Тестування, кероване даними,*** допомагатиме тестувальникам приймати обґрунтовані рішення щодо покриття тестами, оцінки ризиків і прогнозування дефектів.

7. Кількісні показники тестування: метрики тестування будуть розвиватися, забезпечивши розуміння якості програмного забезпечення, та відповідатимуть бізнес-результатам із пріоритетом цінності користувачів.

Отже, тестування програмного забезпечення 2023 р. – це динамічна галузь, яка постійно розвивається в авангарді технологічних інновацій, відіграє вирішальну роль у забезпеченні якості, надійності й безпеки програмного забезпечення. Необхідно звернути увагу на постійну адаптацію, інновації та етичну практику, оскільки тестування програмного забезпечення залишатиметься рушійною силою надання високоякісних і безпечних програмних рішень, що формують цифровий ландшафт.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bass, J. Clements, P. & Kazman, R. (2012). Software Architecture in Practice. Addison-Wesley.
2. Bohner, S. A., Marcin, T. & Carlson, M. (2017). How to Use Containers for IoT Device Security. https://www.linux.com/training-tutorials/how-use-containers-iot-device-security/
3. European Parliament and Council (2016). General Data Protection Regulation (GDPR). *Official Journal of the European Union.*
4. European Parliament and Council (2018). Regulation (EU) 2018/1725 on the protection of natural persons with regard to the processing of personal data by the Union institutions, bodies, offices, and agencies and on the free movement of such data. *Official Journal of the European Union.*
5. Fowler, M. (2003). Patterns of Enterprise Application Architecture. Addison-Wesley Professional.
6. Fowler, M. & Highsmith, J. (2001). The Agile Manifesto. https://agilemanifesto.org/
7. Garousi, V., Felderer, M., Mäntylä, M. V. & Kuhrmann, M. (2019). What practitioners expect of machine learning-based automated testing. Empirical Software Engineering. 24(6). 3633–3667.
8. Gelperin, D. & Hetzel, W.C. (1983). The Growth of Software Testing. *ACM Computing Surveys* (CSUR). 15(3). 211–220.
9. Gruver, M. Young, S. & Macias, T. (2017). Continuous Testing for DevOps Professionals. Pearson.

10. International Software Testing Qualifications Board (2018). ISTQB Glossary of Testing Terms. ISTQB.
11. International Software Testing Qualifications Board (2019). ISTQB Certified Tester Foundation Level Syllabus. ISTQB.
12. IoT World Today (2023). IoT Testing Strategies: The Full Guide. https://www.iotworldtoday.com/2022/07/28/iot-testing-strategies-the-full-guide/
13. ISO/IEC/IEEE. (2011). IEEE Standard Glossary of Software Engineering Terminology. IEEE.
14. IT Governance (2023). The GDPR (General Data Protection Regulation) guide. https://www.itgovernance.eu/gdpr-guide
15. Kaser, D. B. & Lemire, D. (2016). Compressed bitmap indexes: beyond unions and intersections. *Software: Practice and Experience*. 46(6). 723–764.
16. Kasyanov, I. & Zubkov, V. (2017). Application of Blockchain Technology in Cybersecurity. In 2017 10th International Conference on Security of Information and Networks (SIN); IEEE. 267–273.
17. Kruse, R. (2019). Ethical Hacking and Penetration Testing Guide. CRC Press.
18. Microsoft (2023). Introduction to Infrastructure as Code (IaC). Microsoft Docs. https://docs.microsoft.com/en-us/azure/developer/terraform/intro-to-iac
19. Microsoft (2023). What is Quantum Computing? Microsoft Quantum. https://learn.microsoft.com/en-us/azure/quantum/overview-what-is-quantum-computing
20. National Institute of Standards and Technology (NIST). (2017). NIST Special Publication 800-53: Security and Privacy Controls for Federal Information Systems and Organizations. U.S. Department of Commerce.
21. New England Complex Systems Institute. (2023). What is AI? https://necsi.edu/what-is-ai
22. OWASP (2022). OWASP Top Ten. Open Web Application Security Project. https://owasp.org/www-project-top-ten/
23. OWASP (2023). OWASP ZAP. Open Web Application Security Project. https://owasp.org/www-project-zap/
24. Pethuru Raj, C. & Anuragam, A. (2019). IoT Testing for Beginners: Learn the Basics of IoT Testing. Apress.
25. Python Software Foundation. (2023). Docker (Python client). PyPI. https://pypi.org/project/docker/
26. Shannon, C. E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*. 27(3). 379–423.
27. Shrestha, R. & Ali Babar, M. (2019). A Systematic Mapping Study on Blockchain Testing. In Proceedings of the ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems; ACM. 98–107.
28. The New Stack (2023). Shift-Left and Shift-Right: The Testing Conversation. https://thenewstack.io/shift-left-and-shift-right-the-testing-conversation/

REFERENCES

1. Bass, J. Clements, P. & Kazman, R. (2012). Software Architecture in Practice. Addison-Wesley.
2. Bohner, S. A., Marcin, T. & Carlson, M. (2017). How to Use Containers for IoT Device Security. https://www.linux.com/training-tutorials/how-use-containers-iot-device-security/
3. European Parliament and Council (2016). General Data Protection Regulation (GDPR). *Official Journal of the European Union.*
4. European Parliament and Council (2018). Regulation (EU) 2018/1725 on the protection of natural persons with regard to the processing of personal data by the Union institutions, bodies, offices, and agencies and on the free movement of such data. *Official Journal of the European Union.*
5. Fowler, M. (2003). Patterns of Enterprise Application Architecture. Addison-Wesley Professional.
6. Fowler, M. & Highsmith, J. (2001). The Agile Manifesto. https://agilemanifesto.org/
7. Garousi, V., Felderer, M., Mäntylä, M. V. & Kuhrmann, M. (2019). What practitioners expect of machine learning-based automated testing. Empirical Software Engineering. 24(6). 3633–3667.
8. Gelperin, D. & Hetzel, W.C. (1983). The Growth of Software Testing. *ACM Computing Surveys* (CSUR). 15(3). 211–220.
9. Gruver, M. Young, S. & Macias, T. (2017). Continuous Testing for DevOps Professionals. Pearson.
10. International Software Testing Qualifications Board (2018). ISTQB Glossary of Testing Terms. ISTQB.
11. International Software Testing Qualifications Board (2019). ISTQB Certified Tester Foundation Level Syllabus. ISTQB.
12. IoT World Today (2023). IoT Testing Strategies: The Full Guide. https://www.iotworldtoday.com/2022/07/28/iot-testing-strategies-the-full-guide/
13. ISO/IEC/IEEE. (2011). IEEE Standard Glossary of Software Engineering Terminology. IEEE.
14. IT Governance (2023). The GDPR (General Data Protection Regulation) guide. https://www.itgovernance.eu/gdpr-guide
15. Kaser, D. B. & Lemire, D. (2016). Compressed bitmap indexes: beyond unions and intersections. *Software: Practice and Experience*. 46(6). 723–764.
16. Kasyanov, I. & Zubkov, V. (2017). Application of Blockchain Technology in Cybersecurity. In 2017 10th International Conference on Security of Information and Networks (SIN); IEEE. 267–273.
17. Kruse, R. (2019). Ethical Hacking and Penetration Testing Guide. CRC Press.
18. Microsoft (2023). Introduction to Infrastructure as Code (IaC). Microsoft Docs. https://docs.microsoft.com/en-us/azure/developer/terraform/intro-to-iac

19. Microsoft (2023). What is Quantum Computing? Microsoft Quantum. https://learn.microsoft.com/en-us/azure/quantum/overview-what-is-quantum-computing
20. National Institute of Standards and Technology (NIST). (2017). NIST Special Publication 800-53: Security and Privacy Controls for Federal Information Systems and Organizations. U.S. Department of Commerce.
21. New England Complex Systems Institute. (2023). What is AI? https://necsi.edu/what-is-ai
22. OWASP (2022). OWASP Top Ten. Open Web Application Security Project. https://owasp.org/www-project-top-ten/
23. OWASP (2023). OWASP ZAP. Open Web Application Security Project. https://owasp.org/www-project-zap/
24. Pethuru Raj, C. & Anuragam, A. (2019). IoT Testing for Beginners: Learn the Basics of IoT Testing. Apress.
25. Python Software Foundation. (2023). Docker (Python client). PyPI. https://pypi.org/project/docker/
26. Shannon, C. E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*. 27(3). 379–423.
27. Shrestha, R. & Ali Babar, M. (2019). A Systematic Mapping Study on Blockchain Testing. In Proceedings of the ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems; ACM. 98–107.
28. The New Stack (2023). Shift-Left and Shift-Right: The Testing Conversation. https://thenewstack.io/shift-left-and-shift-right-the-testing-conversation/

Отримано редакцією / Received: 03.04.23
Прорецензовано / Revised: 18.04.23
Схвалено до друку / Accepted: 28.04.23