

УДК 004.05

ОГЛЯД СУЧАСНИХ ПІДХОДІВ ДО ОЦІНЮВАННЯ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

І. В. Гілета, Р. О. Кіселик, В. А. Карнаухов, Т. Н. Коминар

*Національний університет «Львівська політехніка»,
вул. Степана Бандери, 12, Львів, 79013, Україна*

Стаття присвячена сучасним підходам до оцінювання якості програмного забезпечення в умовах цифрової трансформації. Розглянуто еволюцію моделей якості, зокрема ISO/IEC 9126, 25010 та CMMI, та їх адаптації до сучасних контекстів. Особливу увагу приділено впливу штучного інтелекту на забезпечення якості: автоматизації тестування, прогностичній аналітиці, самовідновленню сценаріїв. Аналізуються парадигми Shift-Left/Shift-Right та практика DevOps як засади безперервної якості. Виявлено методологічні проблеми в емпіричних дослідженнях та виклики, пов'язані з етикою застосування штучного інтелекту. Запропоновано напрями подальших досліджень, зокрема щодо покращення бенчмарків, пояснюваності штучного інтелекту, інтерпретації результатів та адаптації моделей до міжгалузевих доменів. Стаття формує комплексне уявлення про трансформацію оцінювання якості програмного забезпечення та потребу в етично обґрунтованих, гнучких і прогностичних підходах.

Ключові слова: *якість програмного забезпечення, штучний інтелект, тестування ПЗ, емпіричні дослідження, моделі якості, автоматизація тестування, оцінювання ШІ, етичні аспекти ШІ.*

Постановка проблеми. Якість програмного забезпечення (ПЗ) стала невід'ємним фактором у сучасних взаємопов'язаних комп'ютерних системах. Наслідки низької якості ПЗ не обмежуються лише ІТ-відділами, а впливають на роботу підприємств та людей, призводячи до «величезних навмисних та ненавмисних наслідків» [1]. Це підкреслює критичну суспільну та економічну необхідність забезпечення надійної якості.

Зростаюча складність програмних систем, зумовлена швидкою цифровою трансформацією, широким впровадженням хмарних технологій та інтеграцією передових технологій, таких як штучний інтелект (ШІ) та машинне навчання (МН), є значним фактором, що стимулює попит на комплексні послуги для забезпечення якості програмного забезпечення (ЗЯПЗ) та тестування [2]. Ця динаміка свідчить про те, що якість програмного забезпечення перестала бути суто технічною проблемою, перетворившись на фундаментальний бізнесовий та суспільний імператив.

Аналіз останніх досліджень та публікацій. Ринок забезпечення якості програмного забезпечення переживає значне зростання, з прогнозами, що вказують на зростання до 7839,2 мільйона доларів США до 2033 року, демонструючи

сукупний річний темп зростання 14,47% протягом прогнозного періоду 2025-2033 років [2]. Ця експансія ринку відображає визнання галуззю якості як стратегічного диференціатора, необхідного для зниження ризиків, дотримання нормативних вимог та підтримки довіри клієнтів і безперервності бізнесу [3].

Мета статті. Задача дослідження синтезувати поточні знання та дослідницькі тенденції в сучасних підходах до оцінювання якості програмного забезпечення, зокрема зосереджуючись на періоді з 2020 по 2025 рік, критично оцінить якість доступних доказів та надати збалансовані висновки. Огляд охопить ключові сфери, включаючи методології первинних досліджень, еволюцію моделей якості, трансформаційну роль ІІІ та організаційних контекстів на методи оцінювання, поточні обмеження досліджень, сфери згоди та суперечок серед дослідників, практичні наслідки для розробки програмного забезпечення та напрямки для майбутніх досліджень.

Виклад основного матеріалу дослідження. Дослідження в галузі емпіричної інженерії програмного забезпечення (ЕПЗ) демонструють зростаючий попит з боку програмної індустрії на докази практичного застосування теорії інженерії програмного забезпечення [6]. Найбільшу увагу в дослідженнях ЕПЗ отримали такі теми, як організація, процес розробки програмного забезпечення та людський фактор. Останнім часом найбільша залученість тем для досліджень є продуктивність, кодування, тестування та програмна практика [6].

Поширені емпіричні методи дослідження в інженерії програмного забезпечення включають експерименти, опитування, тематичні дослідження та дослідницькі дії [7]. Ці методи використовуються для дослідження, опису, прогнозування та пояснення явищ шляхом спостереження та досвіду [7].

Незважаючи на зростаючий інтерес до ЕПЗ, галузь стикається зі значними викликами, які обмежують їх вплив та достовірність. Одним з головних викликів є постійно «низька якість емпіричних досліджень», що часто впливає з відсутності достовірності (валідності), зокрема проблем з конструктивною валідністю, коли вимірюються погано зрозумілі явища [7]. Крім того, «область валідності» – визначення сукупностей акторів, технологій, видів діяльності та програмних систем, для яких результати є дійсними – рідко явно визначається, що призводить до суперечливих висновків та труднощів у інтерпретації застосовності [7]. Багато досліджень проводяться дослідниками, які мають особисту зацікавленість у результаті, що потенційно вносить упередження, і сприймаються як прості демонстрації «доказу концепції», а не як суворі оцінки [7].

Зростання ЕПЗ свідчить про дисциплінарне прагнення до наукової строгості, проте воно критично гальмується постійними методологічними недоліками, які обмежують узагальнюваність та промислову застосовність результатів, створюючи «розрив у довірі» між академічними дослідженнями та практичними потребами. Початкова позитивна тенденція «зростаючого інтересу» та «найшвидше зростаючих тем» [6] вказує на потребу емпіричного обґрунтування. Однак, широка критика, що стосується «низької якості», «відсутності достовірності», «невизначеної області» та «особистої зацікавленості», виявляє глибоку внутрішню проблему. Це означає, що хоча намір створювати знання, засновані на доказах, присутній, виконання часто

не відповідає очікуванням, що призводить до «розриву між дослідниками та практиками інженерії програмного забезпечення».

Інтеграція штучного інтелекту (ШІ) та машинного навчання (МН) у тестування програмного забезпечення швидко перетворюється зі спекулятивної концепції на стратегічний імператив [5]. Галузеві звіти вказують, що понад 75% компаній планують розширити використання ШІ у 2025 році, причому тестування програмного забезпечення визначено як основний варіант використання [5].

Конкретні методи ШІ, включаючи згорткові нейронні мережі (ЗНМ), глибокі нейронні мережі (ГНМ) та великі мовні моделі (ВММ), використовуються для фундаментальної трансформації способів верифікації та валідації програмних додатків [10].

Тестування, кероване ШІ, пропонує значні переваги: воно суттєво зменшує час та зусилля, традиційно витрачені на повторювані та рутинні завдання тестування [10]. Ця автоматизація дозволяє тестувальникам переорієнтувати свою увагу на більш складні та творчі аспекти тестування, такі як дослідницьке тестування та оцінка користувацького досвіду [10]. Крім того, підходи, керовані ШІ, демонструють покращення якості програмного забезпечення шляхом підвищення тестового покриття, що призводить до відчутних переваг, таких як економія витрат та підвищення задоволеності клієнтів за рахунок значного зменшення ймовірності потрапляння критичних дефектів програмного забезпечення у виробництво [10].

ШІ застосовується в різних видах тестування, включаючи генерацію нових тестових випадків, забезпечення можливостей самовідновлення в тестових сценаріях, створення реалістичних тестових даних, пріоритизацію критичних тестів, виявлення аномалій та ідентифікацію першопричин збоїв системи або тесту [8].

На ринку з'являються спеціалізовані платформи та інструменти ШІ, такі як AIOpsLab для оцінки агентів ШІ в операціях хмарних систем [12], RIVER 2.0 [13], Diffblue Cover для автоматичної генерації модульних тестів [13], а також зростаюча екосистема комерційних послуг від таких постачальників, як Frugal Testing, QA Mentor, Cigniti, TestingXperts, Infosys та Qualitest, які пропонують рішення для тестування з підтримкою ШІ [11].

Швидке впровадження та інвестиції в ШІ для тестування програмного забезпечення свідчать про фундаментальний зсув від реактивного виявлення дефектів до проактивного, інтелектуального забезпечення якості. Ця трансформація переосмислює роль тестувальника, переміщуючи його від ручного виконання до стратегічного нагляду та вирішення складних проблем, одночасно вводячи нові виклики, пов'язані з оцінкою систем ШІ та етичними міркуваннями. Це формує перехід до більш інтелектуального та автономного ЗЯПЗ. Важливим моментом є те, що ШІ «дозволяє тестувальникам зосередитися на більш складних та творчих аспектах» [10], що означає переосмислення ролі людини, а не просту заміну. Однак [9] зазначає, що «впровадження ШІ відстає» попри високий інтерес та вводять «виклики», «обмеження», «упередженість» та «етичні проблеми». Ця подвійність показує, що хоча ШІ пропонує величезні перспективи, його повна реалізація залежить від вирішення цих складних технічних та етичних перешкод, що робить поточний період критичною фазою адаптації та вирішення проблем.

Оцінка якості програмного забезпечення історично ґрунтувалася на встановлених галузевих стандартах та моделях, які надають структуровані рамки для оцінки [15]. Дослідницький огляд виявив кілька ключових моделей якості програмного забезпечення, що ґрунтуються або розширюють широко прийняті галузеві стандарти:

- **ISO/IEC 9126:** Цей міжнародний стандарт зосереджений на шести основних характеристиках якості: функціональність, надійність, зручність використання, ефективність, ремонтпридатність та переносимість.
- **ISO/IEC 25010:** Перегляд та розширення ISO/IEC 9126, цей стандарт пропонує більш комплексну модель з вісьмома характеристиками якості, включаючи безпеку, сумісність та взаємодію.
- **CMMI (Capability Maturity Model Integration):** Ця модель використовується для оцінки та підвищення здатності організації послідовно розробляти програмне забезпечення, зосереджуючись на управлінні проектами, інженерії програмного забезпечення та управлінні процесами.

Інші помітні моделі включають «Інноваційну модель прогнозування дефектів програмного забезпечення з використанням імовірнісного ABC» Кумара та ін. (високий показник виявлення), «НММ-FNN» Юаньсю та ін. (покращена точність прогнозування та ефективність навчання) та «QAS (Система забезпечення якості)» Легово та ін. (перевірена точність у визначенні якості за допомогою гнучкої методології) [15].

Поширення спеціалізованих моделей якості програмного забезпечення свідчить про зростаюче визнання того, що універсальний підхід є недостатнім для різноманітного програмного забезпечення. Ця тенденція відображає перехід до більш тонкої, контекстно-орієнтованої оцінки якості, адаптуючись до таких факторів, як організаційна зрілість, людський талант або конкретні потреби безпеки. Багато моделей явно вказують на свою мету – вирішення конкретного контексту (наприклад, «фокус на людському таланті», «організації на ранній стадії», «інтегрує перспективу клієнта», «сприяє якості та безпеці»). Хоча вони часто використовують встановлені стандарти (ISO/IEC, CMMI) як основу, їхні модифікації або конкретні застосування підкреслюють необхідність адаптації. Це свідчить про те, що галузь рухається до більш детальних, проблемно-орієнтованих рішень, що означає, що ефективна оцінка якості вимагає глибокого розуміння контексту.

Оцінювання якості на основі ШІ. ШІ докорінно змінює забезпечення якості програмного забезпечення, значно підвищуючи точність, швидкість та загальну ефективність протягом усього життєвого циклу розробки програмного забезпечення [5]. Його застосування охоплює усі етапи розроблення.

Ключові застосування ШІ в ЗЯПЗ включають:

- Проектування, генерація та вибір тестових випадків: ШІ генерує нові тестові випадки, пріоритизує критичні тести та оптимізує виконання тестів [4].
- Виявлення дефектів та перевірка коду: ШІ допомагає виявляти аномалії, визначати першопричини збоїв та може автоматизувати аспекти перевірки коду, зменшуючи ручну працю та підвищуючи точність [8].

- Можливості самовідновлення: Однією з найцінніших функцій є здатність ШІ автоматично оновлювати та виправляти тестові сценарії, значно зменшуючи зусилля з підтримки тестів [5].
- Прогнозна аналітика: Прогнозна аналітика на основі ШІ дозволяє ідентифікувати зони високого ризику та прогнозувати дефекти до їх виникнення, що призводить до розумнішого виявлення дефектів та зменшення збоїв після випуску [5].
- Оптимізація продуктивності: ШІ може аналізувати та оптимізувати неефективні конструкції коду та використовується для моніторингу працюючих систем [10].
- Аудити безпеки: ШІ позначає небезпечні практики кодування та рекомендує безпечні альтернативи, посилюючи заходи кібербезпеки в процесі розробки [15].
- Інженерія вимог: ВММ демонструють можливості перетворення потреб зацікавлених сторін у формальні специфікації вимог, класифікації вимог та допомоги у моделюванні та специфікації вимог [15].
- Оцінка зручності використання: Нові інструменти, такі як UX-LLM, що працюють на базі великих візуально-мовних моделей, можуть прогнозувати проблеми зручності використання в мобільних додатках.

Інтеграція ШІ в оцінку якості програмного забезпечення виходить за рамки простої автоматизації; вона представляє зсув у бік «інтелектуального забезпечення якості», яке є прогностичним, адаптивним і здатним вирішувати складні, нефункціональні аспекти якості програмного забезпечення. Ця трансформація принципово змінює роль тестування від ручного виконання до стратегічного нагляду.

Тестування Shift-Left та Shift-Right все частіше визнається як необхідні стратегії для отримання швидшого та ефективнішого зворотного зв'язку протягом усього життєвого циклу розробки програмного забезпечення.

- Тестування Shift-Left: Цей підхід виступає за переміщення діяльності з забезпечення якості якомога раніше в процес розробки, в ідеалі – на етапи вимог та проектування [4].
- Тестування Shift-Right: Розширює тестування на виробниче середовище, зосереджуючись на зворотньому зв'язку від реальних користувачів та поведінці системи [4]. Наскрізне тестування, включаючи тестування інтерфейсу користувача, також може бути успішно валідовано після випуску [8].
- Комбінований вплив: Синергетичне застосування тестування Shift-Left та Shift-Right є вирішальним для досягнення безперервної якості [5]. Цей збалансований підхід забезпечує раннє вирішення проблем, одночасно гарантуючи стійкість виробництва та безперервне вдосконалення [8].

DevOps став найпопулярнішою методологією в ІТ-організаціях, з рівнем впровадження 49%. Ключові тенденції, що формують DevOps які безпосередньо впливають на безперервну якість, включають:

- Зростання автоматизації: Автоматизація перетворилася з бажаною функцією на важливий компонент, завдяки складним інструментам, що дозволяють майже повну автоматизацію конвеєрів безперервної інтеграції/безперервної доставки (CI/CD).

- Інтеграція ІІІ та машинного навчання: ІІІ та МН стають невід’ємною частиною процесів DevOps, пропонуючи можливості для автоматизації, ефективності та інновацій, зокрема, прогнозування збоїв системи та автоматизації повторюваних завдань.
- Безперервна інтеграція та безперервна доставка (CI/CD): Ці практики є фундаментальними для DevOps, скорочуючи час виходу на ринок для нових функцій та виправлень помилок, а також мінімізуючи помилки розгортання.
- DevSecOps: Цей підхід підвищує безпеку шляхом вбудовування практик безпеки та автоматизованого тестування протягом усього життєвого циклу.
- Спостережуваність та моніторинг: Впровадження надійних рішень моніторингу надає інформацію про продуктивність системи в режимі реального часу [4].

DevOps, особливо з глибоким впровадженням ІІІ та акцентом на DevSecOps, є комплексною культурною та технічною трансформацією. Він внутрішньо вбудовує якість та безпеку на кожному етапі життєвого циклу програмного забезпечення, перетворюючи забезпечення якості на безперервну відповідальність між командами.

Незважаючи на значний прогрес у галузі оцінки якості програмного забезпечення, існують помітні обмеження в дослідженнях, які потребують уваги.

- **Методологічні недоліки в емпіричних дослідженнях:** Емпіричні дослідження в інженерії ПЗ часто страждають від низької якості, що проявляється в низькій валідності, невизначеній області застосування та потенційній упередженості [7].
- **Виклики в оцінці систем ІІІ:** Існуючі бенчмарки ІІІ часто зосереджені на вузькому наборі завдань, не завжди надійно вимірюють реальні можливості інженерії ПЗ [14].
- **Розрив між дослідженнями та практикою:** Багато досліджень не мають промислового контексту, що дає розрив між академічними відкриттями та реальністю та їх вплив досліджень на практичні вдосконалення.
- **Етичні проблеми в тестуванні ІІІ:** Зростаюча роль ІІІ в тестуванні програмного забезпечення викликає етичні проблеми, зокрема щодо забезпечення справедливості, підзвітності та прозорості.
- **Складність програмних систем:** Зростаюча складність програмних систем, особливо тих, що інтегрують ІІІ, робить комплексне тестування та забезпечення якості більш складним та дорогим [3]. Це вимагає постійного розвитку нових методів та інструментів, які можуть впоратися зі складністю.

Сучасні тенденції та дослідження в оцінці якості програмного забезпечення мають значні практичні наслідки для організацій, що займаються розробкою програмного забезпечення.

- Стратегічні інвестиції в забезпечення якості, кероване ІІІ: Організації повинні інвестувати в інструменти та фреймворки для тестування на основі ІІІ, щоб прискорити цикли тестування, покращити виявлення дефектів [3].
- Підвищення кваліфікації фахівців із забезпечення якості: Командам із забезпечення якості необхідно підвищувати кваліфікацію в тестуванні, керованому ІІІ та етичних практиках з ІІІ, щоб залишатися актуальними та ефективними [4].

- **Впровадження парадигм безперервної якості:** Впровадження практик Shift-Left, Shift-Right та DevSecOps є вирішальним для вбудовування якості та безпеки протягом усього життєвого циклу розробки програмного забезпечення, забезпечуючи безперервний зворотний зв'язок та стійкість виробництва [5].
- **Індивідуальні підходи до оцінки:** Визнання того, що різні типи програмного забезпечення вимагають специфічних методів оцінки є важливим для ефективного забезпечення якості. Організації повинні адаптувати свої стратегії тестування до унікальних вимог кожного продукту.
- **Акцент на нефункціональних вимогах:** Збільшення акценту на нефункціональному тестуванні є життєво важливим, оскільки ці аспекти значно впливають на користувацький досвід та ділову цінність.
- **Вирішення проблем етичного ШІ:** Пріоретизація етичних аспектів тестування ШІ є необхідною для забезпечення справедливості, прозорості та відповідності, зменшуючи юридичні та репутаційні ризики, пов'язані з упередженими результатами ШІ [5].

Поточний стан дослідницький ландшафт вказує на кілька ключових напрямків для майбутніх досліджень, які матимуть вирішальне значення для подальшого розвитку оцінки якості програмного забезпечення.

- **Покращена оцінка та бенчмаркінг ШІ:** Необхідна розробка більш надійних, різноманітних та відповідних реальному світу бенчмарків для систем ШІ, що вирішуватимуть такі проблеми, як забруднення даних та конструктивна валідність [14]. Дослідження кількісних показників для складних можливостей ШІ, що виходять за рамки генерації коду, також є пріоритетним.
- **Пояснюваний ШІ у тестуванні:** Подальші дослідження щодо забезпечення прозорості та інтерпретованості процесів прийняття рішень ШІ для інженерів програмного забезпечення, включаючи інструменти для узагальнення міркувань ШІ та вирішення суперечливих рекомендацій ШІ [10]. Це допоможе підвищити довіру та прийняття ШІ в критично важливих додатках.
- **Інтеграція ШІ з людським досвідом:** Вивчення оптимальних систем тестування за допомогою ШІ з участю людини в циклі, де ШІ доповнює людські можливості, а не повністю замінює їх.
- **Вирішення етичних проблем ШІ:** Продовження досліджень методів виявлення та пом'якшення упереджень у моделях ШІ, що використовуються для тестування, забезпечення справедливості, підзвітності та відповідності нормативним актам, що розвиваються [4].
- **Емпірична строгість та промислова значущість:** Більший акцент на проведеному високоякісних емпіричних досліджень у промислових контекстах, з більшою кількістю реплікацій та тріангуляцією дослідницьких дизайнів для забезпечення надійності та узагальнюваності висновків [7].
- **Побудова теорії в інженерії програмного забезпечення:** Розробка та тестування більш специфічних для інженерії програмного забезпечення теорій, а також покращення документації та систематизації існуючих теорій для сприяння кумулятивному накопиченню знань [7].

- **ШІ для нових програмних парадигм:** Дослідження оцінки якості для нових програмних парадигм, таких як програми, що працюють на базі фундаментальних моделей (FMware), з урахуванням їх стохастичної природи та унікальних викликів [12].
- **Міжгалузеві та міждоменні розширення:** Вивчення передачі знань та відкритих дослідницьких питань щодо застосування підходів до оцінки якості в різних галузях та доменах [20].

Висновки. Оцінка якості програмного забезпечення переживає період інтенсивної трансформації, зумовленої зростаючою складністю систем, швидкою цифровою трансформацією та повсюдним впровадженням штучного інтелекту. Сучасні підходи виходять за рамки традиційних моделей, інтегруючи інноваційні методології, такі як тестування на основі ШІ, підходи Shift-Left та Shift-Right, а також принципи DevOps. Ці зміни свідчать про фундаментальний зсув у парадигмі якості: від реактивного виявлення дефектів до проактивного, інтелектуального та безперервного забезпечення якості, що вбудовується в кожен етап життєвого циклу розробки програмного забезпечення.

Штучний інтелект, зокрема ВММ, кардинально змінює тестування, автоматизуючи рутинні завдання, покращуючи тестове покриття, забезпечуючи прогнозний аналіз дефектів та можливості самовідновлення. Це переосмислює роль тестувальника, дозволяючи їм зосередитися на більш складних, творчих та стратегічних аспектах забезпечення якості..

Незважаючи на ці досягнення, галузь стикається зі значними викликами. Методологічні недоліки в емпіричних дослідженнях, проблеми з валідністю бенчмарків ШІ та розрив між академічними дослідженнями та промисловою практикою залишаються критичними перешкодами. Крім того, етичні міркування, пов'язані з упередженістю ШІ та прозорістю його рішень, вимагають постійної уваги та досліджень.

Майбутні дослідження повинні зосередитися на розробці більш надійних бенчмарків для ШІ, покращенні пояснюваності систем ШІ, оптимізації взаємодії, а також на подальшій інтеграції етичних практик у тестування ШІ. Також критично важливим є підвищення емпіричної строгості та промислової значущості досліджень, а також розробка нових теорій для інженерії програмного забезпечення.

Таким чином, оцінка якості програмного забезпечення в цифрову епоху вимагає збалансованого, адаптивного та етично свідомого підходу. Успіх майбутніх програмних продуктів залежатиме від здатності індустрії та дослідницької спільноти долати ці виклики, використовуючи потенціал ШІ та безперервних практик для забезпечення надійності, безпеки та довіри до програмного забезпечення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Evidence-Based Software Engineering Guidelines Revisited [Електронний ресурс]. – Режим доступу: URL. – Назва з екрана. – Дата звернення: травня 21, 2025.
2. Software Quality Assurance Market Size & Analysis 2025-2033 [Електронний ресурс]. – Режим доступу: URL. – Назва з екрана. – Дата звернення: 21 травня, 2025.

3. Software Quality Assurance (SQA) Testing Market Size & Forecast [Електронний ресурс]. – Режим доступу: URL. – Назва з екрана. – Дата звернення: 21 травня, 2025.
4. Software Testing in 2025 – Emerging Trends and Technologies - DEV Community [Електронний ресурс]. – Режим доступу: URL. – Назва з екрана. – Дата звернення: 21 травня, 2025.
5. Top Software Testing Trends 2025: AI & ML Take Charge - Techment [Електронний ресурс]. – Режим доступу: URL. – Назва з екрана. – Дата звернення: 21 травня, 2025.
6. Using text mining for research trends in empirical software engineering Deneysel yazılım mühendisliğindeki araştırma eğ - DergiPark [Електронний ресурс]. – Режим доступу: URL. – Назва з екрана. – Дата звернення: 21 травня, 2025.
7. web-backend.simula.no [Електронний ресурс]. – Режим доступу: URL. – Назва з екрана. – Дата звернення: 21 травня, 2025.
8. The top 5 software testing trends for 2025 - Xray Blog [Електронний ресурс]. – Режим доступу: URL. – Назва з екрана. – Дата звернення: 21 травня, 2025.
9. Expectations vs Reality -- A Secondary Study on AI Adoption in Software Testing - arXiv [Електронний ресурс]. – Режим доступу: URL. – Назва з екрана. – Дата звернення: 21 травня, 2025.
10. AIST 2025 - 5th International Workshop on Artificial Intelligence in ... [Електронний ресурс]. – Режим доступу: URL. – Назва з екрана. – Дата звернення: 21 травня, 2025.
11. Top AI Software Testing Services to Consider in 2025 [Електронний ресурс]. – Режим доступу: URL. – Назва з екрана. – Дата звернення: травня 21, 2025.
12. FSE 2025 - Tutorials - FSE 2025 - Conferences - Researchr [Електронний ресурс]. – Режим доступу: URL. – Назва з екрана. – Дата звернення: травня 21, 2025.
13. A-Test 2020 | The A-Test 2025 Workshop [Електронний ресурс]. – Режим доступу: URL. – Назва з екрана. – Дата звернення: травня 21, 2025.
14. Challenges and Paths Towards AI for Software Engineering - arXiv [Електронний ресурс]. – Режим доступу: URL. – Назва з екрана. – Дата звернення: травня 21, 2025.
15. Advancing Software Quality: A Standards-Focused Review of LLM-Based Assurance Techniques - arXiv [Електронний ресурс]. – Режим доступу: URL. – Назва з екрана. – Дата звернення: травня 21, 2025.

REFERENCES

1. Computer Society. (2025). *Evidence-Based Software Engineering Guidelines Revi-sited*. IEEE. <https://www.computer.org/csdl/journal/ts/2025/03/10834515/23lkj7oc6c>.
2. Global Growth Insights. (2025). *Software Quality Assurance Market Size & Analysis 2025-2033*. <https://www.globalgrowthinsights.com/market-reports/software-quality-assurance-market-106845>.
3. Verified Market Research. (2025). *Software Quality Assurance (SQA) Testing Market Size & Forecast*. <https://www.verifiedmarketresearch.com/product/software-quality-assurance-sqa-testing-market/>.
4. Winst, N. (2025). *Software Testing in 2025 — Emerging Trends and Technologies*. DEV Community. <https://dev.to/nicholaswinst14/software-testing-in-2025-emerging-trends-and-technologies-2jp2>.

5. Techment. (2025). *Top Software Testing Trends 2025: AI & ML Take Charge*. <https://www.techment.com/ai-in-software-testing-2025-key-trends/>.
6. DergiPark. (2025). *Using text mining for research trends in empirical software engineering*. <https://dergipark.org.tr/en/download/article-file/1415398>.
7. Simula Research Laboratory. (2025). *Empirical Software Engineering: Challenges and Solutions*. <https://web-backend.simula.no/sites/default/files/publications/Simula.SE.13.pdf>.
8. Xray Blog. (2025). *The top 5 software testing trends for 2025*. <https://www.getxray.app/blog/top-2025-software-testing-trends>.
9. arXiv. (2025). *Expectations vs Reality — A Secondary Study on AI Adoption in Software Testing*. <https://arxiv.org/pdf/2504.04921>.
10. AIST 2025. (2025). *5th International Workshop on Artificial Intelligence in Software Testing*. <https://conf.researchr.org/home/icst-2025/aist-2025>.
11. Frugal Testing. (2025). *Top AI Software Testing Services to Consider in 2025*. <https://www.frugaltesting.com/blog/top-ai-software-testing-services-to-consider-in-2025>.
12. FSE 2025. (2025). *Tutorials - FSE 2025**. <https://conf.researchr.org/track/fse-2025/fse-2025-tutorials>.
13. A-Test. (2025). *The A-Test 2025 Workshop*. <https://a-test.org/a-test-2020/>.
14. arXiv. (2025). *Challenges and Paths Towards AI for Software Engineering*. <https://arxiv.org/html/2503.22625v1>.
15. arXiv. (2025). *Advancing Software Quality: A Standards-Focused Review of LLM-Based Assurance Techniques*. <https://arxiv.org/html/2505.13766v1>.

doi: 10.32403/1998-6912-2025-1-70-72-82

AN OVERVIEW OF MODERN APPROACHES TO SOFTWARE QUALITY ASSESSMENT

I. V. Hileta, R. O. Kiselyk, V. A. Karnauchov, T. N. Komynar

*Lviv Polytechnic National University,
12 Stepana Bandera str., Lviv, 79013, Ukraine
e-mail: ivan.v.hileta@lpnu.ua*

The digital transformation era has elevated software quality from a technical concern to a critical business and societal imperative. This comprehensive review synthesizes advancements in software quality assessment (SQA) methodologies from 2020–2025, addressing the convergence of evolving quality models, AI-driven automation, and organizational paradigms like DevOps. The analysis reveals a paradigm shift from reactive defect detection to proactive, intelligent quality assurance (QA) embedded throughout the software lifecycle.

Traditional frameworks like ISO/IEC 9126 and its successor ISO/IEC 25010 provide structured quality attributes. CMMI remains pivotal for assessing organizational maturity in delivering high-quality software. However, specialized models now address context-

specific needs—such as security-critical systems or early-stage startups—signaling a move beyond one-size-fits-all approaches. This trend underscores the necessity for domain-adaptive quality assessment, where contextual factors shape evaluation criteria.

Artificial Intelligence (AI) and Machine Learning (ML) are redefining SQA: Test Automation, Predictive Capabilities, Self-Healing Tests, Enhanced Non-Functional Testing. DevOps and Shift-Left/Shift-Right testing are foundational to continuous quality.

Software quality assessment is transitioning toward intelligent, continuous, and ethical assurance. AI's predictive power and automation capabilities redefine tester roles—shifting focus from execution to strategic oversight. DevOps and Shift-Left/Right methodologies enable holistic quality cycles. However, methodological rigor in empirical research, ethical AI deployment, and context-aware quality models remain urgent priorities. Success hinges on bridging industry-academia gaps and advancing adaptive, human-centric QA frameworks for resilient, trustworthy software.

Keywords: *Software Quality Assurance, Artificial Intelligence in Testing, DevOps, Quality Models, Shift-Left Testing, Shift-Right Testing, AI Ethics, Predictive Analytics, Empirical Software Engineering, Test Automation, Continuous Quality*

Стаття надійшла до редакції 28.04.2025.

Received 28.04.2025.