



Інформаційно-комунікаційні технології в освіті

УДК 004.738.5:371.331.1:681.3.06

DOI <https://doi.org/10.5281/zenodo.20662785>

Методичні засади інтеграції тестування вебдоступності у підготовку майбутніх фахівців з розробки програмного забезпечення

Кархут Володимир Ярославович,

кандидат педагогічних наук, доцент, доцент кафедри комп'ютерної та програмної інженерії факультету математики, інформатики та фізики Українського державного університету імені Михайла Драгоманова, м. Київ, вул. Пирогова 9, 01601, Україна, v.y.karhut@udu.edu.ua ORCID

<https://orcid.org/0009-0004-9449-6517>

Галицький Олександр Вадимович,

кандидат педагогічних наук, доцент, доцент кафедри комп'ютерної та програмної інженерії факультету математики, інформатики та фізики Українського державного університету імені Михайла Драгоманова, м. Київ, вул. Пирогова 9, 01601, Україна, o.v.galutskyi@udu.edu.ua ORCID

<https://orcid.org/0000-0002-7694-3019>

Прийнято: 15.05.2026 | Опубліковано: 30.05.2026

Анотація. Підготовка конкурентоспроможних фахівців з інженерії програмного забезпечення в умовах євроінтеграції України вимагає системного врахування міжнародних стандартів якості програмних продуктів, зокрема вимог веб-доступності. Тестування доступності є повноправним видом QA-практики, закріпленим у міжнародному стандарті ISO/IEC 25010 та регульованим стандартом WCAG 2.1, однак залишається системно відсутнім як у виробничих процесах розробки, так і у вітчизняних освітніх програмах підготовки відповідних фахівців. У статті здійснено аналіз типових викликів, що виникають у виробничих командах у процесі тестування доступності



програмного забезпечення. Виокремлено сім ключових проблем: відсутність *accessibility*-критеріїв у тест-плануванні та визначенні готовності продукту, пізнє виявлення дефектів доступності, розмитість командної відповідальності, поверхневість розуміння вимог доступності учасниками команди, ризик регресії при виправленні дефектів, хаотичність планування робіт з доступності, а також інструментальна та методологічна некомпетентність у сфері автоматизації тестування. Показано, що зазначені виклики мають системний характер і беруть початок у дефіциті відповідних знань і навичок, що мають формуватися ще на етапі фахової підготовки. На основі аналізу стандарту вищої освіти зі спеціальності 121 «Інженерія програмного забезпечення» виявлено нормативну прогалину: стандарт містить компетентності та результати навчання, що логічно передбачають знання з доступності - зокрема K16, K17, ПР14, ПР19 та ПР20 - однак жодного прямого згадування WCAG чи *accessibility*-тестування не містить. Аналіз навчальних програм провідних університетів України підтвердив, що відповідні QA-дисципліни наявні в усіх розглянутих закладах, проте тематика тестування доступності у їхньому змісті відсутня. За результатами дослідження запропоновано систему методичних підходів до інтеграції тестування доступності у підготовку фахівців, що ґрунтується на принципі прямої відповідності між викликами виробничого середовища та навчальними результатами. Для кожного з виявлених викликів визначено конкретний методичний підхід, відповідну дисципліну та формат реалізації - від формування *acceptance criteria* з WCAG-вимогами і практичних сесій із залученням допоміжних технологій до побудови спеціалізованих регресійних тест-наборів. Подальші дослідження спрямовуватимуться на розроблення моделі компетентностей фахівця з тестування доступності та практичну апробацію запропонованих підходів у навчальному процесі.

Ключеві слова: Веб-доступність, WCAG, підготовка розробників програмного забезпечення, європейські стандарти



Methodological foundations for integrating web accessibility testing into the training of future software development specialists

Volodymyr Karkhut

candidate of pedagogical sciences, associate professor, associate professor of the department of computer and software engineering faculty of mathematics, informatics and physics Mykhailo Drahomanov Ukrainian State University, Kyiv, St. Pirohova 9, 01601, Ukraine, v.y.karkhut@udu.edu.ua ORCID <https://orcid.org/0009-0004-9449-6517>

Oleksandr Halytskyi

candidate of pedagogical sciences, associate professor, associate professor of the department of computer and software engineering faculty of mathematics, informatics and physics Mykhailo Drahomanov Ukrainian State University, Kyiv, St. Pirohova 9, 01601, Ukraine, o.v.galutskyi@udu.edu.ua ORCID <https://orcid.org/0000-0002-7694-3019>

Abstract. Training competitive software engineers in Ukraine requires integrating international standards like WCAG 2.1 and ISO/IEC 25010 into the curriculum. Despite its importance, accessibility testing is often absent from both industry practices and educational programs. This study identifies seven systemic industrial challenges: lack of accessibility in "Definition of Done," late defect detection, blurred responsibility, superficial requirement knowledge, regression risks, chaotic planning, and tool incompetence. Analysis of the 121 "Software Engineering" state standard reveals a regulatory gap: while competencies and outcomes imply quality standards, accessibility is never explicitly mentioned. University curricula reviews confirm this topic's absence in QA courses. This paper proposes a methodological framework mapping industrial challenges to specific learning

outcomes, utilizing practical sessions with assistive technologies and WCAG-based acceptance criteria. Future research will focus on a formal competency model for accessibility testers.

Keywords: web accessibility, WCAG, software engineering education, inclusive design, digital inclusion.

Постановка проблеми. Підготовка конкурентноспроможних фахівців з інженерії програмного забезпечення полягає у суперечності між зростаючими вимогами міжнародних стандартів якості програмного забезпечення щодо забезпечення вебдоступності та фактичною відсутністю системної підготовки майбутніх фахівців з інженерії програмного забезпечення і QA до тестування доступності. Попри нормативне закріплення вимог доступності у стандартах ISO/IEC 25010 та WCAG 2.1, а також наявність у стандарті вищої освіти спеціальності 121 «Інженерія програмного забезпечення» компетентностей, дотичних до забезпечення якості програмних продуктів, освітні програми українських закладів вищої освіти не забезпечують формування практичних знань і навичок з accessibility-тестування. Це призводить до виникнення системних проблем у виробничому середовищі: пізнього виявлення дефектів доступності, відсутності критеріїв accessibility у QA-процесах, нечіткого розподілу відповідальності та недостатньої методологічної й інструментальної підготовки команд. Унаслідок цього виникає потреба в науково обґрунтованих методичних підходах до інтеграції тестування доступності у професійну підготовку майбутніх фахівців з інженерії програмного забезпечення.

Аналіз останніх досліджень і публікацій. Проблема підготовки фахівців до роботи в інклюзивному цифровому середовищі стає предметом прискіпливої уваги наукової спільноти. У роботі В. Кархут, О. Галицький [1] було здійснено системний аналіз освітніх програм спеціальності 121 «Інженерія програмного забезпечення» у провідних університетах України та визначено перелік дисциплін, що є пріоритетними для інтеграції знань з вебдоступності. Було

встановлено, що попри наявність технічної бази, методичне забезпечення підготовки майбутніх розробників до верифікації інклюзивних інтерфейсів потребує суттєвого оновлення.

Дослідження К. Z. Zhou, С. Liu, J. Shan та ін. [2] демонструє ефективність використання ігрових методів та емпатійного підходу у навчанні основ доступності; автори доводять, що впровадження таких методик дозволяє суттєво підвищити рівень обізнаності та розуміння потреб людей з інвалідністю у майбутніх дизайнерів та розробників програмного забезпечення, що є критично важливим для подолання існуючого дефіциту знань у галузі інклюзивного проєктування.

У дослідженні Y. N. El-Glaly [3] представлено досвід розроблення та впровадження вибіркової навчальної дисципліни з вебдоступності для магістрантів спеціальності «Програмна інженерія»; автором деталізовано результати трьох ітерацій викладання курсу, визначено ключові результати навчання та стратегії оцінювання, що створює практичне підґрунтя для інтеграції аналогічних модулів у сучасні освітні програми підготовки фахівців з ІТ.

У систематичному дослідженні С. М. Baker, Y. N. El-Glaly та К. Shinohara [4] проаналізовано сучасний стан викладання дисциплін із цифрової доступності в межах комп'ютерних наук; автори встановили, що лише приблизно 33% навчальних програм розглядають доступність як окрему тему, а також виявили суттєвий брак інструментарію та методичних ресурсів для викладачів, що підтверджує необхідність розроблення нових підходів до формування відповідних компетентностей у майбутніх ІТ-фахівців.

У дослідженні А. R. Mohamad Syafiq та ін. [5], за допомогою методу MANOVA та інтерв'ю було проаналізовано сприйняття стандартів вебдоступності майбутніми розробниками; результати свідчать про критично низький рівень обізнаності студентів із настановами WCAG, водночас автори доводять, що ключовими чинниками формування інклюзивних практик у

розробці є цілеспрямована освіта, безпосередній контакт із людьми з інвалідністю та розвиток емпатії у процесі навчання.

Ж. Р. Рол [6] представив результати порівняльного аналізу дев'яти сучасних інструментів тестування вебдоступності, що використовуються командами із забезпечення якості автор наголошує на важливості поєднання автоматизованих та напівавтоматизованих підходів до валідації інклюзивності, що безпосередньо вказує на перелік технічних компетенцій, які мають бути сформовані у майбутніх тестувальників у процесі професійної підготовки.

Т. Бондаренко аналізує роль інформаційно-комунікаційних технологій у забезпеченні доступності та розвитку інклюзивної освіти. Вона показує, що ІКТ стають ключовим фактором подолання освітніх бар'єрів для осіб з особливими потребами, проте наголошує на необхідності формування відповідних компетентностей у майбутніх педагогів [7]. У роботі В. Ю. Клим та Ю. С. Тарасенко [8] обґрунтовано методіку тестування вебсайтів на інклюзивність для користувачів із порушеннями зору та слуху. В. Биков, О. Спирін та О. Пінчук (2017) відзначали потребу реалізації нового методичного підходу до питань доступності [9].

Попри розвинену міжнародну дослідницьку базу з питань навчання веб-доступності у середній та вищій школі, вітчизняні публікації з даної тематики фактично відсутні, що підкреслює актуальність цього дослідження в українському освітньому контексті.

Метою статті полягає в аналізі сучасного стану інтеграції вебдоступності у навчальний процес підготовки фахівців з розробки та тестування програмного забезпечення – з акцентом на аспекті забезпечення якості. Дослідження охоплює огляд чинних освітніх стандартів, навчальних програм профільних дисциплін та міжнародних галузевих практик з метою виявлення системних прогалин у підготовці майбутніх тестувальників щодо перевірки програмних продуктів на відповідність стандартам доступності. Спираючись на аналіз реальних викликів, притаманних процесам розробки та тестування у

професійному середовищі, пропонуються науково обґрунтовані методичні підходи до вдосконалення навчальних програм — зокрема, через включення тестування вебдоступності як окремого наскрізного елементу підготовки фахівця з якості програмного забезпечення.

Методи дослідження. Для виконання поставлених завдань було використано теоретичні та емпіричні методи наукових досліджень, а саме: метод системного аналізу, порівняння та узагальнення для теоретичного обґрунтування й розроблення теоретико-методичного супроводу професійної підготовки відповідно до її функцій та ролі у структурі фахової підготовки майбутнього фахівця з розробки програмного забезпечення; бібліосемантичний метод – для вивчення психолого-педагогічної, наукової літератури, нормативних документів з питань формування професійної компетентності здобувачів вищої освіти спеціальності 121 «Інженерія програмного забезпечення»; емпіричні методи – бесіди зі здобувачами вищої освіти та науково-педагогічними працівниками, аналіз способів змістових підходів до підготовки фахівців з розробки програмного забезпечення.

Результати досліджень. Забезпечення якості програмного забезпечення є одним із ключових завдань сучасної інженерії програмного забезпечення та охоплює широкий спектр характеристик продукту - від функціональної правильності, швидкодії чи то зручності використання. Міжнародний стандарт якості програмних продуктів ISO/IEC 25010 визначає модель якості, що включає вісім основних характеристик, серед яких ми бачимо придатність до використання а також містить підхарактеристику доступності - здатності продукту бути використаним людьми з різними фізичними, сенсорними або когнітивними особливостями. Таким чином, доступність є не просто опцією, а невід'ємним стандартизованим виміром якості програмного продукту.

Головним практичним інструментом забезпечення веб-доступності є принципи з доступності веб-контенту WCAG, розроблені W3C. Стандарт WCAG 2.2 структурований навколо чотирьох принципів - сприйнятності,

керованості, зрозумілості та надійності (POUR) - і визначає три рівні відповідності: А, АА та ААА. Рівень АА наразі є де-факто галузевим мінімумом і закріплений у законодавстві низки країн як обов'язкова вимога до публічних цифрових сервісів. В Україні відповідний державний стандарт ДСТУ EN 301 549 також орієнтується на вимоги WCAG. Отже, дотримання стандартів доступності є не лише етичним зобов'язанням, а й правовою та контрактною вимогою до програмних продуктів.

Попри це, тестування доступності залишається системно недооціненою практикою у процесах забезпечення якості. Дослідження засвідчують, що переважна більшість веб-продуктів не відповідає навіть базовому рівню WCAG: за даними WebAIM Million Report (2024), 95,9% із мільйона найпопулярніших вебсайтів містять щонайменше одне порушення рівня А [12]. Це свідчить про те, що проблема має не технічний, а системний характер - вона закорінена у процесах розробки, планування та підготовки фахівців. Тестування доступності є окремим видом QA-практики, що охоплює як автоматизовану перевірку за допомогою спеціалізованих інструментів (axe, WAVE, Lighthouse), так і ручне тестування із залученням допоміжних технологій - зокрема, програм зчитування екрану та навігації з клавіатури [11]. Відповідно, підготовка фахівців з тестування ПЗ має включати як теоретичне розуміння стандартів доступності, так і практичні навички їх верифікації - що на сьогодні залишається невирішеним завданням вітчизняної системи вищої освіти.

Аналіз сучасних практик розробки програмного забезпечення дозволяє виокремити ряд системних проблем, що перешкоджають ефективній інтеграції тестування доступності у виробничий процес. Ці проблеми не є випадковими або поодинокими - вони відтворюються у командах різного масштабу і свідчать про відсутність сформованої культури доступності як складової забезпечення якості [4]. Нижче розглянуто ключові виклики, ідентифіковані у контексті реальних процесів розробки та підтверджені профільними дослідженнями.

Виклики реалізації доступності в професійній розробці:



1. Однією з найпоширеніших проблем є системне виключення вимог доступності зі складу критеріїв якості на етапі планування. У більшості команд accessibility-критерії відсутні як у визначенні готовності, так і у тест-кейсах і acceptance criteria окремих завдань [4]. Це означає, що QA-фахівець апріорі не має інструкції щодо того, що саме перевіряти і відповідно до якого стандарту. За таких умов тестування доступності або не відбувається взагалі, або здійснюється ситуативно - на розсуд конкретного фахівця і поза межами формального процесу.

2. Навіть у командах, де тестування доступності практикується, воно, як правило, зміщено на пізні етапи життєвого циклу - зокрема, на фазу приймального тестування або виявляється вже після релізу продукту. Цей феномен суперечить принципу «зміщення вліво» (shift-left testing), відповідно до якого виявлення дефектів на ранніх стадіях - рівні вимог, проєктування або написання коду - суттєво знижує вартість їх усунення [4]. Стосовно дефектів доступності це особливо критично: структурні порушення семантики HTML або архітектурно некоректне застосування ARIA-атрибутів на пізніх етапах потребують переопрацювання компонентів, а не точкових виправлень.

3. Характерною особливістю командних процесів є відсутність чіткого розподілу відповідальності за доступність між учасниками розробки. На практиці розробник вважає питання доступності прерогативою дизайнера, дизайнер - розробника, тестувальник - обох. Результатом є так звана «нічия зона», де жоден учасник не відчуває персональної відповідальності за стан доступності продукту.

4. Навіть за умови фіксації дефектів доступності у баг-трекері, їхнє виправлення нерідко має суто формальний характер. Типовим прикладом є додавання атрибуту alt до зображення з нерелевантним або автоматично згенерованим текстом, що технічно усуває помилку валідатора, але не вирішує проблему користувача з порушенням зору. Подібна поверхневність виникає через відсутність у команди розуміння функціонального призначення вимог

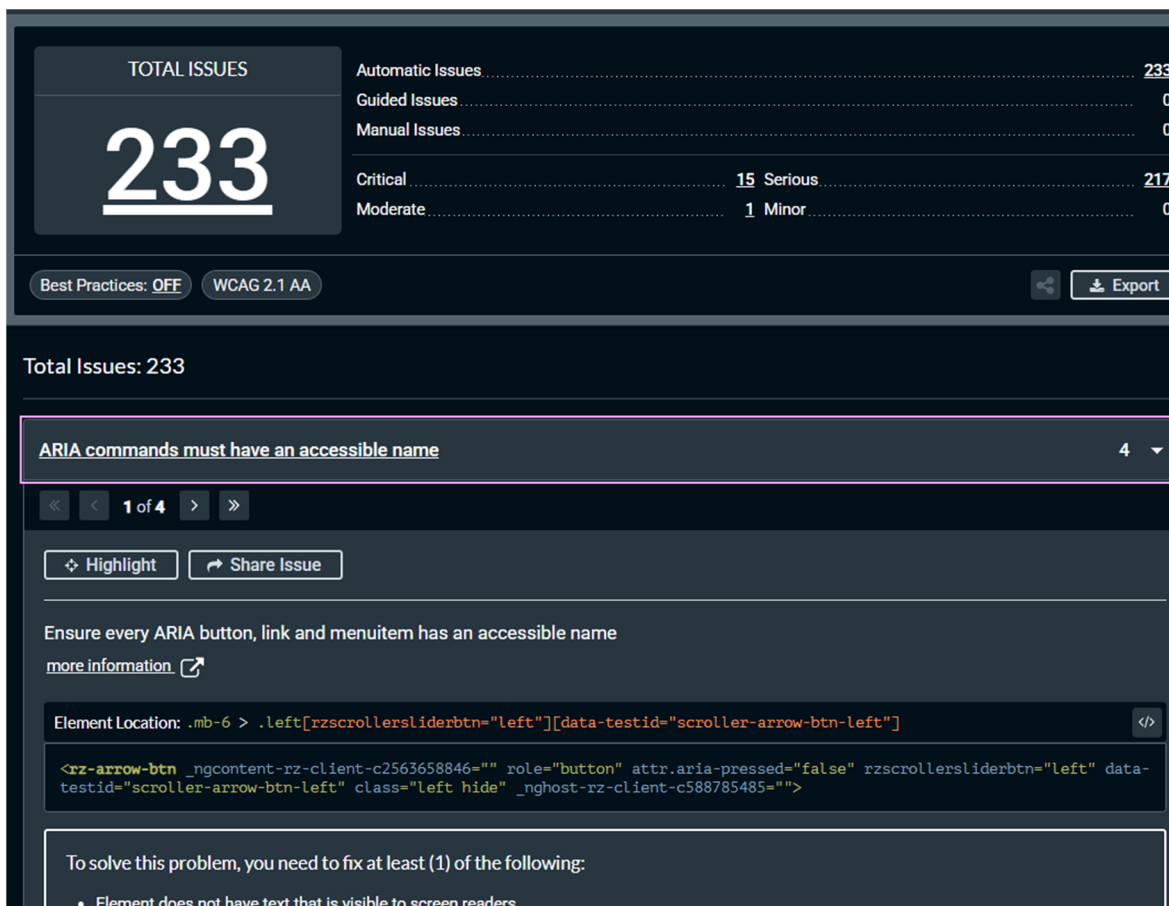
доступності - тобто того, яким чином конкретна технічна характеристика впливає на реальний користувацький досвід людей із різними обмеженнями. Це перетворює роботу над доступністю на формальне виконання чеклісту замість осмисленого проектування інклюзивного продукту.

5. Специфічною технічною проблемою є висока вірогідність виникнення регресії при усуненні accessibility-дефектів. На відміну від ізольованих функціональних помилок, порушення доступності часто мають каскадний характер: зміна ARIA-ролі одного компонента, коригування порядку фокусу або реструктуризація DOM-дерева можуть порушити доступність суміжних елементів інтерфейсу. За відсутності спеціалізованих регресійних сценаріїв для accessibility подібні побічні ефекти залишаються непоміченими до наступного циклу тестування або скарги користувача.

6. Суттєвою операційною проблемою є відсутність системного підходу до планування робіт з доступності. У командах, де доступність розглядається виключно як набір знайдених дефектів, а не як проактивна характеристика якості, виникає ряд закономірних наслідків: дублювання однотипних звітів у баг-трекері, конфлікти між паралельними виправленнями, втрата пріоритизації між критичними та другорядними проблемами. Це призводить до неефективного використання ресурсів команди і знижує загальну ефективність процесу забезпечення якості.

7. Тестування доступності висуває специфічні вимоги до автоматизованих тестових сценаріїв, які суттєво відрізняються від стандартних end-to-end або UI-тестів. Зокрема, повноцінне тестування доступності потребує сценаріїв навігації клавіатурою, перевірки коректності оголошень screen reader, валідації порядку фокусу та контрастних співвідношень у динамічних станах компонентів. QA-фахівці, які не мають відповідної підготовки, не лише не використовують accessibility-інструменти, а й не усвідомлюють меж їхніх можливостей - що формує хибне відчуття повноти перевірки.

Таким чином, виокремлені виклики мають взаємопов'язаний характер і утворюють системну проблему, вирішення якої не може бути зведене до точкових технічних заходів. Кожна з описаних проблем - від відсутності критеріїв у тест-плануванні до інструментальної некомпетентності - бере початок у дефіциті базових знань і практичних навичок, що мають формуватися ще на етапі фахової підготовки. Сліпа довіра до інструментів автоматизованої перевірки дозволяє знаходити велику кількість проблем (Рис.1) та не дозволяє добитись повного покриття проблем. Це підтверджує необхідність системного перегляду змісту освітніх програм у частині тестування та забезпечення якості програмного забезпечення.



TOTAL ISSUES	
Automatic Issues	233
Guided Issues	0
Manual Issues	0
Critical	15
Serious	217
Moderate	1
Minor	0

Best Practices: OFF WCAG 2.1 AA Export

Total Issues: 233

ARIA commands must have an accessible name 4

1 of 4

Highlight Share Issue

Ensure every ARIA button, link and menuitem has an accessible name
[more information](#)

Element Location: `.mb-6 > .left[rzscrollersliderbtn="left"][data-testid="scroller-arrow-btn-left"]`

```
<rz-arrow-btn _ngcontent-rz-client-c2563658846="" role="button" attr.aria-pressed="false" rzscrollersliderbtn="left" data-testid="scroller-arrow-btn-left" class="left hide" _ngghost-rz-client-c588785485="">
```

To solve this problem, you need to fix at least (1) of the following:

- Element does not have text that is visible to screen readers

Рис. 1. Приклад автоматизованої перевірки доступності через плагін

Виокремлені вище розділі виклики виробничого середовища закономірно постають як питання до системи фахової підготовки: чи формує вітчизняна вища освіта у майбутніх QA-фахівців компетентності, необхідні для їх подолання. Відповідь на це питання потребує аналізу двох рівнів нормативного



регулювання - стандарту вищої освіти зі спеціальності 121 «Інженерія програмного забезпечення» та конкретних освітніх програм провідних університетів.

Стандарт вищої освіти: наявні передумови та нормативна прогалина. Стандарт вищої освіти України зі спеціальності 121 «Інженерія програмного забезпечення» містить перелік компетентностей і результатів навчання, які, при уважному розгляді, утворюють нормативне підґрунтя для включення тематики доступності до навчального процесу - однак жодного разу не згадують її явно.

Серед спеціальних компетентностей безпосередньо релевантними є K16 - «здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами» - і K17 - «здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі». Обидві компетентності логічно охоплюють WCAG як міжнародний галузевий стандарт і ДСТУ EN 301 549 як його національний відповідник, проте жодна з них не конкретизує доступність як обов'язковий вимір якості. Додатково релевантними є K08 («здатність діяти на основі етичних міркувань») та K10 («здатність діяти соціально відповідально»), що утворюють ціннісне підґрунтя для розуміння інклюзивності як професійної норми, а не лише технічної вимоги [13].

На рівні результатів навчання найбільш значущими є ПР19 - «знати та вміти застосовувати методи верифікації та валідації програмного забезпечення» - і ПР20 - «знати підходи щодо оцінки та забезпечення якості програмного забезпечення». Саме ці результати є прямою нормативною основою для включення тестування доступності у відповідні навчальні дисципліни. Не менш важливим є ПР14 - «застосовувати інструментальні програмні засоби тестування» - яке при розширеному тлумаченні охоплює спеціалізовані accessibility-інструменти. Окремо варто відзначити ПР08 - «вміти розробляти людино-машинний інтерфейс» - і ПР04 - «знати і застосовувати професійні



стандарти» - як результати, що безпосередньо пов'язані з доступним проектуванням інтерфейсів і дотриманням WCAG відповідно [13].

Таким чином, стандарт містить щонайменше шість результатів навчання і шість компетентностей, які логічно передбачають знання з доступності - але жодного прямого згадування WCAG, accessibility-тестування чи інклюзивного проектування у тексті стандарту немає. Це є ключовою нормативною прогалиною: доступність *de facto* охоплюється стандартом, але *de jure* залишається невидимою - і отже, необов'язковою для університетів при формуванні навчальних планів.

Навчальні програми університетів: фокус на дисциплінах якості та тестування. У попередньому дослідженні авторів було проаналізовано навчальні плани спеціальності «Інженерія програмного забезпечення» у чотирьох провідних університетах України - Львівській політехніці, КНУ, КПІ та ХНУРЕ [14; 15; 16; 17]. У межах поточної роботи ми звужуємо фокус до дисциплін, безпосередньо пов'язаних із забезпеченням якості та тестуванням програмного забезпечення, оскільки саме вони є природним методичним майданчиком для інтеграції тематики доступності.

Аналіз засвідчує, що в усіх розглянутих програмах присутні дисципліни відповідного профілю, однак їхні назви, обсяг і акцентуація суттєво відрізняються між університетами. Так, у ХНУРЕ наявний окремий курс «Інструментальні засоби автоматизації тестування», що створює особливо сприятливі умови для практичного опрацювання accessibility-інструментів. У КПІ та КНУ тематика верифікації та валідації охоплюється в межах ширших курсів з якості та методологій розробки. В усіх програмах наявний курс або модуль, присвячений аналізу вимог, де доступність може бути введена як обов'язкова категорія нефункціональних вимог.

Показово, що жодна з виявлених дисциплін не включає тестування доступності як окрему тему - ані в описі курсу, ані в переліку лабораторних робіт чи оцінювальних завдань. Це означає, що навіть за наявності відповідних

дисциплін тематика залишається поза навчальним процесом через відсутність методичного забезпечення та, вочевидь, через низьку обізнаність викладачів із даною проблематикою.

Таблиця 1. QA-дисципліни та потенціал інтеграції тестування доступності

Дисципліна	Наявність	Потенціал	Зв'язок з викликами
Якість та тестування ПЗ	Всі університети	Високий	1, 3, 6
Інструментальні засоби автоматизації тестування	ХНУРЕ	Високий	7
Аналіз вимог до ПЗ	Всі університети	Середній	1, 4
Методи верифікації та валідації	КПІ, КНУ	Середній	2, 5
Архітектура ПЗ / Конструювання ПЗ	Всі університети	Низький	5

Підсумовуючи, можна стверджувати, що і на рівні нормативного стандарту, і на рівні конкретних навчальних програм існує структурна невідповідність між реальними вимогами виробничого середовища - зафіксованими вище та змістом підготовки фахівців. Ця невідповідність є не випадковою, а системною: вона відтворюється в усіх розглянутих університетах і не може бути усунена зусиллями окремого викладача чи кафедри без цілеспрямованих методичних рішень.

Виявлені у попередніх розділах виклики виробничого середовища та прогалини освітніх програм дозволяють сформулювати конкретні методичні підходи до інтеграції тестування доступності у підготовку майбутніх фахівців з інженерії програмного забезпечення. Ключовим принципом запропонованого підходу є відповідність між реальними проблемами індустрії та навчальними результатами, які має сформувати дисципліна. Інакше кажучи, кожен методичний елемент обґрунтовується не абстрактно, а через конкретний виклик, з яким стикається фахівець у виробничому середовищі.

Першочерговим завданням є формування у студентів розуміння доступності як повноправного критерію якості програмного продукту, а не додаткової або факультативної характеристики. Це завдання найбільш органічно вирішується у межах курсів «Якість та тестування програмного забезпечення» і «Аналіз вимог до програмного забезпечення».

На рівні вимог студенти мають навчитися формулювати acceptance criteria, що включають WCAG умови прийняття - наприклад, контраст, навігація з клавіатури. Відповідна тема може бути введена у лекційний блок, присвячений нефункціональним вимогам, із подальшим практикумом у форматі написання тест-кейсів для конкретного інтерфейсного компонента. Такий підхід безпосередньо адресує виклик 1 - відсутність accessibility-критеріїв у тест-плануванні - і формує звичку включати доступність у Definition of Done ще на етапі навчання.

Додатково у межах цих курсів доцільно розглянути методологію класифікації та пріоритизації дефектів доступності - зокрема, розмежування між блокуючими порушеннями (неможливість навігації клавіатурою, відсутність текстових альтернатив) і менш критичними (декоративні елементи без ARIA-атрибутів). Це адресує виклик 6 - хаотичне планування і дублювання дефектів у баг-трекері.

Принцип «зміщення вліво» є одним із фундаментальних у сучасній методології тестування, однак щодо доступності він рідко реалізується навіть у досвідчених командах. У навчальному процесі цей принцип може бути впроваджений через поетапне ознайомлення студентів із тим, як і на якому етапі розробки виявляються різні категорії порушень доступності.

Практичним форматом є лабораторна робота, побудована за принципом «знайди і поясни»: студентам пропонується готовий фрагмент коду або розгорнутий веб-компонент, і вони мають ідентифікувати порушення доступності до запуску будь-яких автоматизованих інструментів - тобто на рівні code review. Після цього ті самі порушення виявляються автоматизованим інструментом, і студенти порівнюють результати. Такий формат наочно демонструє, що частина проблем діагностується вже на рівні HTML-структури, і не потребує дорогого UAT-тестування. Це формує практичне розуміння shift-left не як абстрактного принципу, а як конкретної техніки роботи.

Інструментальна компетентність є необхідною передумовою для будь-якої практичної QA-діяльності у сфері доступності. Курс «Інструментальні засоби автоматизації тестування» є найбільш природним місцем для її формування, однак відповідна тема може бути інтегрована і у загальний курс тестування у вигляді окремої лабораторної роботи.

Мінімальний інструментальний стек, з яким має бути ознайомлений студент, включає: браузерне розширення ахе DevTools для автоматизованої перевірки DOM; Google Lighthouse для аудиту доступності у контексті загальних показників якості продукту; WAVE як альтернативний візуальний інспектор; та щонайменше один screen reader - NVDA (Windows) або VoiceOver (macOS/iOS) - для ручного тестування із залученням допоміжних технологій. При цьому студенти мають чітко розуміти обмеження кожного інструменту: зокрема, що автоматизовані засоби виявляють лише близько 30–40% порушень, тоді як перевірка реального користувацького досвіду потребує ручного тестування.

Важливим методичним акцентом є завдання на виявлення хибнонегативних результатів - тобто ситуацій, коли автоматизований інструмент не фіксує порушення, яке реально існує. Це формує критичне ставлення до результатів автоматизованої перевірки і усвідомлення меж її застосування, що безпосередньо адресує виклик 7.

Виклик розмитості відповідальності (3) є організаційним за своєю природою і не може бути усунутий лише технічними знаннями. Його вирішення у навчальному процесі потребує формування у студентів розуміння того, як розподіляється відповідальність за доступність між різними учасниками команди на різних етапах розробки.

Ефективним форматом є рольовий практикум у межах проєктного завдання, де студенти виконують різні командні ролі - розробника, тестувальника, аналітика - і спільно опрацьовують один і той самий компонент з точки зору доступності. Розробник забезпечує семантичну розмітку і ARIA-

атрибути, аналітик включає accessibility-критерії у вимоги, тестувальник складає і виконує тест-кейси. Такий формат моделює реальну командну взаємодію і дозволяє студентам на практиці усвідомити, що доступність є наскрізною відповідальністю, а не задачею однієї ролі.

Поверхневість розуміння вимог доступності (4) виникає передусім через відсутність у фахівців емпатії до досвіду користувача з обмеженими можливостями. Дослідження у сфері освіти доступності переконливо свідчать, що найбільш ефективним способом подолання цієї проблеми є безпосередній контакт студентів із реальним користувацьким досвідом.

Практично це може бути реалізоване у форматі сесії тестування, де студент намагається виконати типові завдання на навчальному або реальному вебсайті виключно за допомогою клавіатури або із ввімкненим screen reader - без використання миші та зорового сприйняття інтерфейсу у звичному режимі.

Виклик регресії при виправленні дефектів (2.5) потребує формування у студентів навичок побудови спеціалізованих регресійних сценаріїв для доступності. У межах курсу автоматизованого тестування це може бути реалізоване як фінальне проєктне завдання: студенти отримують вебзастосунок із зафіксованими порушеннями доступності, виправляють їх і будують мінімальний regression suite, що верифікує як виправлення, так і відсутність нових порушень у суміжних компонентах.

Такий формат інтегрує одразу кілька навичок: технічну роботу з axe-core або Playwright у контексті accessibility assertions, розуміння каскадного характеру DOM-структури та ARIA-залежностей, а також методологію побудови тестових наборів. Завдання може виконуватись індивідуально або у парах, що додатково моделює реальний робочий контекст.

Узагальнення запропонованих підходів наведено у таблиці 2, де кожен методичний елемент співвіднесено з відповідним викликом індустрії, дисципліною та форматом реалізації.

Запропоновані підходи не передбачають введення окремої дисципліни з веб-доступності - що є реалістичним з огляду на перевантаженість навчальних планів - а натомість орієнтовані на інтеграцію відповідних тем і форматів у вже існуючі курси. Розвиток альтернативних напрямків забезпечення доступності будуть розглянуті наших наступних роботах.

Таблиця 2. Відповідність методичних підходів викликам індустрії

Виклик	Методичний підхід	Дисципліна	Формат
2.1 Відсутність критеріїв	Acceptance criteria з WCAG	Аналіз вимог / Якість ПЗ	Лекція та практикум
2.2 Пізні виявлення	Shift-left: code review для a11y	Тестування ПЗ	Лабораторна робота
2.3 Розмитість відповідальності	Рольовий командний практикум	Якість ПЗ	Проектне завдання
2.4 Поверхневість розуміння	Empathy testing / screen reader	Будь-яка QA-дисципліна	Практична сесія
2.5 Ризик регресії	Побудова regression suite	Автоматизація тестування	Проектне завдання
2.6 Дублювання дефектів	Класифікація та пріоритизація	Якість ПЗ	Лекція
2.7 Інструментальна некомпетентність	Практична робота з axe, WAVE, NVDA	Автоматизація тестування	Лабораторна робота

Висновки. У результаті проведеного дослідження встановлено, що тестування веб-доступності є системно відсутнім елементом як у виробничих процесах розробки програмного забезпечення, так і у вітчизняних освітніх програмах підготовки відповідних фахівців. Аналіз реальних практик виробничих команд дозволив виокремити сім взаємопов'язаних викликів - від відсутності accessibility-критеріїв у тест-плануванні та пізнього виявлення дефектів до інструментальної некомпетентності та ризиків регресії - які у сукупності свідчать про відсутність сформованої культури доступності як складової забезпечення якості. Паралельний аналіз стандарту вищої освіти зі спеціальності 121 «Інженерія програмного забезпечення» виявив нормативну прогалину: стандарт містить щонайменше шість компетентностей і шість результатів навчання, що логічно передбачають знання з доступності - зокрема



K16, K17, PR14, PR19 та PR20 - однак жодного прямого згадування WCAG, accessibility-тестування чи інклюзивного проектування у його тексті немає. Це означає, що доступність *de facto* охоплюється стандартом, але *de jure* залишається необов'язковою для університетів при формуванні навчальних планів.

На основі виявленої відповідності між викликами індустрії та прогалинами освітніх програм запропоновано систему методичних підходів до інтеграції тестування доступності у підготовку фахівців. Запропоновані підходи не передбачають введення окремої дисципліни, а орієнтовані на органічну інтеграцію у вже існуючі курси - передусім «Якість та тестування програмного забезпечення», «Інструментальні засоби автоматизації тестування» та «Аналіз вимог до програмного забезпечення». Для кожного з семи виявлених викликів визначено конкретний методичний підхід, відповідну дисципліну та формат реалізації - від лекційних блоків з формування *acceptance criteria* до рольових практикумів, *empathy*-сесій зі *screen reader* та побудови спеціалізованих регресійних тест-наборів. Такий підхід відповідає принципу наскрізної доступності і є реалістичним з огляду на обмеження навчальних планів.

Подальшим напрямом дослідження є розроблення моделі компетентностей фахівця з тестування доступності, яка дозволить системно описати знання, вміння та навички, що мають бути сформовані у процесі підготовки, і стане основою для розроблення конкретних навчальних програм та оцінювальних інструментів. Окремим завданням є практична апробація запропонованих методичних підходів у реальному навчальному процесі з подальшим аналізом їх ефективності - зокрема, у форматі педагогічного експерименту із застосуванням кількісних та якісних методів оцінювання результатів навчання. Реалізація цих завдань сприятиме системному подоланню розриву між вимогами сучасного виробничого середовища та рівнем підготовки випускників у сфері інженерії програмного забезпечення.

Список використаних джерел

1. В. Кархут, О. Галицький, ІНТЕГРАЦІЯ ПРИНЦИПІВ ВЕБДОСТУПНОСТІ В ПІДГОТОВКУ МАЙБУТНІХ РОЗРОБНИКІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ. (2025). Науковий часопис Українського державного університету імені Михайла Драгоманова Серія 2 Комп'ютерно-орієнтовані системи навчання, 24 (31), 109-117. [https://doi.org/10.31392/UDU-nc.series2.2025.24\(31\).10](https://doi.org/10.31392/UDU-nc.series2.2025.24(31).10)(дата звернення: 03.05.2025).
2. Zhou, K. Z., Liu, C., Shan, J., Kletenik, D., & Adler, R. F. (2024). Accessible adventures: Teaching accessibility to high school students through games. *Proceedings of the Association for Information Science and Technology*, 61, 457–468. <https://doi.org/10.1002/pras.1042>(дата звернення: 03.05.2025).
3. El-Glaly, Y. N. (2020). Teaching accessibility to software engineering students. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)* (pp. 121–127). ACM. <https://doi.org/10.1145/3328778.3366914>(дата звернення: 03.05.2025).
4. C. M. Baker, Y. N. El-Glaly, and K. Shinohara, «A systematic analysis of accessibility in computing education research», у *Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20)*, 2020, pp. 107–113. DOI: <https://doi.org/10.1145/3328778.3366843>(дата звернення: 03.05.2025).
5. Mohamad Syafiq, A. R., et al. (2022). Perceptions of web accessibility guidelines by student website and app developers. *Behaviour & Information Technology*, 41(12). <https://doi.org/10.1080/0144929X.2021.1940278>
6. Pool, J. R. (2023). Accessibility metatesting: Comparing nine testing tools. In *Proceedings of the 20th International Web for All Conference (W4A '23)* (Article 4, pp. 1–4). ACM. <https://doi.org/10.1145/3587281.3587282>
7. Бондаренко Т. Використання інформаційно-комунікаційних технологій для забезпечення доступності і розвитку інклюзивної освіти.



Information technologies and learning tools. 2018. Т. 67, № 5. С. 31–43. DOI: <https://doi.org/10.33407/itlt.v67i5.2241> (дата звернення: 07.09.2025).

8. В. Ю. Клим і Ю. С. Тарасенко, «ТЕСТУВАННЯ ВЕБ-РЕСУРСІВ НА ІНКЛЮЗИВНІСТЬ: АУДИТ ДЛЯ РЕІНЖИНІРИНГУ », Вісник ВПІ, вип. 3, с. 60–66, Черв. 2022.

9. Биков В., Спірін О., Пінчук О. Проблеми та завдання сучасного етапу інформатизації освіти. Наукове забезпечення розвитку освіти в Україні: актуальні проблеми теорії і практики (до 25-річчя НАПН України). 2017. С. 191–198.

10. "Digital Accessibility Foundations", *W3C (W3Cx)*. [Online]. Available: <https://www.w3.org/WAI/courses/foundations-course/> (Accessed: Sep. 7, 2025) (in English). (дата звернення: 03.05.2025).

11. "Accessibility courses", *Deque University*. [Online]. Available: <https://dequeuniversity.com> (Accessed: Sep. 7, 2025) (in English). (дата звернення: 03.05.2025).

12. WebAIM Million 2024 — webaim.org/projects/million/ (дата звернення: 03.05.2025).

13. Стандарт вищої освіти України: перший (бакалаврський) рівень, галузь знань 12 «Інформаційні технології», спеціальність 121 «Інженерія програмного забезпечення». Затв. наказом МОН України № 1166 від 29.10.2018. URL: <https://mon.gov.ua/static-objects/mon/sites/1/vishcha-osvita/zatverdzeni%20standarty/12/21/121-inzhener.programn.zabezp.bakalavr-1.pdf> (дата звернення: 03.05.2026).

14. Освітня програма спеціальності Інженерія програмного забезпечення Львівської політехніки. URL: <https://directory.lpnu.ua/majors/ikni/6.F2.00.00/8/2025/ua/full> (дата звернення: 07.09.2025).

15. Освітня програма спеціальності Інженерія програмного забезпечення Київського національного університету імені Тараса Шевченка.



URL: <https://iss.csc.knu.ua/uk/admission/bachelor.html> (дата звернення: 07.09.2025).

16. Освітня програма спеціальності Інженерія програмного забезпечення КПІ ім. Ігоря Сікорського. URL: https://osvita.kpi.ua/sites/default/files/opfiles/121_oppb_ipzis_2024.pdf (дата звернення: 03.05.2025).

17. Освітня програма спеціальності Інженерія програмного забезпечення Харківського національного університету радіоелектроніки. URL: https://nure.ua/wp-content/uploads/education_programs/2024/2024_bak_121_np_pzpi_2025.pdf (дата звернення: 03.05.2025).