

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»

Олена Трофименко, Анастасія Дика

Тестування та забезпечення якості програмних систем

Навчальний посібник
для підготовки здобувачів вищої освіти
галузі знань 12 «Інформаційні технології»

Одеса
Фенікс
2024

УДК 004.4 (076)
Т 76

*Рекомендовано Вченою радою
Національного університету «Одеська юридична академія»,
протокол № 5 від 6 березня 2024 р.*

Автори:

Трофименко О. Г. – кандидат технічних наук, доцент, доцент кафедри інформаційних технологій Національного університету «Одеська юридична академія»;

Дика А. І. – асистент кафедри інформаційних технологій Національного університету «Одеська юридична академія».

Рецензенти:

Савельєва О. С. – доктор технічних наук, професор, завідувач кафедри комп'ютеризованих систем та програмних технологій інституту комп'ютерних систем Національного університету «Одеська політехніка»;

Ісмаїлова Н. П. – доктор технічних наук, професор кафедри інженерної механіки Військової академії.

Трофименко О. Г., Дика А. І.

Т 76 Тестування та забезпечення якості програмних систем : навч. посіб. [Електронне видання] / О. Г. Трофименко, А. І. Дика ; Нац. ун-т «Одес. юрид. академія». – Одеса : Фенікс, 2024. – 195 с. – **Режим доступу:** <https://hdl.handle.net/11300/27717>

ISBN 978-617-8395-88-9

Розглянуто особливості процесу тестування програмного забезпечення: фази та технології, тестову документацію, критерії та метрики тестів тощо. Вивчаються засоби програмної інженерії щодо оцінки якості у загальному процесі проєктування та конструювання ПЗ, методи верифікації та валідації програмного забезпечення. Містить детальний теоретичний матеріал щодо аналізу вимог та показників якості програмного забезпечення, видів та підготовки і проведення тестування програмного продукту, який вивчають у рамках навчальної дисципліни. Призначено для студентів з метою вивчення та закріплення лекційного матеріалу, підготовки до лабораторних занять та самостійних робіт з дисципліни «Тестування та забезпечення якості програмних систем».

Для підготовки здобувачів вищої освіти галузі знань 12 «Інформаційні технології».

УДК 004.4 (076)

ISBN 978-617-8395-88-9

© О. Г. Трофименко, А. І. Дика, 2024

ЗМІСТ

Передмова	6
Розділ 1. Аналіз вимог до програмного забезпечення	7
1.1 Декомпозиція програм	7
1.1.1 Навіщо потрібна декомпозиція програм.....	7
1.1.2 Що таке MindMap.....	8
1.1.3 MindMap у тестуванні.....	11
1.2 Розробка специфікації вимог до програмного продукту	15
1.2.1 Складові специфікації вимог	15
1.2.2 Як тестуються вимоги?	18
1.2.3 Зразок (шаблон) специфікації вимог	19
Контрольні запитання для самоконтролю	23
Розділ 2. Підготовка до тестування програмного продукту	24
2.1 Термінологія тестування.....	24
2.2 Види і ціна помилок програмного забезпечення	27
2.2.1 Види помилок програмного забезпечення	27
2.2.2 Вартість виправлення помилок.....	30
2.2.3 Класифікація помилок програмного забезпечення	35
2.2.4 Серйозність і пріоритет дефектів	41
2.2.5 Життєвий цикл дефектів.....	47
2.3 В чому різниця між QC і QA	49
2.4 Рівні тестування	51
2.4.1 Компонентне тестування.....	51
2.4.2 Інтеграційне тестування	52
2.4.3 Системне тестування	53
2.4.4 Приймальне тестування.....	53
2.5 Класифікація видів тестування програмного забезпечення.....	54
Контрольні запитання для самоконтролю	58
Розділ 3. Проведення тестування	59
3.1 Життєвий цикл тестування програмного забезпечення (STLC) і фази STLC.....	59
3.1.1 Визначення та фази STLC	59
3.1.2 Різниця між STLC і SDLC	69
3.2 Результати тестування (Test Deliverables)	71
3.3 Добір видів тестування	79
3.4 Розробка тестового сценарію	81
3.5 Розробка плану тестування програмного забезпечення	85
3.6 Розробка тест-кейсів.....	88
3.6.1 Поняття та види тест-кейсів.....	88

3.6.2 Мета написання тест-кейсів	89
3.6.3 Життєвий цикл тест-кейса.....	90
3.6.4 Атрибути (поля) тест-кейса.....	91
3.6.5 Властивості якісних тест-кейсів	94
3.6.6 Інструменти для створення тест-кейсів	96
3.7 Техніки тест-дизайну при розробленні тест-кейсів	98
3.7.1 Техніки тест-дизайну	98
3.7.2 План розробки тест-кейсів	100
3.7.3 Еквівалентний поділ (Equivalence Partitioning – EP)	101
3.7.4 Аналіз граничних значень (Boundary Value Analysis – BVA)	102
3.7.5 Таблиця прийняття рішень.....	104
3.7.6 Pairwise testing (попарне тестування).....	105
3.7.7 Pairwise testing (попарне тестування).....	107
3.7.8 Pairwise testing (попарне тестування).....	107
3.8 Робота з чеклістами.....	107
3.8.1 З чого складаються чеклісти	107
3.8.2 Правила складання чеклістів.....	110
3.8.3 Переваги використання чеклістів	110
3.9 Розробка звітів про помилки	111
3.9.1 Правила складання Bug Report	111
3.9.2 Що включити у звіт про помилку.....	113
3.9.3 Складові звіту про помилку	113
Контрольні запитання для самоконтролю	116
Розділ 4. Види тестування.....	118
4.1 Тестування юзабіліті	118
4.1.1 Специфіка тестування зручності використання.....	118
4.1.2 Приклад сценарію тестування зручності використання	121
4.1.3 План юзабіліті тестування.....	121
4.1.4 Методи юзабіліті тестування: від тестування інтерфейсів до глобальних оцінок	123
4.1.5 Розробка анкети тестування зручності використання.....	123
4.2 Кросбраузерне тестування.....	126
4.2.1 Підходи до кросбраузерного тестування.....	126
4.2.2 Вибір браузерів для тестування	127
4.2.3 Типи кросбраузерного тестування	128
4.2.4 Інструменти тестування кросбраузерності верстки	128
4.3 Тестування API	129
4.3.1 Термінологія взаємодії з вебсерверами	129
4.3.2 Оформлення запитів.....	130
4.3.3 Тестування API.....	132
4.4 Тестування клієнт-серверної архітектури.....	133
4.4.1 Поняття клієнт-серверної архітектури.....	133
4.4.2 Ключові області тестування вебзастосунків	136
4.4.3 Функціональне тестування вебзастосунків	136
4.4.4 Тестування HTTP-протоколу	137

4.4.5 Запити та відповіді: їхня структура, формування і методи	137
4.4.6 Використання JSON для тестування API.....	139
4.5 Тестування продуктивності.....	142
4.5.1 Тестування навантаження	142
4.5.2 Стратегії навантажувального тестування	143
4.5.3 Як виконати тестування навантаження.....	144
4.5.4 Стрес-тестування.....	144
4.5.5 Чекліст тестування продуктивності	145
4.5.6 Засоби навантажувального тестування.....	146
4.6 Тестування трафіка між клієнтом і сервером	147
4.6.1 Аналіз трафіка	147
4.6.2 Операції над запитамі	148
4.6.3 Статистика продуктивності.....	148
4.6.4 Застосування фільтрів.....	148
Контрольні запитання для самоконтролю	149
Розділ 5. Автоматизація тестування.....	151
5.1 Автоматизація тестування вебзастосунків	151
5.1.1 Поняття автоматизованого тестування:	151
5.1.2 Типи автоматизованого тестування	151
5.1.3 Інструменти автоматизованого тестування.....	152
5.2 Selenium WebDriver.....	153
Контрольні запитання для самоконтролю	155
Розділ 6. Показники якості програмного забезпечення	156
6.1 Якість та тестування.....	156
6.1.1 Поняття якості програмного забезпечення.....	156
6.1.2 Характеристики програмного забезпечення	158
6.1.3 Моделі якості програмного забезпечення	160
6.1.4 Атрибути якості програмного забезпечення	164
6.2 Управління якістю програмного забезпечення	168
6.3 Метрики тестування	171
6.3.1 Поняття метрик тестування програмного забезпечення.....	171
6.3.2 Метрики ефективності тестів.....	172
6.3.3 Метрики контролю якості для вимірювання ефективності команди	174
6.3.4 Використання метрик тестування	175
Контрольні запитання для самоконтролю	183
Додаток А. Software Requirements Specification.....	184
Список використаної та рекомендованої літератури.....	189

Передмова

У навчальному посібнику подано теоретичний матеріал для дисципліни «Тестування та забезпечення якості програмних систем». Мета розробки навчального посібника полягає у підвищенні ефективності засвоєння теоретичних знань та організації самостійної роботи студентів по засвоєнню матеріалу навчальної дисципліни. Систематизований та логічно послідовний виклад змісту курсу сприятиме поліпшенню якості підготовки студентів в цілому.

Навчальна дисципліна «Тестування та забезпечення якості програмних систем» знайомить майбутніх розробників програмного забезпечення з особливостями процесу тестування програмного забезпечення (ПЗ): фази та технології, тестова документація, критерії та метрики тестів, надає уявлення про важливість оцінки якості у загальному процесі проєктування та конструювання ПЗ, методи верифікації та валідації програмного забезпечення.

Метою дисципліни «Тестування та забезпечення якості програмних систем» є формування у майбутніх розробників програмного забезпечення базових знань про способи оцінки та забезпечення якості ПЗ, основні види та методи тестування ПЗ, життєвий цикл тестування ПЗ (STLC), набуття навичок роботи з тестовою документацією.

Після успішного вивчення дисципліни здобувач вищої освіти буде здатен:

- здійснювати професійну діяльність на основі знань ключових понять, принципів, методів і засобів тестування програмного забезпечення, як діяльності, спрямованої на забезпечення якості програмних систем;
- володіти питаннями верифікації та валідації програмних продуктів, моделями та принципами контролю якості програмного забезпечення, відповідними поняттями та вимогами державних та міжнародних стандартів у галузі розробки ПЗ;
- вміти ідентифікувати, класифікувати та формулювати вимоги до ПЗ, розробляти та працювати з тестовою документацією (Test Plan, Test Cases, Test Strategy, Bug Report);
- знати провідні фактори та атрибути якості ПЗ, процедури, процеси і ресурси, необхідні для впровадження управління якістю відповідно до стандартів ISO 25010, ISO 9000 та ін., професійні та етичні вимоги до фахівців з програмної інженерії, правила дисципліни зобов'язань як форми відносин і соціальних взаємин, що виникають при колективній розробці ПЗ;
- застосовувати на практиці принципи системного підходу до оцінки якості програмного продукту за показниками надійності, знання метрик контролю та забезпечення якості ПЗ як функції, яка гарантує, що стандарти, процеси та процедури відповідають проєкту та правильно реалізовані.

УВАГА!

Повна версія навчального посібника розташована
на сайті дистанційного навчання факультету
кібербезпеки та інформаційних технологій
Національного університету "Одеська юридична
академія"

за адресою: <http://cyber.onua.edu.ua>

З усіх питань стосовно повної версії навчального
посібника звертатися за адресою:
trofymenko@onua.edu.ua

Список використаної та рекомендованої літератури

1. 10 Free Test Automation Tools: What You Need to Know. URL: <https://www.rainforestqa.com/blog/free-test-automation-tools>
2. 10+ Bug Report Templates – PDF, DOC, XLS. URL: <https://www.template.net/business/report-templates/bug-report-template/> ;
3. 2024 Must-Know Cyber Attack Statistics and Trends. URL: <https://www.embroker.com/blog/cyber-attack-statistics/>
4. API Testing Tutorial: A Complete Guide For Beginners. URL: <https://www.softwaretestinghelp.com/api-testing-tutorial/>
5. Baskirt O. Software Testing Metrics and KPIs. URL: <https://www.swtestacademy.com/software-testing-metrics-kpi/>
6. Best practices for writing test cases. URL: <https://www.thirdrocktechkno.com/blog/best-practices-while-writing-test-cases/>
7. Boeing Starliner Flight’s Flaws Show ‘Fundamental Problem,’ NASA Says. URL: <https://www.nytimes.com/2020/02/07/science/boeing-starliner-nasa.html>
8. Cohn M.. Succeeding with Agile. Software Development Using Scrum. Addison-Wesley Professional; 1st edition. 2009. 512 p. ISBN-13: 978-0321579362.
9. Cost Of Poor Software Quality in the U.S: A 2022 Report. URL: <https://www.it-cisq.org/the-cost-of-poor-quality-software-in-the-us-a-2022-report/>
10. Defect Severity And Priority In Testing With Examples And Difference. URL: <https://www.softwaretestinghelp.com/how-to-set-defect-priority-and-severity-with-defect-triage-process/>
11. Developers spend 20% of their time fixing problems – and it’s killing your company. URL: <https://venturebeat.com/dev/developers-spend-20-of-their-time-fixing-problems-and-its-killing-your-company/>
12. Difference Between Test Scenario Vs Test Case. URL: <https://www.softwaretestingmaterial.com/test-scenario-vs-test-case/>
13. Error, Defect, and Failure. URL: <https://www.toolsqa.com/software-testing/istqb/error-defect-failure/>
14. Finastra brings servers back online after ransomware attack. URL: <https://www.fintechfutures.com/2020/03/finastra-brings-servers-back-online-after-ransomware-attack/>
15. Goebelbecker E. How much could software errors be costing your company? URL: <https://raygun.com/blog/cost-of-software-errors/>

16. Google suffers global outage with Gmail, YouTube and majority of services affected. URL: <https://www.theguardian.com/technology/2020/dec/14/google-suffers-worldwide-outage-with-gmail-youtube-and-other-services-down>
17. How AI Will Transform Project Management. URL: <https://hbr.org/2023/02/how-ai-will-transform-project-management>.
18. How To Create Test Scenarios With Examples: URL <https://www.softwaretestingmaterial.com/test-scenarios/>
19. How To Create Test Strategy Document [Sample Test Strategy Document]. URL: <https://www.softwaretestingmaterial.com/test-strategy/>
20. How to Write a Neat Bug Report? [Complete Guide] URL: <https://www.testbytes.net/blog/bug-report/>
21. How to Write a Test Plan with the IEEE 829 Standard. URL: <https://ieeexplore.ieee.org/document/573169-with-the-ieee-829-standard/>
22. How to Write Test Cases: Test Case Template With Examples. URL: <https://www.softwaretestingmaterial.com/test-case-template-with-explanation/>
23. IEEE 1061-1998 - IEEE Standard for a Software Quality Metrics Methodology. URL: <https://ieeexplore.ieee.org/document/749159>.
24. IEEE 610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology. URL: <https://ieeexplore.ieee.org/document/159342>.
25. IEEE 829-1983 Standard for Software Test Documentation. URL <https://ieeexplore.ieee.org/document/573169>
26. IEEE Guide to Software Engineering Body of Knowledge, SWEBOK, 2004.
27. ISO 9000:2015(en) Quality management systems – Fundamentals and vocabulary. URL: <https://www.iso.org/obp/ui/#iso:std:iso:9000:ed-4:v1:en>
28. ISO/IEC 25000:2014. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). URL: <https://www.iso.org/standard/64764.html>.
29. ISO/IEC 25010:2011 Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models. URL: <https://www.iso.org/standard/35733.html>
30. ISO/IEC 9126-1:2001. Software engineering – Software product quality – Part 1: Quality model.
31. ISTQB Glossary. URL: <https://glossary.istqb.org/>
32. Major Computer Outage Stirs Chaos at London Heathrow Airport. URL: <https://airlinegeeks.com/2020/02/17/major-computer-outage-stirs-chaos-at-london-heathrow-airport/>
33. Manual vs. Automated Testing: A Practical Guide. URL: <https://www.rainforestqa.com/blog/manual-vs-automated-testing>
34. Martin R., Melnik G. Tests and Requirements, Requirements and Tests: A Möbius Strip. 2008. Vol. 25, No. 1. URL: <http://gmelnik.com/papers/Melnik-Martin-Tests-and-Requirements-The-Moebius-Strip-IEEE-Software-2008.pdf>

35. Mind Map у тестуванні - або легкий спосіб тестувати складні програми. URL: <https://habr.com/ru/post/515990/>
36. Paasivaara M., Kruchten Ph. Agile Processes in Software Engineering and Extreme Programming – Workshops. Copenhagen: Springer, 2020. 330 p.
37. Requirements Traceability Matrix in Project Management (Example+Template). URL: <https://pmbasics101.com/requirements-traceability-matrix/>
38. Sample Test Case Template With Test Case Examples. URL: <https://www.softwaretestinghelp.com/test-case-template-examples/>
39. Siddiqui S. Learning Test-Driven Development: A Polyglot Guide to Writing Uncluttered Code. 1st Edition. O'Reilly, 2022. 280 p.
40. Software engineer salary in United States. URL: https://www.indeed.com/career/software-engineer/salaries?from=top_sb
41. Software Test Metrics – Product Metrics & Process Metrics. URL: <https://www.softwaretestingmaterial.com/test-metrics/#ProcessMetrics>
42. Software Test Plan Template with Detailed Explanation [Sample Test Plan Document]. URL: <https://www.softwaretestingmaterial.com/test-plan-template/>
43. Software Testing Life Cycle – Different Stages of Testing. URL: <https://www.edureka.co/blog/software-testing-life-cycle/#testcycleclosure>
44. Test Deliverables in Software Testing – Detailed Explanation. URL: <https://www.softwaretestingmaterial.com/test-deliverables/>
45. Test Scenarios of Gmail: URL <https://www.softwaretestingmaterial.com/test-scenarios-of-gmail/>
46. Test Scenarios Registration Form: URL <https://www.softwaretestingmaterial.com/test-scenarios-registration-form/>
47. Test Strategy vs Test Plan – Difference between Test Strategy and Test Plan. URL: <https://www.softwaretestingmaterial.com/test-strategy-vs-test-plan/>
48. The Cost of Poor Software Quality in the US: A 2020 Report. URL: <https://www.it-cisq.org/cisq-files/pdf/CPSQ-2020-report.pdf>
49. The Snowplow Strategy: Improve Automation Test Coverage in Five Steps. URL: <https://www.rainforestqa.com/blog/automation-test-coverage>
50. The top 5 quality assurance testing metrics. URL: <https://customertimes.com/insights/the-top-5-qa-testing-metrics/>.
51. Top 30 Agile Testing Interview Questions. URL: <https://www.softwaretestingmaterial.com/agile-testing-interview-questions/>
52. Understanding, Defining and Implementing Software QA and Testing Metrics. URL: <https://www.practitest.com/qa-learningcenter/best-practices/software-qa-testing-metrics/>
53. Urma R., Warburton R. Real-World Software Development. O'Reilly, 2020. 202 p.

54. What are some examples of software quality models and how do they differ? URL: <https://www.linkedin.com/adice/0/what-some-examples-software-quality-models-how>
55. What is a Software Bug? Cost of bug fix! URL: <https://www.testbytes.net/blog/what-is-a-software-bug/>
56. What is Requirements Traceability Matrix (RTM) & How To Create It. URL: <https://www.softwaretestingmaterial.com/requirements-traceability-matrix/>
57. What is Software Testing Life Cycle (STLC) & STLC Phases. URL: <https://www.softwaretestingmaterial.com/stlc-software-testing-life-cycle/>
58. Working with JSON. URL: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>
59. You Only Need These 5 QA Metrics to Improve Software Testing. URL: <https://www.rainforestqa.com/blog/quality-assurance-metrics>
60. Your step-by-step mobile application testing process. URL:
61. Аномалія, баг, помилка чи відмова? Базові терміни зі словника тестувальника. URL: <https://training.epam.ua/ua/blog/489>
62. Гусак В., Яблоков С.В. Новітні технології навчання іноземним мовам – метод “mind map” як спосіб підвищення ефективності навчання. *Vox philologi*. Збірник наукових статей. 2019. № 8. Маріуполь: МДУ. С. 21. URL: https://repository.mdu.in.ua/jspui/bitstream/123456789/1271/1/VoxPhilologi_2019_Vyp.8.pdf
63. ДСТУ ISO/IEC 25010:2016 Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (SQuaRE). Моделі якості системи та програмних засобів (ISO/IEC 25010:2011, IDT). URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=69134
64. Дубовик І.Ф. Структуризація навчально-пізнавального процесу з використанням технології майндмеппінг. *International scientific and practical conference*. Wloclawek, Republic of Poland, February 26–27, 2021. С. 169-172. DOI: <https://doi.org/10.30525/978-9934-26-041-4-43>.
65. Єршова О.Л., Перхун Л.П., Ставицький О.В. Лабораторний практикум з дисципліни «Інформатика». Київ: Національна академія статистики, обліку та аудиту, 2021. 163 с.
66. Життєвий цикл дефектів / помилок при тестуванні програмного забезпечення. URL: <https://uk.csstricks.net/8222578-defectbug-life-cycle-in-software-testing>
67. Золотухіна О.А., Негоденко О.В., Резник С.Ю., Разіна С.Я. Якість та тестування інформаційних систем: навч. посібник. Київ: ННІТ ДУТ, 2020. 128 с.
68. Лаврішчева К.М. Програмна інженерія. К. 2008. 319 с.
69. Міні-курс по роботі в TestRail. URL: <https://www.gurock.com/testrail/videos/suites-test-cases/>

70. ОПІ та ТПЗ : лабораторний практикум для студентів напряму підготовки “Комп’ютерні науки” / Укл.: Петровський С.С. Хмельницький: ХНУ, 2020. 76 с.
71. Основи програмної інженерії: навч.-метод. посібник / О.Г. Трофименко, С.Ю. Манаков, Д.Г. Ларін. Одеса: Фенікс, 2022. 197 с. ISBN 978-966-928-808-0
72. Основи тестування програмного забезпечення. URL: <https://training.qatestlab.com/>
73. Особливості тестування додатків на мобільних пристроях. URL: <https://training.qatestlab.com/blog/technical-articles/testing-mobile-devices/>
74. Принципи тестування. URL: <https://qalight.ua/ru/baza-znaniy/printsipyi-testirovaniya/>
75. Рівні тестування. URL: <https://qalearning.com.ua/theory/lectures/material/testing-levels/>
76. Специфікація вимог. URL: <https://qalight.ua/baza-znaniy/cpetsifikatsiya-vimog/>
77. Тестування баз даних: на що звернути увагу? URL: <https://training.qatestlab.com/blog/technical-articles/database-testing-what-to-look-for/>
78. Тестування програмного забезпечення : методичні рекомендації / укл. Ю. І. Скорін. Харків : ХНЕУ ім. С. Кузнеця, 2022. 47 с.
79. Тестування та забезпечення якості програмних систем : навч.-метод. посіб. [Електронне видання] / О.Г. Трофименко, А.І. Дика; Нац. ун-т «Одес. юрид. академія». Одеса: Фенікс, 2024. 140 с. URL: <https://hdl.handle.net/11300/27246>. ISBN 978-617-8395-32-2
80. Топ 150 запитань та відповідей на тестування програмного забезпечення. URL: <https://uk.csstricks.net/8222704-top-150-software-testing-interview-questions-and-answers>
81. Трофименко О.Г., Дика А.І., Лобода Ю.Г. Аналіз інструментів тестування вебзастосунків. *Кібербезпека: освіта, наука, техніка*. 2023. № 4(20). С. 62-71. DOI: 10.28925/2663-4023.2023.20.6271. URL: <https://csecurity.kubg.edu.ua/index.php/journal/article/view/467>
82. Трофименко О.Г., Пастернак Ю.Ю., Манаков С.Ю., Лобода Ю.Г. Автоматизація тестування вебсайтів електронної комерції. *Сучасна спеціальна техніка*. 2021. № 2(65). С. 46-59. DOI: [https://doi.org/10.36486/mst2411-3816.2021.2\(65\).5](https://doi.org/10.36486/mst2411-3816.2021.2(65).5).
83. Філіппов О. Тестування вимог. URL: <https://quality-lab.ru/blog/testing-requirements/>
84. Цілі тестування. URL: <https://qalight.ua/baza-znaniy/tsili-testuvannya/>
85. Шість помилок тестування клієнтської частини та як їх виправити. URL: <https://codeguida.com/post/2999>
86. Що таке тест-план, для чого він потрібен і з чого складається URL: <https://training.qatestlab.com/blog/technical-articles/test-plan/>

87. Що таке тестування програмного забезпечення? URL: <https://qalight.ua/baza-znaniy/shho-take-testuvannya-programnogo-zabezpechennya/>
88. Що таке тестування системної інтеграції (SIT): дізнайтеся на прикладах. URL: <https://uk.myservername.com/12-best-line-graph-maker-tools>
89. Які випробування вам потрібні? Частина 2. Матриця видів тестування. URL: <https://habr.com/ru/post/257159/>
90. Якість ПЗ за ISO 25010:2011. URL: <https://qalight.ua/baza-znaniy/yakist-pz-za-iso-250102011/>
91. Якість програмного забезпечення та тестування: базовий курс: навч. посібник / За ред. Крепич С.Я., Співак І.Я. Тернопіль: ФОП Паляниця В.А., 2020. 478 с.
92. Якість програмного забезпечення та тестування: методичні рекомендації / укл. Ю. І. Скорін. Харків : ХНЕУ ім. С. Кузнеця, 2019. 48 с.

Навчальне видання

*Олена Григорівна Трофименко,
Анастасія Іванівна Дика*

Тестування та забезпечення якості програмних систем

Навчальний посібник
для підготовки здобувачів вищої освіти
галузі знань 12 «Інформаційні технології»

Електронне видання

В авторській редакції

Ум-друк. арк. 11,7.
Заказ № 2403-14.

Видано в ПП «Фенікс»
(Свідоцтво суб'єкта видавничої справи ДК № 1044 від 17.09.02).
Україна, м. Одеса, 65009, вул. Зоопаркова, 25.
e-mail: fenix-izd@ukr.net