

Висновок. Платформа 10G є поєднанням таких технологій, як DOCSIS 4.0 а також мережі розподіленого доступу, які нададуть інтернету швидкість у 10 разів вищу, ніж сьогоденні мережі. 10G не лише забезпечує більш високу симетричну швидкість, але також меншу затримку, підвищену надійність та кращу безпеку у масштабованому форматі за допомогою впровадження технології Remote PHY (R-PHY). Технологія Remote PHY (R-PHY) вирішує проблеми, пов'язані з якістю сигналу, масштабованістю, ефективністю використання спектра і доставкою інтерактивних послуг у кабельних телевізійних і широкосмугових мережах, в результаті чого покращується загальна продуктивність і надійність кабельних мереж.

Список використаних літературних джерел

- [1] Бедард, Ф. (2023). Посібник із проектування конвергентної мережі Cisco Remote PHY. <https://xrdocs.io/design/blogs/cin-design-guide>
- [2] Віларуель, Ф. (2019). Що таке конвергентна міжсистемна мережа (CIN). <https://www.ciena.com/insights/articles/What-is-a-Converged-Interconnect-Network-CIN.html>
- [3] Лабораторія Кабельного Телебачення (2020). Специфікації інтерфейсу служби передачі даних через кабель. <https://community.cablelabs.com/wiki/plugins/servlet/cablelabs/alfresco/download?id=4658dab3-e43c-41b5-be9a-351d1656fa7c>

Павло Чуківський, Ольга Мокрицька*

магістрант кафедри КН НЛТУ України, Львів, Україна,
Chukivskyu.P@nltu.lviv.ua.

*доцент кафедри САП НУ “Львівська політехніка”,
olha.v.mokrytska@lpnu.ua.

Використання автоматизованого тестування програмного забезпечення як методу оптимізації витрат під час розробки програмного продукту

Анотація – Програмне забезпечення набуло ключового значення у житті кожної людини. Незалежно від сфери застосування, різноманітне програмне забезпечення так чи інакше використовується абсолютною більшістю людей на

планеті. Телебачення чи зв'язок, обробка деревини чи виготовлення макетів космічних кораблів – усі ці сфери залежать від програмного забезпечення та його надійності. Щороку інвестиції у ІТ та у розробку програмних продуктів лише зростають, а разом з ними зростають і витрати на забезпечення якості. Використання автоматизованого тестування є одним із способів оптимізації витрат на розробку та тестування, а також має прямий вплив на покращення якості розроблених продуктів.

Ключові слова - TypeScript, ROI, Automation, Manual, автоматизація, тестування.

Розробка програмного забезпечення - це багаторівневий процес, який вимагає різноманітних інвестицій, які не обмежуються лише грошовими коштами. Згідно із прогнозами[1], світові витрати на ІТ сектор у 2023-2024 роках становитимуть 4,7 – 5,1 трильйона доларів, з яких більша частина припадатиме власне на розробку програмних продуктів.

Розробка навіть найпростішого мобільного чи веб-додатку вимагає ретельного планування, вибору необхідних технологій та набору команди, знання та технічні навички якої будуть достатні для успішної реалізації поставленої мети. Витрати на розробку базової аплікації можуть коливатися в діапазоні від \$40000 до \$300,000+[2], та будуть залежати від складності, набору функцій, а також кількості платформ що підтримуються. Складність програмного продукту також прямо відіб'ється і на орієнтовному часі, що потрібний на його розробку. Варто також пам'ятати, що підтримка додаткових функцій чи платформ буде прямо відбиватися на необхідному для розробки часу.

Проте, грошових вливань чи інвестицій у ресурси зазвичай не достатньо для випуску якісного продукту, який буде відповідати заявленим вимогам та відповідати очікуванням кінцевих користувачів. Для забезпечення контролю за дотриманням вимог, а також перевірки програми на відсутність дефектів, у життєвому циклі розробки програмного забезпечення використовується процес тестування.

Тестування програмного забезпечення – це процес, при якому відбувається перевірка готового продукту, чи продукту у стані розробки, на відповідність вимогам. Тестування може відбуватися як вручну (manual testing), яке звичай не вимагає знань спеціалізованих інструментів тестування, і вимагає більше зусиль і людей, так і автоматизовано (automation testing) як за допомогою готових спеціалізованих інструментів (testing frameworks) так і написаних власноруч плагінів чи програм, що використовують популярні мови програмування, такі які Java, JavaScript, TypeScript та ін. Автоматизоване тестування підвищує точність тестування та дає змогу оптимізувати зусилля та час тестувальника порівняно з манувальним тестуванням. В той час, як ручне тестування є більш вартісним та трудомістким процесом, але й схильним до помилок.

Оскільки тестування програмного забезпечення це тривалий та, фактично, безперервний процес на усьому життєвому циклі програмного забезпечення, компаніям необхідно виділяти на нього певний бюджет, який в окремих випадках може наближатися до бюджету, що був витрачений на пряму розробку.

Інвестиції у тестування найбільше окуповують себе на ранніх етапах розробки, оскільки ціна виправлення дефекту, що був знайдений при розробці, буде в рази нижчою ніж виправлення цього дефекту у готовому продукті [3]. Що є зрозумілим, оскільки окрім прямих витрат на виправлення проблеми, компанія може понести репутаційні витрати, а що гірше, ціною помилки може бути чиєсь життя чи здоров'я.

Саме тому, компанії все більше звертають увагу на використання автоматизованого тестування у процесі перевірки продукту на якість. Використання тестових фреймворків дає змогу виконувати більшу кількість тестів залучаючи для цього меншу кількість людино-ресурсів, таким чином дозволяючи збільшувати покриття та отримувати більше рентабельності від вкладених у тестування ресурсів (ROI).

Спробуємо виділити основні переваги автоматизованого тестування у порівнянні із звичайним, ручним, тестуванням:

Скрипт чітко виконує написані кроки, та дозволяє не пропустити дефекти там, де ручний тестувальник може закрити на них очі;

Автоматизований скрипт дозволяє суттєво зекономити час на виконання тестів що регулярно повторюються;

Автоматизовані тести дозволяють зменшити час необхідний для передре-лізного тестування (regression testing, та ін);

Автотести дозволяють зекономити кошти, покриваючи більшу кількість коду, у порівнянні з ручним тестуванням.

Для оптимізації витрат на контроль якості використовуючи автоматизоване тестування, необхідно, перш за все, зрозуміти які витрати (інвестиції) необхідні для запуску цього тестування. Загалом, рентабельність від автоматизації – це цифра, яка показує, наскільки вигідно було запускати автоматизації у порівнянні зі мануальним тестуванням.

Для базової оцінки рентабельності інвестицій (ROI) у автоматизацію, запропонуємо наступну формулу:

$$ROI = \frac{\text{Витрати на ручне тестування} - \text{Витрати на автоматизоване тестування}}{\text{Витрати на автоматизоване тестування}}$$

Розглянемо складові формули:

Витрати на ручне тестування – це сукупні витрати часу на виконання та написання тестів, їхню підтримку та аналіз результатів, помножені на погодинну оплату працю інженера

Витрати на автоматизоване тестування – це сукупні витрати на написання тестів, їхню автоматизацію, час виконання тестів та аналіз результатів помножені на погодинну оплату праці інженерів.

Зробимо умовний розрахунок рентабельності інвестицій на проєкті, з наступними ввідними даними:

	Ручне	автоматизоване
Кількість тестів	300	
Час виконання	10хв	1хв
Час написання	30хв	15хв
Аналіз результатів	120хв	60хв
Погодинна оплата праці [4]	36.3	38.4
Кількість інженерів	1	1

Використовуючи дані наведені вище, розрахуємо рентабельність інвестицій у автоматизоване тестування:

$$\begin{aligned}
 ROI &= \frac{\left(\frac{300 * 30}{60} * 36,3 + \frac{300 * 10 + 120}{60} * 36,3 * 20\right) - \left(\frac{300 * 15}{60} * 38,4 + \frac{300 * 1 + 60}{60} * 38,4 * 20\right)}{300 * \frac{30}{60} * 36,3 + \frac{300 * (1 + 15) + 120}{60} * 38,4} \\
 &= \frac{(5445 + 37752) - (2880 + 4608)}{2880 + 4608} = \frac{35709}{7488} = 4,76 * 100\% = 476\%
 \end{aligned}$$

Як видно з розрахунків, рентабельність інвестицій у автоматизацію складає 476%, у середньостроковій перспективі. Важливо розуміти, що ці розрахунки є поверхневими, та не включають у себе такі аспекти як витрати на ліцензії, додаткове обладнання, тощо.

Висновки: Автоматизоване тестування є вагомим інструментом у оптимізації витрат на тестування програмного забезпечення, а у середньо- і довгостроковій перспективі, компанії здатні суттєво оптимізувати свої витрати на контроль якості за рахунок скорочення часу необхідного на проходження кожного циклу тестування.

Список використаних літературних джерел

- [1] Statista: Information technology (IT) spending forecast worldwide from 2012 to 2024, by segment: <https://www.statista.com/statistics/268938/global-it-spending-by-segment/>
- [2] The Cost of Custom Software Development: All Price-Changing Aspects in 2023: <https://www.cleveroad.com/blog/software-development-cost/>
- [3] What is the cost of defects in software testing?: <https://tryqa.com/what-is-the-cost-of-defects-in-software-testing/>
- [4] PayScale: Salary for Industry: Information Technology (IT) Services: [https://www.payscale.com/research/US/Industry=Information_Technology_\(IT\)_Services/Salary](https://www.payscale.com/research/US/Industry=Information_Technology_(IT)_Services/Salary)