

# Регресійне тестування

**№ уроку:** 12 **Курс:** Manual QA

**Засоби навчання:** Браузер, Microsoft Office

## Огляд, мета та призначення уроку

Метою даного уроку є ознайомлення з цілями, обсягами, видами регресійних тестів, розгляд різних підходів у проведенні регресії, а також визначення часових рамок та послідовності виконання тестів, що входять до об'єму регресії.

## Вивчивши матеріал даного заняття, учень зможе:

- розуміти, що таке регресійне тестування;
- визначати необхідний набір тестів для регресії;
- визначати час початку та приблизну оцінку часу для регресійного тестування.

## Зміст уроку

1. Що таке регресія?
2. Мета регресійного тестування
3. Підходи у регресійному тестуванні
4. Коли проводити регресійне тестування?
5. Автоматизувати чи не автоматизувати

## Резюме

### Що таке регресійне тестування?

Регресійне тестування зазвичай є типом тестування програмного забезпечення, який використовується для підтвердження того, що останні зміни, пов'язані з програмою або кодом, не вплинули несприятливо на існуючі функції. Під час регресійного тестування визначається, що програмне забезпечення або програма працюють нормально щодо нових змін та виправлень помилок. Регресійне тестування в основному полягає у повторному виконанні вже виконаних тестових випадків, щоб переконатися, що програма працює належним чином.

### Навіщо нам потрібно проводити регресійне тестування?

Є кілька причин, через які слід проводити регресійне тестування. Основна мета регресійного тестування - переконатися, що програма працює нормально, коли в код вносяться зміни. Ми повинні перевірити, чи змінений код впливає на інші частини програми чи ні. Якщо ні, чудово! Але якщо це так, мають бути внесені відповідні зміни. Це призводить до підвищення якості продукту та підтвердження того, що виправлені проблеми більше не повторяться.

Деякі переваги виконання регресійного тестування включають:

- Перевіряє, що будь-які зміни коду не вплинуть негативно на інші функції.
- Переконайтеся, що повторювані проблеми не повторюються.
- Під час тестування він є методом управління ризиками.
- Це просто дізнатися, зрозуміти та оцінити.

Якими є побічні ефекти регресійного тестування?

Там, де є кілька переваг проведення регресійного тестування, є деякі недоліки.

- Регресійне тестування необхідно проводити навіть для незначних змін у додатку, щоб проаналізувати вплив зміни. Іноді може бути досить неприємно.
- Якщо в проєкті не використовується автоматизація, виконання одного тесту знову і знову може стати дуже стомлюючим.

Проблеми під час виконання регресійного тестування

Регресійне тестування допомагає виявляти помилки при додаванні нових функцій або оновлень у існуючу кодову базу, а також усувати збої програм та вузькі місця у продуктивності. Однак, при виконанні регресійного тесту тестер стикається з низкою проблем.

Деякі з основних проблем, з якими стикаються тестувальники, включають:

- **Попередня вартість та час.** Коли вводяться нові функції, набір регресійних тестів необхідно постійно оптимізувати. В результаті кількість тестових випадків варіюється і існуючі тести необхідно повторно запускати разом з новими тестами, виконання яких займає багато часу. Паралельне тестування може бути корисним, оскільки воно дозволяє виконувати тестові випадки паралельно у різних браузерах та операційних системах, заощаджуючи кілька тижнів часу виконання замовлення.
- **Складність.** Кількість тестових випадків та їх складність збільшуються в міру ускладнення проєкту чи додатку. В результаті на впровадження та підтримку йде багато часу.
- **Технічне обслуговування.** У міру розширення програми кількість тестів у наборах регресійних тестів збільшується. В результаті підтримання складності та часу виконання має вирішальне значення.
- **Великий обсяг.** Якість регресійного тестування визначається типом створеного вами набору тестів. Пакет регресії необхідно оновлювати щоразу під час збирання або випуску. Наразі тестувальникам буде складно запам'ятати кожен змін, а без цього регресійне тестування неможливе. Це єдиний спосіб охопити усі ці елементи. В результаті величезна широта є серйозною проблемою.
- **Підхід до тестування.** Вибір найкращого підходу до регресійного тестування має вирішальне значення для успіху. У вас може бути доступ до чудових ресурсів та інструментів, але якщо у вас немає чіткої стратегії, коли і як проводити регресійне тестування, вам буде важко.

### Типи регресійного тестування

Якщо ми хочемо провести регресійне тестування, ми спочатку повинні знайти помилки в коді. При виявленні помилок вживаються відповідні заходи щодо їх усунення. Після цього з набору тестів вибираються відповідні тестові випадки, які охоплюють як змінені, так і торкнулися частини коду. Сім різних типів регресійного тестування включають:

- **Повторно протестувати все** — Один із способів проведення регресійного тестування — повторне виконання всього набору тестів. Насправді це дорогий процес, оскільки він потребує більше часу, ресурсів та зусиль для повторного виконання всього набору тестів. Про помилки, виявлені у цьому процесі, повідомляється для виправлення.
- **Коригуючий регресійний тест** – Коригуючий регресійний тест – одна з найпростіших форм регресійного тестування, що вимагає мінімальних зусиль. Коригуюче регресійне тестування не вимагає внесення змін до існуючої кодової бази та додавання нових функцій до програми. Вам просто потрібно протестувати існуючу функціональність та пов'язані з нею тестові приклади, а не створювати нові.
- **Модульне регресійне тестування** – Модульне регресійне тестування є невід'ємною частиною регресійного тестування, при якому код тестується ізольовано. Всі інші взаємодії, інтеграція та залежності відключаються під час виконання модульного регресійного тестування, і акцент

робиться на одиночний модульний код. Як правило, це тестування проводиться в годинник з низьким трафіком і непіковий годинник.

- **Вибіркове регресійне тестування** – Вибіркове регресійне тестування аналізує вплив існуючого коду, а також як новий, так і існуючий код. Загальні елементи, такі як змінні та функції, включені до програми для швидкого визначення результатів, не впливаючи на процес.

- **Прогресивне регресійне тестування** – Тестові випадки створюються з урахуванням вимог прогресивного регресійного тесту. Коли продукт вносяться лише незначні поліпшення, нові тестові приклади розробляються без зміни існуючого коду продукту.

- **Повне регресійне тестування** – Деякі незначні або суттєві зміни можуть вплинути на продукт. Повне регресійне тестування використовується в тому випадку, коли до поточного коду вносяться значні зміни. Це допомагає у ремонті будь-яких модифікацій, зроблених у процесі тестування.

- **Часткове регресійне тестування** – Коли новий код додається до існуючої кодової бази, проводиться часткове регресійне тестування. Це допомагає виявляти критичні помилки в існуючому коді та дозволяє їх тестувати, не торкаючись системи.

## Закріплення матеріалу

- Які основні задачі вирішує регресійне тестування?
- Які методи використовують для визначення тестових випадків у регресійному тестуванні?
- Які типи помилок може бути виявлено в результаті регресійного тестування?
- Які критерії вибираються з метою оцінки якості регресійного тестування?
- Які інструменти використовують для автоматизації регресійного тестування?
- Які ризики пов'язані із недостатнім регресійним тестуванням?
- Які переваги надає використання безперервної інтеграції для регресійного тестування?
- Які підходи можна використовувати для оптимізації регресійного тестування та прискорення його процесу?

## Самостійна діяльність учня

### Завдання 1

На прикладі будь-якого інтернет-магазину (Rozetka, Allo, Ali Express etc) напишіть 5 напівпріоритетніших end2end сценаріїв для регресійного тестування. Зазначте, чи рекомендували б ви їх для автоматизації

### Завдання 2

Зареєструйтесь на сайті <https://trello.com/>

Створіть нову дошку та вивчіть як вони працюють.

Розробіть тести для регресійного тестування компоненту «Дошка».

У вас немає ліміту на кількість, лише ви вирішуєте що є достатньою кількістю.

Зазначте всі необхідні поля у тестах. Виконайте регресійне тестування застосунку відповідно до плану та опишіть результати тестування.

### Завдання 3

Регресійний сет вашої програми складається з 2000 тестів. Та зараз у вас є час лише на 200. Напишіть ваші дії з формування такого регресійного сету кейсів.

## Рекомендовані ресурси

1. Software Testing Help (<https://www.softwaretestinghelp.com/>) – блог з інформацією про регресійне тестування та інші аспекти тестування ПЗ.
2. Ministry of Testing (<https://www.ministryoftesting.com/>) – спільнота тестувальників з інформацією про регресійне тестування, автоматизацію тестування та інші аспекти тестування ПЗ.
3. Testing Excellence (<https://www.testingexcellence.com/>) – ресурс з інформацією про регресійне тестування, методології тестування та інструменти автоматизації тестування.
4. Software Testing Material (<https://www.softwaretestingmaterial.com/>) – блог з інформацією про регресійне тестування, інструменти тестування та інші аспекти тестування ПЗ.
5. Testim (<https://www.testim.io/blog/>) – блог компанії, яка займається автоматизацією тестування, з інформацією про регресійне тестування, інструменти тестування та автоматизації.
6. TestingWhiz (<https://www.testing-whiz.com/blog>) – блог з інформацією про регресійне тестування, автоматизацію тестування та інструменти тестування.
7. DZone (<https://dzone.com/>) – ресурс для розробників та тестувальників з інформацією про регресійне тестування, автоматизацію тестування та інші аспекти тестування ПЗ.
8. Guru99 (<https://www.guru99.com/>) – ресурс з інформацією про регресійне тестування, методології тестування та інструменти автоматизації тестування.