|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Характеристика | Функціональне | Нефункціональне | Smoke | Регресійне |
| Що перевіряється | Перевіряє, чи відповідає програмне забезпечення функціональним вимогам, які були визначені в специфікації або вимогах до програми. Це вид тестування, який зосереджений на перевірці того, як програма працює і як взаємодіє з користувачем та іншими компонентами системи. | Перевіряє характеристики та якості програмного забезпечення ("як добре" програма виконує свої функції, а не "що" вона робить) | Перевірка основних функціональних та нефункціональних характеристик програми, щоб впевнитися, що базовий функціонал не пошкоджений, і нові зміни не спричинили критичних проблем | Основна мета регресійного тестування забезпечити стабільність і надійність програмного продукту після внесення змін, а також переконатися, що виправлені помилки не повертаються |
| Коли застосовується | * Під час розробки * Після завершення розробки * Перед випуском версії * Під час оновлення або змін * При зміні зовнішнього середовища | * Перед випуском версії * Після завершення розробки * При зміні робочого середовища * При аналізі ризиків * При оцінці якості | На ранніх етапах тестування, зазвичай відразу після успішної збірки або зміни програмного продукту | * Після внесення змін * Після нової збірки |
| Обмеження | * Доступ до вимог(потребує доступу до чітких і повних вимог або специфікацій програмного забезпечення) * Час(може бути часово та ресурсозатратним, оскільки потребує планування тестових сценаріїв, виконання тестів і аналізу результатів) * Навколишнє середовище(можуть знадобитися спеціальні тестові середовища або реальні умови використання програми) * Область покриття(може бути обмежене в покритті всіх можливих сценаріїв та комбінацій, особливо великих програм або систем зі складною логікою) | * Суб'єктивність(оцінка деяких характеристик програми, може бути суб'єктивною і залежати від індивідуальних думок тестувальників) * Обсяг тестів(забезпечення повного покриття всіх можливих варіантів та сценаріїв тестування нефункціональних характеристик може бути дуже складним та ресурсоємким завданням) * Автоматизація(автоматизація нефункціонального тестування може бути вимогливою з точки зору розробки скриптів та підтримки тестових засобів) * Часові рамки(може займати багато часу, особливо при проведенні тестів продуктивності або навантаження, що може бути обмеженням у випадку скорочених термінів проекту) | * Поверхневість перевірки(фокусується на основних функціональностях і характеристиках програми, тому воно не замінює більш детального і глибокого тестування) * Відсутність детальних тест-кейсів(зазвичай не розробляють детальних тест-кейсів, а перевірка виконується на базі загальних сценаріїв) * Вимога до стабільності(продукт повинен бути в стабільному стані, щоб перевірка базової функціональності не завершилася збоями або некоректною роботою) | * Недостатність часу(В надзвичайних ситуаціях або проектах з обмеженими ресурсами може бути недостатньо часу для виконання всіх регресійних тестів) * Залежність від автоматизації(Якщо регресійне тестування базується на автоматизованих тестах, будь-які зміни в коді програмного продукту можуть вплинути на роботу автоматизованих скриптів, що може призвести до потреби вносити зміни в тести) * Вибір тестової бази(Важливо правильно вибрати тестову базу для регресійного тестування, щоб впевнитися, що покриваються всі критичні аспекти програмного продукту) * Перевірка нового функціоналу(не може замінити випробування нового функціоналу або нових фіч, оскільки його основна мета перевірити вплив змін на вже існуючий функціонал) |
| Особливості | * Зорієнтованість на вимоги(фокусується на перевірці відповідності програми визначеним функціональним вимогам) * Використання тестових даних(вимагає ретельного планування тестових даних, які використовуються для перевірки різних сценаріїв) * Тестування реальних умов(допомагає перевірити програмне забезпечення в різних умовах, таких як різні операційні системи, браузери та інше) | * Оцінка UX(забезпечення зручності та задоволення від використання програми) * Незалежність від функцій(перевіряються характеристики програми, які не пов'язані з його безпосередньою функціональністю, але впливають на його продуктивність, надійність і якість) * Важливість в реальних умовах(допомагає визначити, як програма поводиться в реальних умовах експлуатації) * Вимоги до продуктивності(є важливим аспектом нефункціонального тестування, оскільки вони допомагають забезпечити швидкість та ефективність програми в умовах реального використання) | * Не замінює інших видів тестування(слід розглядати як допоміжний інструмент, який доповнює інші види тестування, такі як функціональне тестування, регресійне тестування та інші) * Мінімальний обсяг ресурсів(не вимагає великого обсягу ресурсів, оскільки воно є швидким і поверхневим) * Спрощений звіт(Результати smoke test можуть бути представлені у спрощеному звіті, який містить лише основні висновки про стан програми) | * Автоматизація регресійного тестування(Використання автоматизованих тестів допомагає забезпечити швидке та ефективне проведення регресійного тестування після змін у коді) * Врахування залежностей(необхідно враховувати залежності між різними частинами програмного продукту та їх вплив на інші компоненти) * Швидка реакція на зміни(допомагає вчасно виявляти проблеми, які можуть виникати після внесення змін, тим самим забезпечуючи швидку реакцію на проблеми та їх усунення) * Оптимізація часу тестування(допомагає зекономити час, оскільки воно зосереджене на тестуванні тільки тих частин продукту, які були змінені) |

Регресія та ретестинг - це два різних підходи, використовувані в тестуванні програмного забезпечення для забезпечення якості продукту. Регресія виконується після внесення змін у програму і має перевірити, чи не з'явилися нові помилки або проблеми, забезпечуючи стабільність системи. Ретестинг проводиться після виявлення дефекту та його виправлення, і має перевірити ефективність виправлення та впевнитися, що програма працює правильно після цього. Регресійні тести допомагають запобігти появу нових проблем, тоді як ретестинг підтверджує коректність виправлень.

**2lvl**

Проведення тільки функціонального тестування без перевірки нефункціональних вимог залежить від типу продукту, його призначення. Але бажано проведення і функціонального, і нефункціонального тестування для забезпечення якості продукту та задоволення потреб користувачів.

Ось причини, чому нефункціональне тестування важливе:

UX(Погана продуктивність може зробити продукт малопридатним для використання, навіть якщо він виконує всі необхідні функції)

Безпека(Перевірка на наявність потенційних уразливостей та інших проблем безпеки, що можуть стати причиною небезпеки для користувачів або організації)

Масштабованість(При збільшенні обсягів даних або користувачів продукт може зазнати навантаження. Тестування на масштабованість допомагає виявити можливі проблеми з продуктивністю при зростанні обсягів)

Надійність(допомагає виявити потенційні проблеми з надійністю продукту, які можуть призвести до збоїв або непередбачуваних ситуацій)

Проведення тільки функціонального тестування можливе у випадках дуже простих продуктів, але у більшості випадків потрібно виконувати як функціональне, так і нефункціональне тестування, щоб забезпечити якість і надійність продукту.

Smoke тестування є необхідним, оскільки воно допомагає швидко виявити серйозні дефекти або помилки, які можуть заважати продовженню подальшого тестування.

Випадки коли воно є доречним:

* Перед початковим тестуванням
* Перед циклом регресійного тестування
* Після змін або оновлень