4. Data-driven testing - методологія автоматизації тестування, заснована на використанні в скриптах параметрів виконання тестів. Параметри, що задають логіку роботи тестів (наприклад, вхідні значення та очікувані результати), знаходяться в деякому зовнішньому сховищі. Подібний підхід дозволяє організувати виконання сценаріїв з різними наборами вхідних параметрів і підвищити гнучкість тестування.

При використанні keyword-based testing створюється спеціалізований словник ключових слів, що описують системні події (наприклад, «Logon User»). З кожним ключовим словом пов'язані необхідні параметри (наприклад: «UserID», «Password») та очікувані результати. Для кожного ключового слова має бути задане опис. Даний підхід дозволяє писати функціональні тести практично на природній мові, не вимагаючи від тестувальника навичок програмування.

Автоматизовані тести, як правило, є регресійний (від лат. Regression - рух назад), тобто спрямованими на виявлення помилок у вже протестованих ділянках вихідного коду при внесенні змін.

Однією з головних проблем автоматизованого тестування є його трудомісткість: незважаючи на те, що воно дозволяє усунути частина рутинних операцій і прискорити виконання тестів, великі ресурси можуть витрачатися на оновлення самих тестів. Проте подібні інвестиції в більшості випадків виправдані, так як ручне тестування вимагає набагато більше ресурсів.

5. Ручне тестування - це процес пошуку дефектів у роботі програми, коли тестувальник перевіряє працездатність всіх компонентів програми, як якщо б він був користувачем. Часто, для точності перевірки, тестувальник використовує заздалегідь заготовлений план тестування, в якому визначені найбільш важливі аспекти роботи програми.

Ручне тестування - це ключовий етап розробки програмного забезпечення. Тестер може не дотримуватися строго планом тестування, а відхилятися від нього для більш повного тестування, наближеного до використання програми звичайним користувачем.

Великі проекти дотримуються суворої методології тестування з метою виявлення максимальної кількості дефектів. Системний підхід до тестування включає в себе кілька етапів:

- Вибір методології тестування, придбання необхідного обладнання (комп'ютери, програмне забезпечення), прийняття людей на посаду тестерів;

- Складання тестів з описом виконання і очікуваним результатом;

- Передача наборів тестів тестерам, які вручну виконують тести і записують результати;

- Передача результатів тестів розробникам в докладному доповіді з описом всіх виявлених проблем для обговорення і виправлення дефектів.

Для тестування можуть бути використані статичний і динамічний підходи. Динамічні підхід включає в себе запуск програмного забезпечення. Статистичне тестування включає в себе перевірку синтаксис та інші особливості коду програми.

Тестування може бути функціональним і не функціональним. Функціональне тестування - це перевірка робочої області програмного забезпечення. Чи нефункціональне тестування - перевірка продуктивності, сумісності та безпеки тестової системи.

Ручне тестування може застосовуватися лише до програм, які мають обмежену кількість варіантів використання. При розробці складних програмних систем можливості ручного тестування сильно обмежені, тому що при внесенні змін в код потрібно організувати повторне виконання тестів. Проте при ручному тестуванні можна виявити надзвичайно витончені помилки, що вкрай складно зробити з використанням автоматизованого тестування.