**Performance Testing (Тестування продуктивності)**— це тип нефункціонального тестування, в якому продуктивність системи оцінюється при очікуваному або більш високому навантаженні в умовах різних сценаріїв поведінки. Ціль Performance Testing перевірити швидкодію реакції системи на певний рівень навантаження, працездатність, стабільність. Параметри оцінки продуктивності: час відгуку, надійність, використання ресурсу, масштабованість тощо.

Тестування продуктивності дозволяє виявити можливі уразливості та недоліки в програмі з метою запобігання їх впливу на роботу системи під час використання.

**Load Testing (Тестування навантаження)** — це тип Performance Testing й техніка автоматизованого тестування метою якого є перевірити, що наша програма працює у відповідності до вимог за нормальних умов, тобто при звичайному очікуваному користувацькому навантаженні.

Під час перевірки навантаження ми оцінюємо час відгуку на запити різних типів, пропускну спроможність, використання ресурсів, частоту помилок та інші параметри програми по бажанню. По результатах формуємо Метрики у якості звіту для зацікавлених сторін.

**Тестування навантаження (Load Testing) виконується тільки для нормальних умов!!!**

**Stress Testing** — це також один із типів Performance Testing. Стрес-тестування передбачає тестування продуктивності, шляхом збільшення робочого навантаження на програму більше ніж очікується — створення штучного контрольованого стресу для неї. Стрес-тестування проводиться для виявлення витоків пам’яті та перевірки надійності програми.

**Об’ємне тестування (VolumeTesting)** — це тип тестування продуктивності, який проводиться зі збільшенням не навантаження і часу роботи, а збільшенням кількості оброблюваних даних, які зберігаються і використовуються в програмі. За допомогою VolumeTesting перевіряється масштабованість програми. Ось, таким тестуванням можна ідентифікувати вузьке місце куди не «влізає» задокументований обсяг даних.

**Reliability Testing** — це тип тестування програмного забезпечення на витривалість, який досліджує працездатність додатку при тривалій багатогодинній роботі, при середньому для програми навантаженні. Ціль виявити витоки ресурсів. Тобто у процесі тестування ретельно моніторяться ресурси системи (пам’ять, процесор, завантаження диску, файлові дескриптори, сокети та ін. показники).

**Soak Testing** — схожий тип тестування на попередній **Reliability Testing**, але різниця полягає в тому що досліджується працездатність на витривалість при навантаженні вище середнього значення. Soak Testing потрібне щоб дізнатися чи зможе система витримувати навантаження, наприклад високими об’ємами оброблюваних даних та побачити, що відбуватиметься поза дизайнерськими очікуваннями.

**Endurance Testing, Stability Testing —** синоніми до **Soak Testing.**

**Тестування Spike** — це тип Performance Testing, в якому вимірюється продуктивність програми, коли за імпровізованих умов раптово збільшується кількість активних користувачів під час тестування навантаження.

**Usability Testing** — це тестування додатку на предмет: краси, корисності, зрозумілості, легкості та зручності використання користувачами. Наскільки інтерфейс додатку їм імпонує. Його мета є віднайти дефекти та покращити так зване юзабіліті додатку.

Узагалі класно, коли зручність використання тестують кінцеві користувачі, а не тестувальники. Завдання тестувальника… краще коли полягає в підготовці набору практичних значень, розробці детальних користувацьких інструкцій пов’язаних з реальною діяльністю, повторюваних тестових завдань, які повинен буде виконати кожен користувач.

Важливу роль у Usability Testing відіграє збір та аналіз даних, зведення та інтерпретація результатів спостережень, своїх тестерських, зроблених користувачами, з відповідей користувачів на анкети після проведення тестів тощо. На основі цього аналізу розробники вносять в програмне забезпечення відповідні зміни.

Інтерфейс користувача — це зовнішній вигляд програми. А як відомо зустрічають по «одежі». Отже, **UI Testing (Тестування інтерфейсу)**— це вид тестування, який має на меті знайти дефекти графічного інтерфейсу користувача та перевіряти, чи відповідає GUI специфікаціям.

**UI Testing (Тестування користувацького) інтерфейсу включає в себе:**

* перевірку того, як різні елементи інтерфейсу (панелі інструментів, меню, зображення, діалоги, поля введення, списки, кнопки тощо),
* реагують на дії користувача;перевірка того, яким чином додаток обробляє дії користувача з клавіатури та миші.

[**Accessibility Testing**](https://qagroup.com.ua/publications/shcho-take-accessibility-testing/) — це перевірка сайту або додатку на доступність і простоту користування ним людьми з обмеженими можливостями.

**Compatibility Testing (Тестування сумісності)**— тестування програмного забезпечення, призначене щоб побачити, наскільки сумісне програмне забезпечення з певним середовищем — операційною системою, платформою чи обладнанням.

Наприклад:

* Мережевими пристроями;
* Периферія (принтери, мишка, USB, веб-камери);
* Операційними системами (Unix, Windows, MacOS, …);
* Бази даних (Oracle, MS SQL, MySQL, …);
* Системним програмним забезпеченням (веб-сервер, файрволл, антивірус, …);
* Браузерами (Internet Explorer, Firefox, Opera, Chrome, Safari).

**Configuration Testing (Тестування конфігурації)** — це спеціальний вид тестування, спрямований на перевірку роботи програмного забезпечення при різних конфігураціях системи (заявлених у документації: платформах, підтримуваних драйверів, при різних конфігураціях комп’ютерів, спеціалізованого обладнання т. п.) …і оцінка результатів від зміни необхідної конфігурації.

**Тестування локалізації**— це тестування за яким ми оцінюємо налаштування програми (локалізовану версію програми) на певній мові. Наприклад, чи правильно зроблений переклад?«Кракозябри» не з’являються? Стандарт мови правильно увімкнули? Для кирилиці UTF-8.

**Тестування глобалізацією** — це вид тестування, в якому додаток оцінюється крізь призму придатності його функціонування у всьому світі, в різних культурах, на різних мовах, у певному мовному регіоні чи країнах.

**Тестування Безпеки (Security testing)** — це вид тестування програмного забезпечення, який має на меті оцінити та перевірити цілісність системи від стороннього вторгнення. А саме автентифікацію, авторизацію, доступність, конфіденційність і безвідмовність аплікації під тестами.

**Penetration testing, Pentesting (Тест на проникнення)** — це популярна у цілому світі послуга з перевірки інформаційної безпеки системи, а також аналізу ризиків, пов’язаних із забезпеченням цілісного підходу до захисту додатків: атак хакерів, вірусів, несанкціонованого доступу до конфіденційних даних. Цей вид тестування безпеки програмного забезпечення полягає в тестуванні різними зовнішніми методами проникнення, які потенційно міг би застосувати хакер.

**Vulnerability testing** (Тести на вразливість) — охоплює тестування та ідентифікацію вразливостей, недоліків в процедурах безпеки системи, розробці, пошук слабких місць у додатку, що може призвести до порушення політики безпеки системи або, іншими словами, отримання несанкціонованого доступу хакерів.

**Recovery testing (Тестування на відновлення)** — це тип нефункціонального тестування, що включає тестування здатності системи відновитись після помилки чи збою у системі. Найпростіший приклад: вимкнули світло чи живлення.

**Compliance testing (Тестування відповідності)** — це тип нефункціонального тестування, який передбачає перевірку того, що побудований продукт відповідає стандартам, процедурами організації та ін. Як правило **Compliance тестування** проводиться зовнішніми компаніями, яке пропонує, продає, доставляє сертифіковане ПЗ певного застосування.

**Conformance testing** — те ж, що і тестування відповідності.

1. **Тестування стабільності**–**Stability testing** – перевірка працездатності додатку при тривалому тестуванні з очікуваним рівнем навантаження.
2. **Юзабіліті тестування**–**Usability testing** – дослідження для визначення зручності використання ПЗ.
3. **Тестування ефективності**–**Efficiency testing** – перевірка необхідних обсягів коду і ресурсів QA,  що використовуються програмою для виконання окремої функції.
4. **Тестування ремонтопридатності**–**Maintainability testing** – цей підвид нефункціонального тестування визначає наскільки легко підтримувати працездатність системи.
5. **Перевірка портативності**–**Portability testing**– тестування доступності перенесення окремого компонента або всього програмного забезпечення з одного оточення на інше (Windows 8.1 -> Windows 10, Windows -> MacOS).
6. **Тестування «пра-витоків»**–**Baseline testing** – перевірка документації та специфікації, за якою будуть написані тест-кейси. До цього підвиду тестування можна віднести й тестування вимог.
7. **Приймальне тестування**–**Compliance/Acceptance testing** – перевірка продукту на відповідність критеріям готовності.
8. **Тестування документації**–**Documentation testing** – перевірка всієї створеної в рамках тестування документації (від майстер тест-плану до тест-кейсів).
9. **Тестування витривалості системи**–**Endurance testing** – тестування системи при високому навантаженні протягом тривалого періоду часу з метою вивчення її поведінки.
10. **Тестування навантаження**–**Load testing** – як правило, проводиться з метою визначення поведінки ПЗ під очікуваним рівнем навантаження.
11. **Тестування продуктивності**–**Performance testing** – перевірка швидкості роботи ПЗ або його окремих функцій.
12. **Тестування сумісності**–**Compatibility testing** – тестування системи під час роботи в різних середовищах: «залізо», софт частина тощо.
13. **Тестування безпеки**–**Security testing** – проводиться для відповіді на питання «Чи є додаток безпечним/захищеним чи ні?».
14. **Об'ємне тестування**–**Volume testing** – тестування ПЗ з використанням баз даних певного розміру.
15. **Стрес тестування**–**Stress testing** – це тестування в обмежених умовах, наприклад, перевірка поведінки системи (відсутність крешів) за умов нестачі ресурсів комп’ютера (оперативної пам’яті або місця на HDD/SSD дисках).
16. **Тестування швидкості відновлення**–**Recovery testing** – проводиться з метою визначення швидкості відновлення системи у разі софтверного креша (падіння програмного забезпечення) або помилки «заліза».
17. **Тестування локалізації, інтернаціоналізація**–**Localization testing** – перевірка ПЗ на відповідність мовних, культурних та/або релігійних норм. Локалізація – перевірка відображення усіх перекладених текстів програмного забезпечення.