**7 принципів тестування**

Протягом останніх 50 років було запропоновано ряд принципів тестування, які пропонують загальні рекомендації загальний для всіх тестів.

**1. Тестування показує наявність дефектів, а не їх відсутність**

Тестування може показати, що дефекти присутні, але не може довести, що дефектів немає. Тестування знижує ймовірність того, що в програмному забезпеченні залишаться невиявлені дефекти, але, навіть якщо дефектів не буде виявлено, тестування не є доказом правильності.

**2. Вичерпне тестування неможливо**

Перевірка всього (всіх комбінацій вхідних даних і попередніх умов) неможлива, за винятком тривіальних випадків.

Замість того, щоб намагатися провести вичерпне тестування, слід використовувати аналіз ризиків, методи тестування та пріоритети зосередити зусилля на тестуванні.

**3. Раннє тестування економить час і гроші**

Щоб виявити дефекти на ранній стадії, як статичні, так і динамічні випробування слід починати якомога раніше в життєвому циклі розробки програмного забезпечення. Раннє тестування іноді називають «зсувом вліво». Тестування на ранніх етапах життєвого циклу розробки програмного забезпечення допомагає зменшити або усунути дорогі зміни (див. розділ 3.1).

**4. Дефекти групуються разом**

Невелика кількість модулів зазвичай містить більшість дефектів, виявлених під час попереднього тестування, або відповідає за більшість операційних збоїв. Прогнозовані кластери дефектів і фактичні спостережувані кластери дефектів під час тестування чи експлуатації є важливими вхідними даними для аналізу ризиків, які використовуються для зосередження зусиль на тестуванні

(як зазначено в принципі 2).

**5. Остерігайтеся парадоксу пестицидів**

Якщо одні й ті самі тести повторювати знову і знову, зрештою ці тести більше не виявляють нових дефектів.

Для виявлення нових дефектів може знадобитися змінити наявні тести та тестові дані, а також, можливо, знадобляться нові тести написаний. (Тести більше не ефективні для виявлення дефектів, так само як пестициди більше не ефективні для знищення комах через деякий час.) У деяких випадках, таких як автоматизоване регресійне тестування, парадокс пестицидів має сприятливий результат, який полягає у відносно низькій кількості дефектів регресії.

**6. Тестування залежить від контексту**

Тестування проводиться по-різному в різних контекстах. Наприклад, критично важливе для безпеки програмне забезпечення промислового керування протестовано інакше, ніж мобільний додаток для електронної комерції. Як інший приклад, виконується тестування в Agile-проекті відрізняється від тестування в проекті послідовного життєвого циклу (див. розділ 2.1).

**7. Відсутність помилок є помилкою**

Деякі організації очікують, що тестувальники зможуть запустити всі можливі тести та знайти всі можливі дефекти, але принципи 2 і 1, відповідно, говорять нам, що це неможливо. Крім того, це помилка (тобто помилкове переконання) очікувати, що лише пошук і виправлення великої кількості дефектів забезпечить успіх системи. Для наприклад, ретельне тестування всіх визначених вимог і виправлення всіх знайдених дефектів все ще може призвести до a система, якою важко користуватися, яка не відповідає потребам і очікуванням користувачів або є неякісною порівняно з іншими конкуруючими системами.