

Практична робота № 3

Тема: Тест план та робота з ним.

Мета: Навчитися розробляти тест-план

Теоретичні відомості:

Тестувальник звичайно має справу з кількома документами по проекту, але в безпосередньому тестуванні основним для нього є **System Test Plan** (або Software Test Plan, STP, Check-list, Test Specification, Test Cases Document), у побуті «тест-план». Він, в основному, містить безпосередньо самі тести, які команда тестувальників не тільки запускає, але й попередньо їх описує в цьому документі.

Існує ще один документ, що називають теж «тест-план» — **Master Test Plan**, але в нього інші цілі й аудиторія, у ньому знаходяться опис методики тестування, стратегія, ресурси й т.д., але не містяться самі тести. Цим тест-планом займаються тест-менеджери, і тут ми про нього більше згадувати не будемо.

Місце тест-плану в тестовій документації

System Requirements Definition (або Product Requirements Definition, SRD, PRD, вимоги до продукту) — це джерело для тест-плану. Адже головна суть тестування — перевірити, що продукт відповідає задуму замовника, вираженому у вимогах до продукту. Тому, перш ніж видумувати хитромудрі тести, потрібно написати очевидні, чітко відповідні кожній вимозі. Можна сказати, що *тест-план* — це дочірній документ *Вимог*. Зрозуміло, в ідеалі Вимоги повинні бути вже повністю готові й затверджені перед написанням тест-плану, але в житті це трапляється нечасто.

Test Report (тест-звіт) — це для тестувальника практичний результат його роботи, а також вердикт для тестируемой версії продукту, список оцінок проходження кожного тесту у двійковій системі: пройшов (*pass*) або не пройшов (*fail*). Третього не дано. Буває ще *N/A* (not available, немає й не буде можливості виконати тест у цій стадії тестування), *pending* (виникли питання, тест не виконується, тимчасово відкладений до з'ясування обставин), але ці мітки говорять не про якість

продукту, а про якість тест-плану, в ідеальному плані всі нюанси передбачені. Таким чином, *тест-план — це батьківський документ Тест-звітів.*

Коли починати й закінчувати писати тест-план

Може здатися, що тест-план варто починати безпосередньо на стадії тестування, після того як вийшла хоча б перша версія продукту. Адже до цього тест-план і робота самих тестувальників продукту особливо не потрібні. Такий підхід має місце тільки в компаніях, що практикують нерозумну витрату людських і фінансових ресурсів.

Більше вірний і професійний шлях - починати створювати план тестування одночасно із початком практичних робіт над самим продуктом, тобто, відразу після прийняття Вимог, Дизайну й іншої документації, призначеної насамперед для... безпосередніх розроблювачів. Чому, хіба тест-план можна писати, не маючи перед очима сам об'єкт тестування?

Якщо згадати, який документ насправді є вхідним для тест-плану, а також подумати, чого будуть чекати програмісти після випуску першої версії, стане ясно, що кращий час для написання тест-плану - це період створення перших рядків коду об'єкта тестування.

Щоб останнє твердження не звучало категорично, можна згадати про існуючу практику використовувати готовий тест-план як один з документів для... програмістів, що погоджують по ньому очікуване поведження своєї ділянки функціонала.

Який формат документа обрати

Є кілька варіантів для оформлення тест-плану, таких як багатосторінковий текст, електронна таблиця, база даних. Причому немає зв'язку між масштабом компанії, проекту й способом вести документацію, буває, що маленький проект зберігає опису тестів у базі даних, а грандіозний використовує формат RTF.

Звичайно, при виборі формату потрібно виходити з поточних потреб, однак прагнути треба до форми *колективно використовуваної бази даних* або хоча б вибрати формат, що має максимум із цього набору властивостей:

- можливість у потрібний момент внести необхідні зміни в тест-план, які відразу ж доступні всім зацікавленим учасникам процесу;

- можливість негайно одержати додаткову інформацію, таку як початкова вимога до продукту, що «покривається» тим або іншим тестом, або результат виконання конкретного тесту з тест-звітів;
- можливість у будь-який момент швидко й точно скласти звіт про стан тестування: одержати статистику про покриття вимог тестами, кількості тестів взагалі (для планування строків), кількості успішно пройдених тестів, знайдених помилках і т.д., причому все це бажано по окремих компонентах продукту й по різних періодах і фазах тестування;
- можливість розділити доступ до тест-плану, адже комусь можна його змінювати, а комусь тільки читати або не мати доступу зовсім.

Використання інших форматів, що не мають всіх перерахованих властивостей, може викликати змушене дублювання інформації, плутанину у версіях тестів, неприступність останніх змін та інші незручності. Звичайно, така взаємоінтеграція документів можлива, тільки якщо вся документація виконана в такому ж форматі.

Чек-Лист як варіант тест-плану в екстремальних умовах

У ситуаціях, коли не готові ні вимоги, ні дизайн, ні інша документація, що може служити джерелом для тест-плану, а також немає часу на документацію, але якісь тести потрібно скласти й пройти прямо зараз, виходом може стати чек-лист (check-list).

Чек-лист — це список необхідних для перевірки нюансів, елементів, функцій, складений у вільній формі, найчастіше розроблювачем. Тестування за таким списком відбувається теж у вільній формі, на розсуд тестувальника, а не по кроках з конкретними описами очікуваних результатів, як по тест-плану.

Зрозуміло, якість такого тестування залишає бажати кращого, і такий документ не може замінити тест-план, але в екстремальних ситуаціях саме він може бути корисним для швидкого оцінювання програми й знаходження очевидних помилок.

Практичне завдання:

Скласти тест-план для тестування програмного засобу AutoVoiceDetector.

Задані умови: відсутність автоматизованого тестування, щотижневі звіти, 2 тестера персоналу, строк - рік на весь проект, документація та апаратний засіб не підлягають тестуванню, обов'язкове тестування бази даних.

Контрольні питання:

1. Що таке System Test Plan?
2. Що таке Master Test Plan?
3. Місце Test Plan у тестовій документації. Які основні документи використовуються до написання Test Plan і після?