**І рівень**

1. Порівняльна таблиця функціонального, нефункціонального і пов’язаного зі змінами видів тестування.

1.1 Функціональні види

|  | **Що перевіряється** | **Коли застосовується** | **Обмеження** | **Особливості** |
| --- | --- | --- | --- | --- |
| Функціональне тестування | Чи працюють функції відповідно до вимог | В тест-кейсах, при аналізі фактичних та очікуваних результатів  Модульне тестування | Є високий ризик пропустити помилки логіки ПЗ під час перевірки | Може бути проведено відповідно до специфікації, а також відповідно до знань системи |
| Тестування безпеки | Виявляє дефекти, пов'язані з безпекою даних користувача та визначає захищеність системи від різних атак | Для перевірки системи на вразливість до вірусних атак та захищеність даних продукту від зломів  Системне тестування | Даний вид тестування не гарантує повну відсутність небезпеки, а лише зменшує її ризики | Поділяється на тестування конфіденційності, цілісності, ідентифікації, авторизації, доступності та безвідмовності |
| Тестування взаємодії | Перевіряє якість взаємодії між компонентами програми, або всієї програми з ПЗ | Для перевірки взаємодії різних компонентів або системи з програмою, для її коректної роботи  Інтеграційне тестування | Необхідний серйозний підхід, оскільки некоректне тестування може призвести до фінансових втрат | Суть даного виду тестування дозволяє включати в себе тестування сумісності та інтеграційне тестування |

1.2 Нефункціональні види

|  | **Що перевіряється** | **Коли застосовується** | **Обмеження** | **Особливості** |
| --- | --- | --- | --- | --- |
| Стресове тестування | Як поведе себе система при нестандартних умовах роботи, наприклад нестача оперативної пам`яті | При необхідності перевірити систему на роботоспроможність в ситуаціях надмірного навантаження  Системне тестування | Може призвести до негативних наслідків, є складним і дорогим в управлінні | Тестування проводиться із значенням, що перевищує потреби користувачів в декілька разів, зазвичай проводиться після функціонального тестування |
| Тестування установки | Якість інсталяції ПЗ, а також його оновлення та деінсталяцію | При необхідності протестувати застосунок на коректність інсталяції, особливо важливим є для великих важливих проектів, наприклад пов`язаних з банкінгом  Системне тестування | При відсутності інсталяторів потрібно самостійно виконувати встановлення ПЗ, використовуючи документацію у вигляді інструкцій, які крок за кроком описують всі необхідні дії та перевірки | Для даного тестування зазвичай використовується документ Deployment Plan, де покроково описаний весь план тестування |
| Тестування надійності | Чи відповідає ПЗ меті, за якою воно створене, а також протягом якого періоду воно здатне забезпечити надійну роботу | Для виявлення в системі несправностей та причини, що їх спричинили, а також для забезпечення якості системи.  Системне тестування | Потребує багато часу та ресурсів,а також важко тестувати, якщо ПЗ є складним. | Для проведення даного тестування необхідні спеціальні інструменти для вимірювання надійності |

1.3 Тестування, пов’язане зі змінами

|  | **Що перевіряється** | **Коли застосовується** | **Обмеження** | **Особливості** |
| --- | --- | --- | --- | --- |
| Димове тестування | Основний функціонал, тобто перевіряє загальну роботоспроможність продукту | При появі нового білду або при оновленні продукту  Системне тестування | Не є детальним, тестує продукт лише поверхнево | Можна співставити з прийомним тестуванням або тестуванням збірки; добре піддається автоматизації |
| Тестування збірки | Готовність релізу до повного тестування | Перед запуском повного циклу тестування  Системне тестування | Виявляє лише серйозні дефекти | Дозволяє заощадити час тестувальників в випадку виявлення серйозних дефектів. |
| Санітарне тестування | Коротка поглиблена перевірка певної функціональності продукту | При виправленні якої-небудь незначної помилки в системі або є невелика зміна у функціональності.  Модульне  тестування | Перевіряє лише незначні помилки, дані про перевірки зазвичай не документуються | Виконується лише після димового тестування, та при відсутності достатньої кількості часу для детальної перевірки |

1. Порівняння регресії та ретестингу.

Регресійне тестування проводиться на предмет визначення нових помилок при оновленні продукту, або додаванню до нього нових функцій. Ретестинг - це тестування, яке проводиться повторно для підтвердження виправлення певного дефекту. Регресійне тестування виконується після повторного, оскільки має нижчий пріоритет. Тест-кейси зі статусом “Pass” в регресійному тестуванні повинні бути перевірені повторно, тоді коли при ретестингу перевіряються тест-кейси зі статусом “Failed”. В повторному тестуванні, на відміну від регресингу не може бути автоматизованих тест-кейсів.

**ІІ рівень**

1. Чи можливе для продукту проведення тільки функціонального тестування, без перевірки нефункціональних вимог

Я вважаю, що відсутність нефункціонального тестування в продукті можлива, але може бути ряд недоліків, які можуть вплинути на якість продукту. В якомусь сенсі, можна сказати, що створення та тестування нефункціональних вимог спрямовані на потреби користувача або замовника, оскільки саме від цього буде залежати наскільки вони будуть задоволені продуктом. Ще одним нюансом є те, що багато нефункціональних видів тестування є досить затратними та потребують автоматизованого тестування за допомогою спеціальних інструментів. Тому можна зробити висновок, що для великих проектів доцільно використовувати нефункціональне тестування, оскільки це складе краще враження на користувача, знизить фінансові ризики, а також забезпечить безпеку та надійність продукту, що є дуже важливим для глобальних проектів. Для стартапів з низьким бюджетом та низькою складністю проекту, можна провести лише функціональне тестування, яке не потребує додаткових високих витрат. В таких проектах зазвичай не передбачені високі ризики, оскільки розробниками є нові компанії, які ще не знайшли свого користувача. Звичайно, все залежить від масштабу проекту, але зазвичай такі компанії починають з розробки продуктів нижчої складності, отже можна сказати, що до появи попиту на продукт, задля економії коштів можна використовувати лише функціональне тестування.

1. Димне тестування

Smoke testing - це тестування, яке в загальному перевіряє продукт перед релізом. Тобто перевіряє основний функціонал на роботоспроможність. При даному тестуванні можна виявити явні серйозні помилки, що дасть можливість в подальшому сфокусуватись на тестуванні більш незначних помилок за допомогою інших тестів. Отже, даний вид тестування доцільно використовувати для будь-яких проектів, щоб виявити в них значні дефекти перед випуском в експлуатацію або перед більш детальним тестуванням.

**ІІІ рівень**

1. Функціональні тест-кейси до застосунку для обміну світлинами

[6 урок Test-Case](https://docs.google.com/spreadsheets/d/1jy2IIhiEUweIbMKImKtIhaCg9SHCfa3mZOgTjrpaKww/edit?usp=sharing)

1. Нефункціональні вимоги до застосунку

* Додаток повинен коректно встановлюватись та оновлюватися на платформах Android та IOS.

*Тестування установки.*

За допомогою даного виду тестування можна перевірити коректність установки продукту на обох платформах, а також його деінсталяції та оновлення. Для даного тестування можна використати Deployment Plan, де покроково описується повний процес тестування.

* Виведення результатів пошуку повинно виконуватись за 2 секунди

*Тестування продуктивності.*

За допомогою даного тесту можна визначити чи відповідають показники швидкості роботи додатку заданим вимогам.

* Додаток повинен підтримувати українську, англійську і німецьку мови.

*Тестування глобалізації*

За допомогою глобалізації можна перевірити чи коректно працює додаток на даних мовах та орієнтованість додатку на іноземних користувачів.