**9. Довідник**

**Blocker** - термін, який вказує на дефект або проблему, яка має настільки серйозний характер (висока критичність), що блокує нормальне виконання тестових сценаріїв або функціональності програми. Це означає, що цей дефект перешкоджає іншим розробникам або тестувальникам продовжувати роботу з програмним забезпеченням до тих пір, поки він не буде виправлений.

**Critical** - термін, який вказує на дефект або проблему, яка має великий вплив на функціональність (середня критичність), безпеку або відповідність програми і може призвести до серйозних наслідків.

**Major** - вказує на серйозний дефект або проблему, яка може вплинути на коректність роботи програми або на здатність користувача використовувати певну функціональність.

**Minor** - вказує на проблему або дефект, який має незначний вплив на коректність роботи програми або на здатність користувача використовувати певну функціональність.

**Trivial** - вказує на дрібний недолік або помилку, яка має мінімальний вплив на функціональність або використання програми і може бути легко виправлена.

**Білд** - термін, який використовується для позначення конкретної версії або ітерації програмного забезпечення, яка створена з вихідних кодів на певному етапі розробки. Білд включає в себе компіляцію, збірку і пакування програмного забезпечення, яке може бути випущено для тестування, використання або розгортання.

**Баг** - помилка, вада або дефект вкомп'ютерній системі або системі, що викликає в ній неправильний або неочікуваний результат чи неочікувану поведінку.

**Падіння серверу**

(розділ Каті)

**GUI** (Graphical User Interface) - це спосіб взаємодії користувача з комп'ютером із використанням графічних елементів, таких як вікна, кнопки та меню.

**Мануальне тестування** (Manual testing) - при ручному тестуванні (manual testing) тестувальники вручну виконують тести, не використовуючи ніяких засобів автоматизації. Перед тим як автоматизувати тестування будь-якого додатку, необхідно спочатку виконати серію тестів вручну. Мануальне тестування вимагає більших зусиль, але без нього ми не зможемо переконатися в тому, чи можлива автоматизація взагалі.

**Автоматизоване тестування** (Automated testing) **-** припускає використання спеціального програмного забезпечення (окрім того, що тестується) для контролю виконання тестів та порівняння очікуваного і фактичного результату роботи програми. Цей тип тестування допомагає автоматизувати дії, що часто повторюються, але які, в той же час, необхідні для максимального тестового покриття завдання.

**Приймальне тестування** (Acceptancetesting**)** – вид тестування, що проводиться на етапі здачі готового продукту (або готової частини продукту) замовнику. Метою такого тестування є визначення готовності продукту, що досягається шляхом проходу тестових сценаріїв та випадків, які побудовані на основі специфікації вимог до ПЗ, що розробляється.

**Системне тестування** (System testing) – це тестування програмного забезпечення, що виконується на повній, інтегрованій системі, з метою перевірки відповідності системи вихідним вимогам, як функціональним, так і не функціональним.

**Інтеграційне тестування** (Integration testing) – вид тестування, при якому на відповідність вимог перевіряється інтеграція модулів, їх взаємодія між собою, а також інтеграція підсистем в одну загальну систему.

**Модульне тестування** (Unit testing) – тестування кожної атомарної функціональності додатку окремо, в штучно створеному середовищі. Саме потреба у створенні штучної робочого середовища для певного модуля, вимагає від тестувальника знань в автоматизації тестування програмного забезпечення, деяких навичок програмування.

**Функціональне тестування** (Functional testing) – один із видів тестування, спрямованого на перевірку відповідностей функціональних вимог ПЗ його реальним характеристикам. Основним завданням функціонального тестування є підтвердження того, що програмний продукт, який розробляється, володіє усім необхідним замовнику функціоналом.

**Навантажувальне тестування** (Load Testing) – тестування часу відповіді програми на запити різних типів, з метою переконатися, що додаток працює відповідно до вимог при звичайному користувацькому навантаженні.

**Agile** — це методологія гнучкої розробки, яка першочергово сьогодні популярна в ІТ і дозволяє клієнтам швидше отримувати якісне програмне забезпечення.

**Agile Manifesto** — це документ, який описує основні цінності і принципи гнучкої розробки. Маніфест Agile базується і визначає 4 ключові цінності:

* Люди та співпраця важливіші за процеси та інструменти
* Працюючий продукт важливіший за вичерпну документацію
* Позитивна співпраця із замовником важливіша за обговорення умов контракту
* Готовність до змін важливіша за дотримання плану

**Scrum** – це фреймворк, який допомагає команді професіоналів організувати свою роботу максимально ефективно. Як спортивна команда готується до вирішальної гри (до речі, scrum — англ. «битва»), так і Scrum-команда повинна отримувати уроки з набутого досвіду, опановуючи принципи самоорганізації, працюючи над вирішенням проблеми, аналізувати свої успіхи та провали, щоб постійно вдосконалюватись. Scrum сприяє цьому.