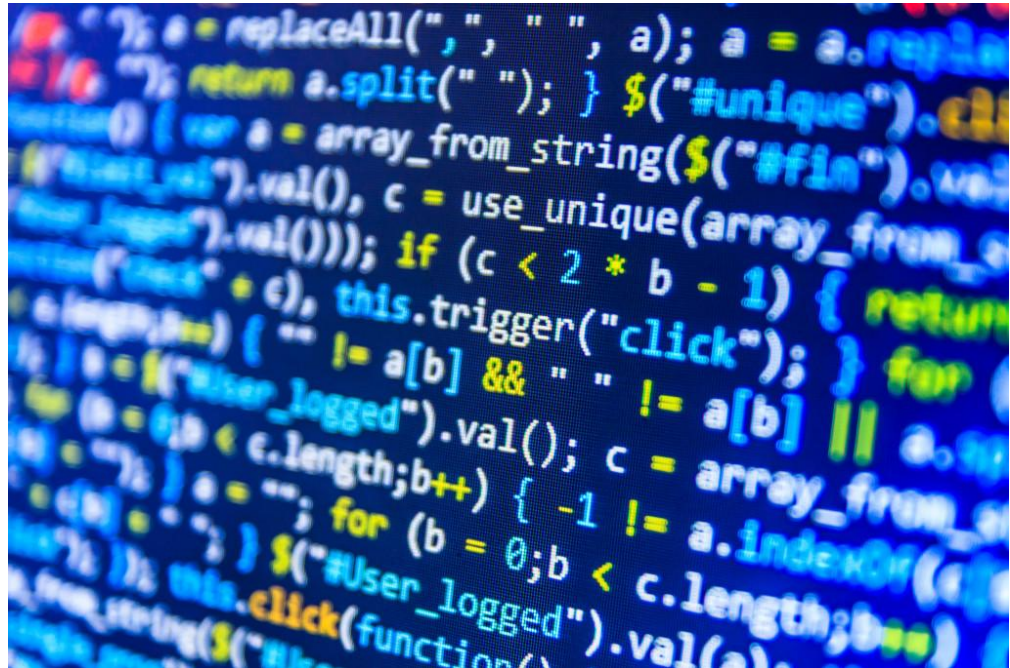


# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Software  
architecture

Луцьк-2022

Лектор: д.т.н., проф. І.Є.Андрющак

# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## ЛЕКЦІЯ 3. ОСОБЛИВОСТІ ТЕСТУВАННЯ АРХІТЕКТУРИ

**Тестування** – техніка контролю якості, що перевіряє відповідність між реальною та очікуваною поведінкою архітектури завдяки кінцевому набору тестів, що обираються певним чином.

**Тестування** - це одна з технік контролю якості, що включає в себе:

- планування робіт (Test Management)
- проектування тестів (Test Design)
- виконання тестування (Test Execution)
- аналіз отриманих результатів (Test Analysis).



Software  
Testing





# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**Тест** – контрольна задача для перевірки коректності функціонування системи та/або її ПЗ.

**Основна ідея тестування** – запустити ПЗ і спостерігати за його роботою та її наслідками. Якщо збій в роботі ПЗ відбувся, тоді аналізується звіт з метою виявлення місцезнаходження помилки, яка його викликала.





# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Вдалим тестом є той, при якому виконання програми закінчилось з помилкою і навпаки.

**Тестування виконує дві основні задачі:**

- 1). демонстрація якості функціонування АПЗ;
- 2). знаходження і усунення помилок в АПЗ.

**Головною метою тестування** є збільшення ймовірності того, що АПЗ, яке тестується, буде відповідати своїм специфікаціям.



# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



BUG FIXING

Тестування - процес ітераційний. Після виявлення і виправлення кожної помилки обов'язково слідує повторення тестів, що має на меті перевірку працездатності програми. Більш того, для ідентифікації причини виявленої проблеми може бути потрібно проведення спеціального додаткового тестування.



# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**Тестування (testing)** – це:

- процес використання АПЗ при умові аналізу або запису отримуваних результатів з метою перевірки (оцінки) деяких властивостей тестованого об'єкту.
- процес аналізу АПЗ з метою виявлення відмінностей між створеним станом АПЗ та зазначеним у специфікації (що свідчить про прояв помилки) при експериментальній перевірці відповідного пункту вимог.
- контрольоване виконання програми на множині тестових даних і аналіз результатів цього виконання для пошуку помилок.





# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 2. Історія розвитку тестування програмного забезпечення.

Перші програмні системи розроблялися в рамках програм наукових досліджень або програм для потреб міністерств оборони. Тестування таких продуктів проводилося строго формалізовано із записом всіх тестових процедур, тестових даних, отриманих результатів. Тестування виділялося в окремий процес, який починався після завершення кодування, але при цьому, як правило, виконувалося тим же персоналом.

### SOFTWARE TESTING SERVICES

Improve your software quality while decreasing your testing cost and accelerating your product's time-to-market.



# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



У 1960-х багато уваги приділялося «вичерпному» тестуванню, яке повинно проводитися з використанням усіх шляхів у коді або всіх можливих вхідних даних.

На початку 1970-х тестування ПЗ розглядалося як «процес, спрямований на демонстрацію коректності продукту» або як «діяльність з підтвердження правильності роботи ПЗ».







# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



У середині 1990-х з розвитком Інтернету й розробленням великої кількості веб-застосунків особливої популярності стало набувати «гнучке тестування» (за аналогією з гнучкими методологіями програмування).

У 2000-х з'явилося ще більш широке визначення тестування, коли в нього було додано поняття «оптимізація бізнес-технологій».





# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**Несправність, Дефект (Fault)** – визнана або передбачувана причина помилки; наслідок відмови деякої системи, що обслуговувала або обслуговує в даний момент часу розглянуту систему. Дефекти також часто називають - **багами** (від англ. - bugs - жучки). Цей термін використовують, якщо вплив дефекту на роботу програми невеликий. Якщо ж помилка пов'язана із специфікаціями або архітектурою програми, то використовують слово – «дефект».





# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**Верифікація (Verification)** - це процес оцінки системи або її компонентів з метою визначити чи задовольняють результати поточного етапу розробки умовам, сформованим на початку цього етапу. Тобто чи виконуються цілі, терміни, завдання з розробки проекту, визначені на початку поточної фази.



# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**Валідація (Validation)** - це визначення відповідності розроблюваного програмного забезпечення між очікуваннями і потребами користувача, вимогам до системи.



# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**План Тестування (Test Plan)** - це документ, що описує весь обсяг робіт з тестування, починаючи з опису об'єкта, стратегії, розкладу, критеріїв початку і закінчення тестування, до необхідного в процесі роботи обладнання, спеціальних знань, а також оцінки ризиків з варіантами їх вирішення.

**Тест дизайн (Test Design)** - це етап процесу тестування програмного забезпечення, на якому проектується і створюються тестові випадки (тест кейси), відповідно до визначених раніше критеріями якості та цілями тестування.





# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

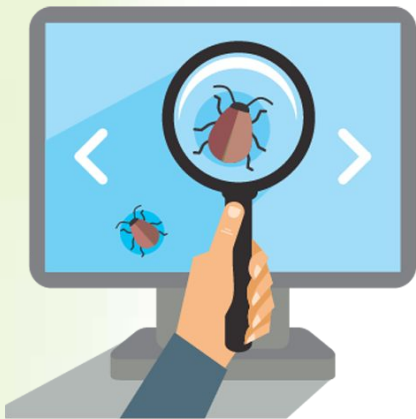
**Тестовий випадок (Test Case)** - це документ, що описує сукупність кроків, конкретних умов і параметрів, необхідних для перевірки реалізації тестованої функції або її частини.

**Баг/Дефект Репорт (Bug Report)** - це документ, що описує ситуацію або послідовність дій (Steps), що призвела до некоректної роботи об'єкта тестування (Misbehavior), із зазначенням причин та очікуваного результату (Expected Result).





# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



**Деталізація Тест Кейсів (Test Case Specification)** - це рівень деталізації опису тестових кроків і необхідного результату, при якому забезпечується розумне співвідношення часу проходження до тестового покриття.

**Час Проходження Тест Кейса (Test Case Pass Time)** - це час від початку проходження кроків тест кейса до отримання результату тесту.

**Тестове Покриття (Test Coverage)** - це одна з метрик оцінки якості тестування, що представляє із себе щільність покриття тестами вимог або коду, що виконується.



# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Якщо при наявності несправності у кодї ПЗ відповідає умовам зазначеним вище, то воно вважається *придатним для тестування (testable)*.

Testing  
Software





# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**Багтрекер** - прикладна програма, розроблена, щоб допомогти тестерам та програмістам відстежувати історію звітів про баги під час своєї роботи. Він може розглядатись як різновид issue tracking system.

**Головний компонент багтрекера** - база даних, що записує факти про відомі баги.

**Локальний багтрекер (LBT)** - звичайно комп'ютерна програма, використовувана командою професійних підтримувачів додатку (часто служба технічної підтримки) відстежування of issues communicated to розробникам програмного забезпечення.

**Bugzilla** (Баґзілла) - система відстеження помилок і ведення завдань (англ. *bugtracker tool*) з веб-оболонкою.

# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Testing



**Якість (Quality)** – ступінь відповідності системи, компоненту або процесу заданим вимогам, потребам або очікуванням користувача. З метою визначення добротності системи, компоненту або процесу використовують так звані атрибути якості – характеристики, що відображають дану властивість.

**Метрика (Metrics)** – кількісна міра ступеня наявності атрибута системи, компоненту або процесу.



# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Приведемо декілька прикладів атрибутів якості з коротким описом:

- властивість, що відповідає за безперервність коректного протікання процесу, відноситься до **надійності (reliability)**;

- властивість, що відповідає за постійну готовність системи, відноситься до **готовності (availability)**;

- властивість, що відповідає за відсутність катастрофічних наслідків в системному середовищі, відноситься до **безпеки (safety)**;

- властивість, що відповідає за запобігання несанкціонованому доступу до інформації, відноситься до **конфіденційності (confidentiality)**;

Software  
TESTING



# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



- властивість, що відповідає за відсутність появи в системі невідповідних змін інформації, відноситься до ***цілісності (integrity)***;

- властивість, що відповідає за здатність системи піддаватися ремонту і розвитку, відноситься до ***ремонтпридатність (maintainability)***.

- властивість, що відповідає за здатність – рівень зусиль, необхідних для навчання, роботи, підготовки вхідних і обробки вихідних даних АПЗ, відноситься до ***зручності роботи (usability)***.







# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## ЗАПИТАННЯ?



# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**ДЯКУЮ  
ЗА УВАГУ!!!**

