

# Якість програмного забезпечення та тестування

Якість та місце тестування в  
життєвому циклі ПЗ.

**ГАЛУЗЕВИЙ СТАНДАРТ ВИЩОЇ ОСВІТИ УКРАЇНИ  
ОСВІТНЬО-КВАЛІФІКАЦІЙНА ХАРАКТЕРИСТИКА – бакалавр  
НАПРЯМУ ПІДГОТОВКИ 050103 - «ПРОГРАМНА ІНЖЕНЕРІЯ»**

**Виробничі функції, типові задачі діяльності та уміння, якими повинні володіти випускники вищого навчального закладу**

*Назва виробничої функції*

**4. Технологічна**

*Назва типової задачі діяльності*

**Верифікація та атестація ПЗ**

*Зміст уміння*

**Визначати та вимірювати атрибути якості**

**Здійснювати модульне та комплексне  
тестування ПЗ**

Якість важлива !!!!!



# Якість програмного продукту

характеризується набором властивостей, що визначають, наскільки продукт "добрий" з погляду Зацікавлених сторін:

- ✓ замовник продукту,
- ✓ спонсор,
- ✓ кінцевий користувач,
- ✓ розробники і тестувальники,
- ✓ інженери підтримки,
- ✓ співробітники відділів маркетингу, навчання і продажів.

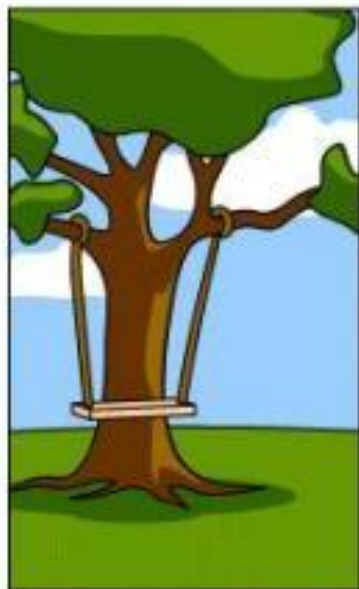
# Задача забезпечення якості ПЗ

включає;

- визначення зацікавлених осіб
- їхніх критеріїв якості
- знаходження оптимального рішення, що задовольняє ці критерії.



How the customer explained it



How the Project Leader understood it



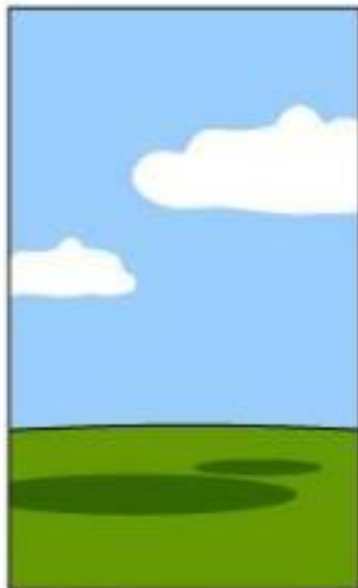
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



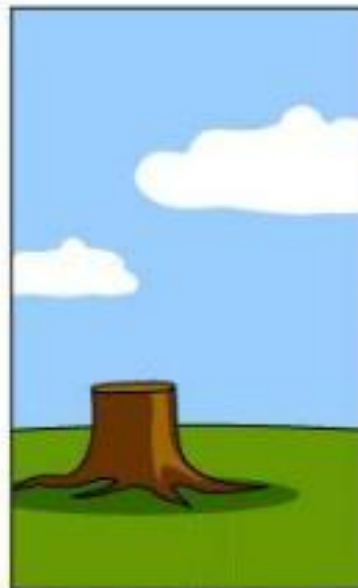
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

# Тестування

- є одним з найбільш важливих способів забезпечення якості ПЗ
- полягає у виконанні програмного продукту на деякій множині вихідних даних і порівнянні одержаних результатів із заздалегідь відомими (еталонними) для встановлення відповідності вимогам різних властивостей і характеристик продукту.

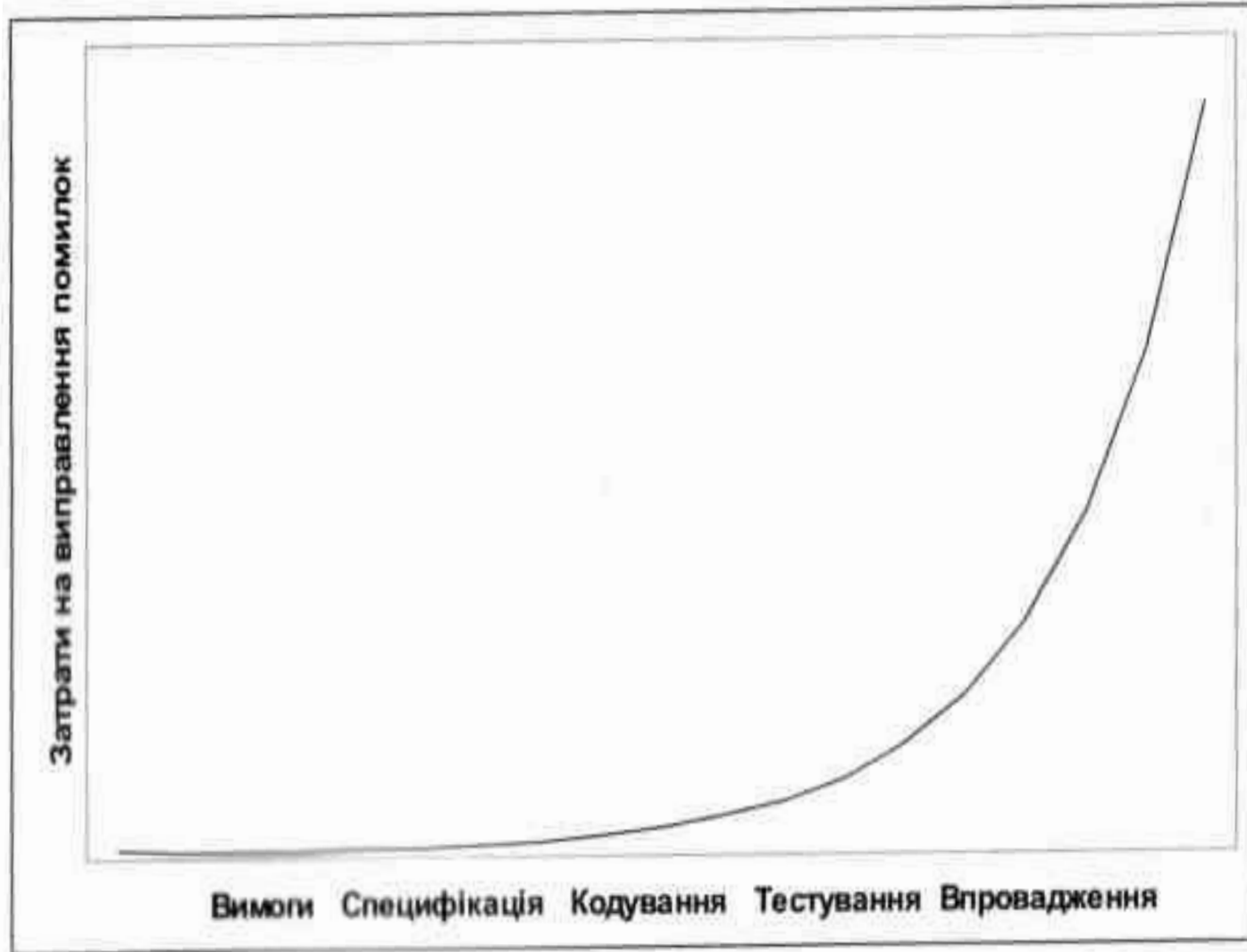
# Тестування

- **Характеризується досить великим внеском до сумарної трудомісткості розроблення продукту**
- **Оцінка розподілу трудомісткості між фазами -**

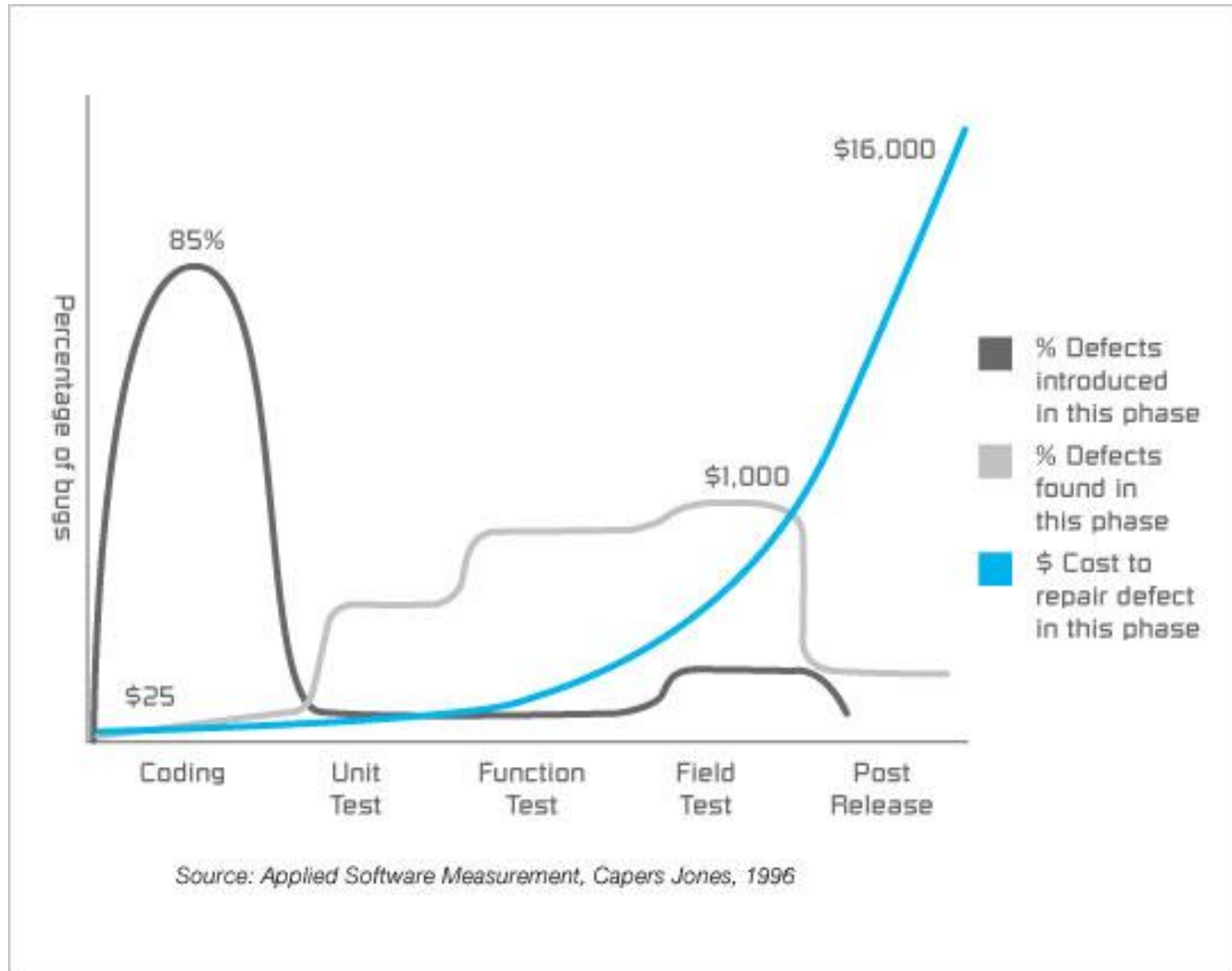
**Проектування - Розроблення коду – Тестування**  
40 % - 20 % - 40 %



# Ціна виправлення помилки



# Ціна виправлення помилки



# Задача найближчого майбутнього

**- рух у напрямку розподілу трудомісткості**

**Проектування - Розроблення коду – Тестування**

**60 %**

**-**

**20 %**

**-**

**20 %**

# *Методи контролю якості*

- Як контролювати якість програмної системи?
  - Як точно дізнатися, що програма робить саме те, що потрібно, і нічого зайвого?
  - Як визначити, що вона достатньо надійна, переносима, зручна у використанні?
- за допомогою процесів  
верифікації і валідації (атестації)

# Верифікація

- перевірка того, що продукт робився правильно, тобто, що він розроблявся відповідно до всіх вимог стосовно процесу та етапів розроблення.

# Валідація

- перевірка того, що сам продукт правильний, тобто встановлення того, що він задовольняє вимоги, очікування користувача, замовника та інших зацікавлених сторін.

# **Тестування - частковий варіант верифікації і валідації**

- Полягає в спостереженні за роботою ПЗ у спеціальних умовах, на деяких даних з метою перевірки відповідності його властивостей вимогам.**
- Такі спеціальні умови називають тестовими, або просто тестами.**

# Види тестування

- **Функціональне ("чорної скриньки")**
  - **за цим тестуванням перевіряють відповідність системи вимогам, користуючись при цьому як для створення тестів, так і для оцінки ступеня відповідності переважно вимогами і специфікаціями.**



# Види тестування

- **Структурне ("білої скриньки")**
  - **тести створюються на основі знань про структуру самої системи і про те, як вона працює. Для оцінювання ступеня відповідності вимогам можуть залучатися додаткові знання про прослідковування вимог до значень внутрішніх даних системи в певних умовах (наприклад, обмеження на значення параметрів викликів, результатів і локальних змінних).**

# Види тестування

- Тестування продуктивності.
  - вимірюються показники продуктивності системи і визначається, наскільки вони відповідають потрібним, а також, чи не може екстраполяція цих даних призвести до порушення вимог в інших можливих ситуаціях. У разі виникнення серйозних підозр у другому випадку тести продуктивності допрацьовуються для перевірки цих додаткових ситуацій.

# Види тестування

- **Навантажувальне тестування.**
  - **особливий вид тестування продуктивності, метою якого є визначення характеристик продуктивності системи за великого навантаження на неї (великій кількості користувачів, інтенсивному обміні даними з іншими системами, великим обсягом переданих/використовуваних даних та ін.).**

# Види тестування

- **Тестування на відмову (smoke testing)**
- **має на меті виявлення ситуацій, в яких система виходить з ладу, хоча не повинна робити цього.**

# **Види тестування**

- **Тестування зручності використання**
  - **перевіряє, наскільки система зручна у використанні. Робиться це на основі вимірювання часу, необхідного на навчання роботі з системою, і продуктивності праці під час роботи з системою різних груп користувачів (вже знайомих з нею, незнайомих, досвідчених у цій предметній області, новачків та ін.).**

# **Види тестування**

- **Тестування переносимості**
  - **перевіряє працездатність системи у різних оточеннях.**
  
- **Тестування на відповідність**
  - **перевіряє строгу відповідність вимогам будь-якого роду, зафіксованим у достатньо строгому і повному вигляді (специфікаціям, стандартам, корпоративній політиці та ін.).**

# **Види тестування**

- Модульне, інтеграційне і системне тестування - це рівні тестування, які розрізняються масштабом цілей.**

**Модульне тестування призначене для перевірки правильності роботи окремих модулів, незалежно від оточення**

**Інтеграційне - для перевірки того, що модулі коректно працюють разом (використовують один одного, не порушуючи взаємних обмежень на таке використання)**

**Системне - для перевірки правильності роботи системи загалом та її здатності вирішувати заявлені завдання в різних ситуаціях.**

# Інспекція

- **цілеспрямоване вивчення коду і документів на предмет пошуку помилок певного виду (за заздалегідь визначеним набором шаблонів), двозначностей, розбіжностей із стандартами оформлення, невідповідностей між окремими документами або частинами одного документа.**



# **Формальний аналіз**

- формальне доведення властивостей ПЗ і формальний аналіз ефективності алгоритмів.**

# **Неформальний аналіз**

- аналіз статичної семантики мов програмування, автоматизований аналіз коду на предмет виявлення сумнівних місць, аналіз властивостей ПЗ і архітектури.**

# **Вимірювання**

**- визначення метрик ПЗ, проекту, документації, вимірювання продуктивності, вимірювання трудомісткості роботи з ПЗ, швидкості навчання, профілізація (вимірювання часу роботи окремих елементів коду в межах реалістичних сеансів роботи, вимірювання витрат пам'яті на окремі види об'єктів, окремими процедурами).**

# **Використання моделей**

- Моделі використання (які операції використовує користувач, наскільки часто, як довго він працює з окремими екранами і діалогами) для аналізу зручності використання ПЗ;**
- Моделі використання і надійності для аналізу надійності і продуктивності;**
- Моделі функціонування для аналізу властивостей та їх перевірки на моделях.**

# Прототипування

- **побудова прототипів системи для оцінювання досяжності тих або інших цілей і отримання відгуків від зацікавлених осіб.**

# Стандарти стосовно якості

- IEEE 730-2002 Standard for Software Quality Assurance Plans. Описує структуру планування забезпечення якості.
- IEEE 829-1998 Standard for Software Test Documentation, Описує види документів, що слугують для підготовки тестування.
- IEEE 982.1-1988 Standard Dictionary of Measures to Produce Reliable Software.
- IEEE 1008-1987 (R1993, R2002) Standard for Software Unit Testing

# Стандарти стосовно якості

- IEEE 1012-1998 Standard for Software Verification and Validation.
- IEEE 1028-1997 (R2002) Standard for Software Reviews.
- IEEE 1044-1993 (R2002) Standard Classification for Anomalies.
- IEEE 1061-1998 (R2004) Standard for Software Quality Metrics
- Methodology.  
IEEE 1228-1994 (R2002) Standard for Software Safety Plans.

# Стандарти стосовно якості

- IEEE 1465-1998 [Adoption of ISO/IEC 12119: 1994(E)], Standard Adoption of International Standard ISCMEC 12119: 1994(E) Information Technology - Software packages. Quality requirements and testing.
- ISO/IEC 12119:1994 (AS/NZS 4366:1996) Software packages -Quality requirements and testing

# Стандарти стосовно якості

- ISO 9126-1-4. IT. Якість програмних засобів:  
Ч. 1. Модель якості. Ч. 2. Зовнішні метрики.  
Ч. 3. Внутрішні метрики. Ч. 4. Метрики  
якості у використанні.



# Контрольні запитання

- Чим характеризується якість ПЗ?
- Чому важливим є виявлення помилок на ранніх стадіях програмного проекту?
- Які методи контролю якості ви знаєте?
- Що таке верифікація програмного продукту?
- Що таке валідація програмного продукту?
- Які є види тестування програмного продукту?

# Наступне

- Методи пошуку помилок в програмах.

