**Міністерство Освіти І НАУКИ**

**Національний університет "Львівська політехніка"**

Інститут **ІКНІТ**

Кафедра **ПЗ**

### ЗВІТ

До лабораторної роботи № 2

**На тему:***“ Створення плану забезпечення якості (SQAP) для «Дослідження технології Hadoop», частина 2*

**З дисципліни:** *“ Стандарти та засоби керування якістю процесу розробки*

*програмного забезпечення ”*

**Виконав:**

ст. гр. ІПЗм-11

Крохмальна О. П.

**Прийняв:**

доц. кафедри ПЗ

Білас О. Є.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2015 р.

∑= \_\_\_\_ .

Львів – 2015

**Тема роботи:** Створення плану забезпечення якості (SQAP) для «*Дослідження технології Hadoop*», частина 2.

**Мета роботи:** Розробити другу частину плану забезпечення якості (SQAP) для «*Дослідження технології Hadoop*».

**Теоретичні відомості**

SQAP включає розділи, відповідно до стандарту *IEEE 730 Standard for Software Quality Assurance Plans (Стандарт для планування забезпечення якості ПЗ - IEEE 730)*.

SQAP складається з наступних пунктів:

Частина 1

1. Мета.
2. Управління.
3. Документація
4. Стандарти, практики, узгодження і метрики.

Частина 2

1. Тестування.
2. Звіти про помилки та коригувальні дії.
3. Засоби, методи та методології.
4. Контроль середовища.
5. Контроль постачання.
6. Збір, підтримка і зберігання обліку.
7. Навчання.
8. Управління ризиками.

Деякі матеріали можуть бути наведені в інших документах. В такому випадку посилання на ці документи будуть присутні в SQAP.

**Хід роботи**

## 5. Тестування

Проектування та опис тестових випадків здійснюється згідно стандарту ISO 839. Процес тестування, повинен відбуватись за планом функціонального тестування. Також для програмної системи буде проведено навантажувальне тестування та тестування продуктивності.

Мета функціонального тестування – виявлення невідповідностей між реальною поведінкою реалізованих функцій і очікуваною поведінкою відповідно до специфікації і вимог. Функціональні тести повинні охоплювати всі реалізовані функції з урахуванням найбільш ймовірних типів помилок. Тестові сценарії, що поєднують окремі тести, орієнтовані на перевірку якості розв'язку функціональних задач.

Функціональні тести створюються за зовнішніми специфікаціями функцій, проектної інформації і за текстом на МП, що стосуються його функціональних характеристик і застосовуються на процесі комплексного тестування й іспитів для визначення повноти реалізації функціональних задач і їхньої відповідності вхідним вимогам.

Тестування навантаження (Performance and Load Testing) - визначення масштабованості додатків під навантаженням, при цьому відбувається: вимір часу виконання вибраних операцій за певних інтенсивностей виконання цих операцій; визначення кількості користувачів, що одночасно працюють з додатком; визначення меж прийнятної продуктивності при збільшенні навантаження (при збільшенні інтенсивності виконання цих операцій); дослідження продуктивності при високих, граничних, стресових навантаженнях;

Стресове тестування (Stress Testing) дозволяє перевірити наскільки додаток і система в цілому працездатні в умовах стресу і також оцінити здатність системи до регенерації, тобто до повернення до нормального стану після припинення впливу стресу. Стресом у даному контексті може бути підвищення інтенсивності виконання операцій до дуже високих значень або аварійна зміна конфігурації сервера. Також одним із завдань при стресовому тестуванні може бути оцінка деградації продуктивності, таким чином цілі стресового тестування можуть перетинатися з цілями тестування продуктивності;

## 6. Звіти про помилки та коригувальні дії

Для звітності будуть використовуватись текстові документи захищені від зовнішнього втручання. Опис програмних дефектів здійснюється згідно стандарту ISO 839.

Керівник проекту повинен визначити процедуру звітності про проблеми в забезпеченні якості. Вимоги до процедури звітності:

* Для автоматизації звітності потрібно використовувати такі розподілені багтрекери, як Bugs Everywhere, DisTract і Fossil, які дозволяють легко читати, додаватити до бази даних чи оновлювати звіти про помилки, поки розробник не в мережі;

Звіт про проблему повинен містити:

* Відомості про частину системи, яка порушує умови якості;
* Кроки, які привели до помилки;
* Спостереження за поведінкою системи, має відображати, що працює не правильно;
* Очікувана поведінка системи, містить інформацію про правильну роботу системи.

## 7. Засоби, методи і метрики

Для підтримки SQA процесів використовуватиметься Rational Rose.

Інженер з забезпечення якості повинен вимірювати здатність команди передбачати та долати ризики, здавати проект у заплановані сроки та вести статистику порушень якості продукту відносно порушених вимог, програмних модулів, порушників тощо.

Процес збору метрик контролюється і доповнюється керівником проекту.

## 8. Контроль середовища

Для визначення середовища для кожного проміжного результату та документації, у тому числі процесу копіювання і відновлення буде застосовуватись додаткове програмне забезпечення, а саме GitHub, який надає можливість резервного копіювання проекту.

Проект буде знаходитись під захистом відповідного програмного забезпечення на окремо відведеній для розробки цього ж проекту машині, що забезпечить його від несанкціонованого доступу чи пошкодження на всіх етапах життєвого циклу програмного забезпечення.

## 9. Контроль розробника

У цьому розділі повинні бути вказані положення, які гарантують, що програмне забезпечення, що надається розробниками відповідає встановленим вимогам. Крім того, в цьому розділі зазначаються методи, які будуть використовуватися для забезпечення того, що отримувач отримує програмне забезпечення адекватне і відповідне вимогам. Для раніше розроблених програм, цей розділ повинен вказати методи, які будуть використовуватися для забезпечення придатності продукту для використання засобів програмного забезпечення, описаних у SQAP.

## 10. Збір коментарів, їх підтримка і збереження

В процесі забезпечення якості продукту інженер повинен збирати таку осовну інформацію про стан якості на проекті:

* Звіти про порушення якості
* Ризики
* Вимоги до системи
* Стандарти
* Плани тестувань

Оскільки для створення даного програмнго засобу буде використовуватись мова програмування java, тому для збору коментарів на проекті буде використовуватись інструмент javadoc, який є частиною пакета JDK.

## 11. Навчання

Навчальними заходами для виконання проекту буде залучено експерта по забезпеченню якості програмного забезпечення та проведення декількох тренінгів з командою по забезпеченню якості.

## 12. Управління ризиками

Головною метою управління ризиками на проекті є покращення процедури планування. Простежуючи характеристики проблем, які сповільнюють роботу проекту, та визначаючи ймовірність виникнення схожих у майбутньому, можна простежити причини проблем та будувати гнучкіші плани.

Основним правилом для управління ризиками на проекті є зазначення ризику невиконання завдання на етапі планування. Пізніше робиться оцінка точності ризиків та інших їх характеристик, що випливають із діяльності, щодо якої був визначений ризик.

Для управління ризиками використовується програмний засіб Project in a Box.

**Висновки**

**В ході виконання лабораторної роботи** було розроблено другу частину плану забезпечення якості (SQAP) для «*Дослідження технології Hadoop*». Ця частина описує наступні засоби:

* засоби для тестування програмного продукту;
* засоби для автоматизації звітів про помилки, засоби для підтримки SQA процесів;
* засоби для резервного копіювання проекту;
* засоби для збору коментарів.

Також описує заходи з підготовки кадрів, які необхідно провести для задоволення потреб SQAP.