**Міністерство Освіти І НАУКИ**

**Національний університет "Львівська політехніка"**

Інститут **ІКНІТ**

Кафедра **ПЗ**

### ЗВІТ

До лабораторної роботи № 1

**На тему:***“ Створення плану забезпечення якості (SQAP) для «Дослідження технології Hadoop», частина 1*

**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 продукту «*Дослідження технології Hadoop*»*, частина 1*.

**Мета роботи:** Розробити першу 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**

1. **Мета**

Цей план забезпечення якості встановлює процеси, методи, стандарти і процедури які будуть використовуватись для встановлення та забезпечення якості програмного продукту «*Дослідження технології Hadoop*».

Для розробки проекту буде використана каскадна модель. Каскадну модель можна розглядати як модель ЖЦ, придатну для створення першої версії ПЗ з метою перевірки реалізованих в ній функцій. При супроводі і експлуатації можуть бути виявлені різного роду помилки, виправлення яких спричинить повторне виконання всіх процесів, починаючи з уточнення вимог. Особливість такої моделі полягає у фіксації послідовних процесів розроблення програмного продукту. В її основу покладена модель фабрики, де продукт проходить стадії від задуму до виробництва, потім його передають замовнику у вигляді готового виробу, де заміна не передбачена, хоча можна подати інший подібний виріб. Недоліки цієї моделі такі:

* процес створення ПС не завжди вкладається в таку жорстку форму і послідовність дій;
* не враховуються змінювані потреби користувачів, нестабільні умови зовнішнього середовища, які впливають на зміни вимог до ПС під час її розроблення;
* значний розрив між часом внесення помилки (наприклад, на процесі проектування) і часом її виявлення (при супроводі), що призводить до суттєвої переробки ПС.

Мета плану забезпечення якості полягає у створенні набору дій для перевірки процесів забезпечення якості та в результаті виконати формування документа по керуванню якості.

1. **Управління**

Цей розділ описує структуру організації проекту, її завдання, ролі та обов'язки

1. **Організація**

Кожен учасник команди відповідає за якість виконання своєї роботи. Для виконання проекту призначається відповідальний за якість ПП. Відповідальний за якість керує всіма питаннями, пов'язаними із забезпеченням якості в проекті.

1. **Завдання**

Завдання з контролю якості включають в себе:

* Документування;
* Оглядові наради (експертизи, інспекції);
* Верифікацію;
* Валідацію;
* Види діяльності, спрямовані на поліпшення самого процесу забезпечення якості.

Ці завдання деталізовані в даному документі.

* 1. **Ролі та обов'язки**

За управління процесом забезпечення якості відповідає безпосередньо інженер з забезпечення якості. Його робота координується керівником проекту. Всі питання валідуються керівником продукту, що є представником замовника.

***Роль інженера з забезпечення якості***

До обов’язків інженера з забезпечення якості належать:

* створення та підтримка плану забезпечення якості;
* виконання положень плану забезпечення якості, визначених документом, для перевірки дотримання якості та відповідності вимогам програмного продукту;
* надання необхідних пояснень, щодо цього документу, процесу забезпечення якості на проекті;

*Інженер по забезпеченню якості має право:*

* вимагати від учасників команди дотримання якості при розробці програмного продукту;
* отримати, будь яку, інформацію, необхідну для формування плану забезпечення якості чи контролю за дотриманням якості на проекті.

***Роль керівника проекту***

До обов’язків керівника проекту (з точки зору процесу забезпечення якості) належать:

* затвердження положень та змін до процесу забезпечення якості;
* контроль за дотриманням виконання обов’язків інженером з забезпечення якості;
* організація процесу обміну інформацією стосовно забезпечення якості продукту.

*Керівник проекту має право*:

* вимагати від учасників команди дотримання якості при розробці програмного продукту;
* вимагати у інженера обгрунтування положень плану, чи процесу забезпечення якості;
* вимагати у інженера зміни положень плану, чи процесу забезпечення якості, якщо вони суперечать визначеному курсу проекту.

## *Роль керівника продукту*

До обов’язків керівника продукту належать:

* надання своєчасної інформації, необхідної для вирішення питань команди розробників
* підтримка інформованості сторін відносно змін однієї із них;

*Керівник продукту має право:*

* на отримання детального пояснення аспектів розроблюваної системи, процесу розробки та забезпечення якості продукту
* ініціювати зміну аспекту розроблюваної систем, процесу розробки та забезпечення якості продукту, якщо така зміна буде сприяти покращенню продукту та його відповідності до очікувань замовника

Керівник проекту може виконувати роль керівника продукту та інженера по забезпеченню якості продукту одночасно.

### 3. Документація

### 3.1. Мета

### У даному розділі визначається документація, що описує забезпечення якості і буде використана в рамка виконання проекту «*Дослідження технології Hadoop*».

### 3.2. Мінімальні вимоги до документації

Повинні бути створені такі документи (наприклад):

* SQAP: План забезпечення якості (даний документ).
* SCMP: План конфігураційного управління.
* SRD: Специфікація вимог до програмного забезпечення.
* SDD: Дизайн програмного забезпечення.
* STD: Документація з тестування програмного забезпечення.
* Плани проведення верифікації та валідації.
* План по виявленню та класифікації ризиків.
* План супроводу.

#### 3.2.1. Опис вимог до програмного забезпечення (SRD)

SRD визначає вимоги для програмного продукту, які виконують певні функції в конкретному середовищі. Опис вимог повинен визначати вимоги для конкретного програмного продукту, програми або набору програм, які виконують певні функції в конкретному середовищі. Документ може бути записаний постачальником (внутрішнім або зовнішнім), замовником, або обома. У ньому слід розглянути основні питання функціональності, зовнішні інтерфейси, продуктивність, атрибутів та обмежень на реалізацію. Кожна вимога повинна бути однозначно ідентифікована та визначена так, що її задоволення здатне бути об'єктивно перевіреним і підтвердженим.

Опис вимог складається з метою формування спільного бачення роботи системи всім учасниками розробки та надає вичерпний перелік вимог, розділених за їх класами. Документ формується на початкових етапах проекту та може доповнюватись по мірі необхідності.

Кроки для підтвердженння відповіднсті опису вимог:

* При внесенні зміни в опис вимог документ провіряється на відповідність:
* внутрішні практикам та узгодження;
* міжнародним стандартам;
* Кожна нова\змінена вимоги повинні верифікується керівником продукту;

Критерії відповіднсоті для опису вимог:

* Відповідність установленим практикам та домовленостям;
* Відповідність стандартам;
* Затвердження власником продукту.

***3.2.2. Опис дизайну програмного забезпечення (SDD)***

SDD описує, як програмне забезпечення буде будуватися відповідно до вимог SRD. Документ повинен описувати компоненти й підкомпоненти розробки програмного забезпечення, включаючи бази даних і внутрішні інтерфейси. Він може бути підготовлений як архітектура продукту.

Кроки для підтвердження опису дизайну ПЗ:

* При внесенні змін в опис дизайну ПЗ документ перевіряється на відповідність:
* функціональним та нефункціональним вимогам до продукту згідно опису вимог;
* установленим практикам та стандартам.
* Потім документ презентується керівнику продукту;
* У випадку несхвалення документ потрібно переробити з урахування зауважень керівника;
* При необхідності збереження поточних змін в документі опису дизайну ініціюється зміна до опису вимог, що обговорюється з керівником продукту.

Критерії для підтвердження опису дизайну ПЗ:

* Інформованість керівника продукту щодо документу;
* Відповідність стандартам та визначеним практикам;
* Задоволення опису вимог до ПЗ.

#### 3.2.3. Плани верифікації та валідації

Верифікація та валідація використовується для визначення того, чи розроблені програмні продукти відповідають вимогам, і чи є програмне забезпечення є відповідним до призначення і очікуванням користувачів. План включає в себе: аналіз, контроль, оцінку і тестування програмного продукту і процесів, які стосуються випуску продукту. Крім того, тестування програмного забезпечення, атестація і перевірка процесів застосовуються при інтеграції зовнішніх частин розроблюваного продукту.

План визначає завдання верифікації та валідації, входів і виходів, необхідних для підтримки потрібного рівня цілісності програмного забезпечення. Він також надає засоби тестування (інтеграційного та системного рівня) та розробки вимог SRD, щодо розробки, як описано в SDD, і щодо тестування, як описано в проектній документації.

Кроки для підтвердження плану варифікації та валідації:

* При внесенні змін в план верифікації та валідації документ провіряється на відповідність установленим практикам та стандартам;

Далі документ презентується керівнику продукту;

* У випадку несхвалення документ потрібно переопрацювати з урахування зауважень керівника;

Критерії для підтвердження плану варифікації та валідації:

* Інформованість керівника продукту щодо документу;
* Відповідність стандартам та визначеним практикам.

#### 3.2.4. Користувацька документація

Користувацька документація містить опис встановлення, експлуатації, управління та обслуговування (не застосовується при зміні вихідного коду програмного забезпечення) програмного продукту.

Документація описує контроль вихідних даних, вхідних послідовностей, параметри програми, обмеження, і всі інші необхідні дані для програмного продукту. Всі повідомлення про помилки визначені і описані. Описані всі коригувальні дії, направлені на виправлення помилок, описаних в повідомленнях про помилку. Документація поширюється на будь-яку частину вбудованого програмного забезпечення, з якими користувач взаємодіє безпосередньо.

Кроки для підтвердження користувацької документації:

* При внесенні змін в користувацьку документацію потрібно інформувати керівника продукту;
* У випадку несхвалення потрібно врахувати зауваження та внести потрібні поправки.

Критерії для підтвердження користувацької документації:

* Достовірність;
* Повнота;
* Несуперечливість;
* Чіткість та лаконічність.

#### 3.2.5. План конфігураційного управління (SCMP)

SCMP документує дії з управління конфігурацією програмного забезпечення (SCM), що повинні бути виконані, а також те, як вони повинні бути виконані, хто відповідає за конкретні завдання, графік заходів, і які ресурси будуть використані. SCMP також визначає методи і засоби, що використовуються для підтримки, збереження, захисту і документування контрольних версій та пов'язані з ним артефакти, виявлені на всіх етапах життєвого циклу програмного забезпечення.

Кроки для підтвердження плану конфігураційного управління:

* При внесенні змін у план перевіряється реалізованість поправок та їх несуперечливість з існуючими положеннями
* У випадку несуперечливості команда повинна обговорити конфліктні положення

Критерії для підтвердження плану конфігураційного управління:

* Реалізованість;
* Несуперечливість.

#### 3.2.6. Інші документи

Керівник проекту, керівник продукту, чи інженер з забезпечення якості можуть включати в документ нову документацію, відносно якої відбуватиметься контроль забезпечення якості.

## 4. Стандарти, практики, конвенцій, і метрики

Даний розділ описує стандарти, практики, угоди і метрики, які використовуються в проекті. Ці матеріали покликані не тільки забезпечити якість проекту, але й отримати кількісні дані про сам процес контролю якості. Отримані дані повинні бути використані розробниками для підвищення рівня самого програмного продукту, а також наступних проектів.

*Стандарти.*

Для ведення документації використовуються стандарти IEEE з відповідними модифікаціями.

*Практики.*

* Оскільки відкладати забезпечення якості недоцільно, розробники застосовують процедури забезпечення якості безпосередньо під час роботи.
* Група вважає, що жоден засіб забезпечення якості, зовнішній по відношенню до конкретного розробника, не може сам по собі забезпечити більш високу якість, ніж це може зробити сам розробник. Інженерам, які ще не мають досвіду застосування такої системи, рекомендується увійти в контакт з наставником з якості.
* Всі артефакти проекту підлягають інспектуванню і всі вони доступні команді після випуску. Це досягається шляхом переміщенням артефактів в систему управління конфігураціями, що забезпечує доступ до вмісту в будь-який час.
* Для всіх процесів має бути проведений огляд можливості поліпшення хоча б раз, і результати цього огляду документуються окремо у письмовій формі.

*Угоди.*

Правила та стиль написання документів повинні відповідати стандартам, якщо це можливо.

*Метрики.*

Для кожного процесу і кожного документа повинні вимірюватися принаймні три показника.

* Час, витрачений розробником, на виконання кожної підзадачі.
* Інтегральний показник якості за шкалою від 1 до 10 на основі самооцінки та оцінки керівника.
* Кількість дефектів на одиницю об'єму (наприклад, на тисячу рядків коду).

**Висновки**

**В ході виконання лабораторної роботи** було визначено набір дій для перевірки процесів забезпечення якості та в результаті сформовано документ по керуванню якістю.