# Лабораторна робота №3.

**Тема: Визначення процедур забезпечення якості.**

**Мета:** отримати практичні навички у обґрунтованому виборі процедур забезпечення якості програмного продукту**.**

**КОНТРОЛЬНІ ПИТАННЯ:**

1. Що таке забезпечення якості програмного забезпечення?

2. Що включає план забезпечення якості програмного забезпечення?

3. Які ви знаєте стандарти забезпечення якості програмного забезпечення?

4. Які методи контролю якості ви знаєте?

5. Визначте поняття якість ПС і рівні моделі якості ПС.

6. Визначте характеристики якості ПС і їхнє призначення.

7. Які методи визначають показники якості?

8. Визначте метрики програмного продукту і їхні складові.

9. Які існують стандарти з якості ПС?

10. Назвіть основні цілі й завдання системи керування якістю.

**Завдання:**

1. Ознайомитися з теоретичною частиною (лекція 8).
2. Визначте для вашого проекту (над яким працюєте на останніх лабораторних та практичних роботах) перелік показників якості.
3. Для кожного етапу ЖЦ ПЗ визначте процедури, за допомогою яких виконується моніторинг виконання цих показників якості (не менше 7 показників).
4. Заповніть таблицю:

|  |  |  |
| --- | --- | --- |
| Етап ЖЦ ПЗ | Показник якості | Процедура перевірки |
|  |  |  |
|  |  |  |

1. Робота повинна бути виконана до ГОСТ 19.106-78 на аркушах формату А4 відповідно до вимог оформлення лабораторних і практичних робіт (титульний лист, номер роботи, мета, завдання, назва власного проекту, його предметна область). При оформленні використовувати MS Office.
2. Оформлену відповідно до встановлених вимог роботу та здати на перевірку викладачеві

По закінченню практичну роботу потрібно здати на перевірку викладачеві, надіславши електронною поштою на адресу [**t.i.lumpova@gmail.com**](mailto:t.i.lumpova@gmail.com). Якщо викладач знаходить помилки чи неточності, він може повернути роботу на доопрацювання.

Файл з роботою повинен мати назву в такому форматі:

**QA<Номер групи><Номер лекції / лабораторної> [літера позначення типу роботи L – лекція, R – лабораторна]<Прізвище англійською>**.. Наприклад, **QA4101R**buts.doc.

Не копіюйте фрагментів з різних інформаційних джерел, подумайте і викладіть свою точку зору. При наявності робіт-"близнюків" відповідь буде зараховуватися першому за часом надсилання.

Тему в заголовку листа записати

**QA<Номер групи>-ПР<Номер практичної>-<Прізвище >**

**Строк виконання цієї роботи ІПЗ-41 – 24.10.2023**

Всі запитання, що виникнуть, надсилайте на електронну адресу викладача, Тему в заголовку листа записати

**QA <Номер групи>-Запитання-<Прізвище >**.

**ТЕОРЕТИЧНА ЧАСТИНА**

**Забезпечення якості програмного забезпечення (SQA -** software quality assurance**)** це процес, який гарантує, що всі процеси, методи, дії та робочі об'єкти програмної інженерії контролюються та відповідають визначеним стандартам. Ці визначені стандарти можуть бути одним або комбінацією будь-яких, таких як ISO 9000, модель CMMI, ISO15504 тощо.

SQA включає всі процеси розробки програмного забезпечення, починаючи від визначення вимог до кодування і закінчуючи випуском. Його головною метою є забезпечення якості.

План забезпечення якості програмного забезпечення (SQAP) включає процедури, техніки та інструменти, які використовуються для забезпечення відповідності продукту чи послуги вимогам, визначеним у специфікації вимог до програмного забезпечення (SRS).

**Заходи з контролю якості**

**1) Створення плану управління SQA:**

Першочергова діяльність включає складання належного плану щодо того, як буде здійснюватися SQA у вашому проекті (який підхід обираєте, які інженерні заходи будуть виконуватися, а також це включає забезпечення того, щоб у вашій команді була правильний набір компетентностей).

**2) Встановлення контрольних точок:**

Команда SQA встановлює різні контрольні точки, на яких вона оцінює якість проектної діяльності на кожному етапі проекту, що забезпечує регулярний контроль якості та роботу відповідно до графіка.

**3) Застосування технічних прийомів програмної інженерії:**

Застосування деяких методів програмної інженерії допомагає розробнику програмного забезпечення досягти високоякісних специфікацій. Для збору інформації дизайнер може використовувати такі методи, як інтерв’ю та технологію системи функціонального аналізу (FAST). Далі на основі зібраної інформації, розробник програмного забезпечення може підготувати оцінку проекту, використовуючи такі методи, як структурування розподілу робіт (WBS), SLOC (вихідний рядок кодів) та оцінка стану контрольних / функціональних точок (FP).

SLOC(англ. Source Lines of Code) — **кількість рядків коду** — метрика програмного забезпечення, що використовується, щоб виміряти розмір комп'ютерної програми, рахуючи число рядків в тексті вихідного коду програми. Дана метрика від початку розроблена для оцінки зусиль, докладених при розробці програмного забезпечення. Однак через те, що одна й та сама функціональність може бути розбита на декілька рядків або записана в один рядок, дана метрика стала неефективною з появою нових мов програмування, у яких в одному рядку можна записати більше однієї команди.

**4) Виконання офіційних технічних оглядів:**

Вони проводяться для оцінки якості та дизайну прототипу проводиться. У цьому процесі проводиться зустріч з технічним персоналом для обговорення фактичних вимог до якості програмного забезпечення та якості дизайну прототипу. Ця діяльність допомагає виявляти помилки на ранній фазі ЖЦ ПЗ і зменшує зусилля з переробки на пізніх фазах.

**5) Наявність стратегії мультитестування:**

Під стратегією мультитестування мається на увазі, що застосовується не якийсь єдиний підхід тестування, а проводяться декілька типів тестування.

**6) Примусове дотримання процесу:**

Ця діяльність наполягає на необхідності дотримання процесу під час процесу розробки програмного забезпечення. Процес розробки також повинен дотримуватися визначених процедур. Ця діяльність являє собою поєднання двох напрямів:

Оцінка продукції, яка підтверджує, що програмний продукт відповідає вимогам, визначеним у плані управління проектом. Це забезпечує належне дотримання встановлених стандартів проекту.

Моніторинг процесу, де перевіряється чи були вжиті правильні кроки під час розробки програмного забезпечення. Це робиться шляхом узгодження фактично вжитих кроків із задокументованими кроками.

**7) Контроль змін:**

У цій діяльності використовується поєднання ручних процедур та автоматизованих інструментів, щоб мати механізм контролю змін. Перевіряючи запити на зміну, оцінюючи характер змін та контролюючи ефект зміни, забезпечується підтримка якості програмного забезпечення на етапах розробки та обслуговування.

**8) Вимірювання впливу зміни:**

Якщо команда з контролю якості повідомляє про будь-який дефект, тоді відповідна команда усуває дефект. Після цього команда контролю якості повинна визначити вплив змін, спричинених цим виправленням дефекту. Їм потрібно протестувати не тільки, якщо зміна усунула дефект, а й чи сумісна ця зміна з усім проектом. Для цього використовується показники якості програмного забезпечення, що дозволяє менеджерам та розробникам спостерігати за діяльністю та пропонованими змінами від початку до кінця ЖЦ ПЗ та ініціювати коригувальні дії, де це потрібно.

**9) Виконання аудиту SQA:**

Аудит SQA перевіряє весь фактичний процес ЖЦ ПЗ з подальшим порівнянням його із встановленим процесом. Він також перевіряє те, що команда, що повідомляється у звітах про стан справ, виконувалась чи ні. Ця діяльність також виявляє будь-які проблеми з недотриманням вимог.

**10) Ведення записів та звітів:**

Дуже важливо зберігати необхідну документацію, що стосується SQA, та ділитися необхідною інформацією з SQA із зацікавленими сторонами. Результати випробувань, результати аудиту, звіти про огляд, документацію із запитами на зміни тощо слід зберігати для подальшого використання.

**11) Управління хорошими стосунками:**

Дуже важливо підтримувати гармонію між контролем якості та командою розробників. Ми часто чуємо, що тестувальники та розробники часто почуваються вищими один від одного. Цього слід уникати, оскільки це може вплинути на загальну якість проекту.

**Елементи забезпечення якості програмного забезпечення (SQA)**

1. Стандарти програмної інженерії
2. Технічні огляди та аудити
3. Тестування програмного забезпечення для контролю якості
4. Збір та аналіз помилок
5. Управління змінами
6. Освітні програми
7. Управління постачальниками
8. Управління безпекою
9. Безпека
10. Управління ризиками

**Методи SQA**

Існує кілька методів оцінки якості. Аудит - головна техніка, яка широко застосовується. Однак у нас є ще кілька важливих прийомів.

**Різні методи SQA включають:**

* **Аудит:** Аудит включає перевірку робочої продукції та пов'язаної з нею інформації, щоб визначити, чи було дотримано набір стандартних процесів чи ні.
* **Рецензування** : Зустріч, на якій програмний продукт розглядається як внутрішніми, так і зовнішніми зацікавленими сторонами з метою отримання їх коментарів та схвалення.
* **Інспекція коду:** Це найбільш офіційний вид огляду, який проводить статичне тестування, щоб знайти помилки та уникнути зростання дефектів на пізніх стадіях. Це робиться підготовленим посередником / однолітком і базується на правилах, контрольному списку, критеріях входу та виходу. Рецензент не повинен бути автором коду.
* **Інспекція проекту:** Перевірка проекту виконується за допомогою контрольного списку, який перевіряє наступні області проектування програмного забезпечення:
  + Загальні вимоги та дизайн
  + Функціональні характеристики та специфікації інтерфейсу
  + Конвенції
  + Простежуваність вимог
  + Структури та інтерфейси
  + Логіка
  + Продуктивність
  + Обробка помилок та відновлення
  + Випробуваність, розтяжність
  + Зчеплення та згуртованість
* **Моделювання:** Моделювання - це інструмент, який моделює реальну ситуацію з метою віртуального вивчення поведінки досліджуваної системи.
* **Функціональне тестування:** Це методика контролю якості, яка перевіряє, що робить система, не враховуючи, як вона працює. Цей тип тестування чорної скриньки в основному фокусується на тестуванні специфікацій системи або функцій.
* **Стандартизація:** Стандартизація відіграє вирішальну роль у забезпеченні якості. Це зменшує двозначність і здогадки, тим самим забезпечуючи якість.
* **Статичний аналіз:** Це програмний аналіз, який виконується автоматизованим інструментом без фактичного запуску програми. Ця техніка широко використовується для забезпечення якості медичного, ядерного та авіаційного програмного забезпечення. Показники програмного забезпечення та зворотна інженерія - деякі популярні форми статичного аналізу.
* **Покрокові інструкції:** Проходження програмного забезпечення або проходження коду - це своєрідна експертна оцінка, коли розробник направляє членів команди розробників до перегляду продукту та підняття запитів, пропонування альтернативних варіантів, коментарів щодо можливих помилок, стандартних порушень чи будь-яких інших проблем.
* **Тестування шляху:** Це техніка тестування білого ящика де повне покриття філій забезпечується виконанням кожного незалежного шляху принаймні один раз.
* **Стрес-тестування:** Цей тип тестування проводиться, щоб перевірити, наскільки міцною є система, перевіряючи її під великим навантаженням, тобто поза нормальними умовами.
* **Шість сигм:** Six Sigma - це підхід до забезпечення якості, який націлений на майже ідеальні товари чи послуги. Він широко застосовується в багатьох областях, включаючи програмне забезпечення. Головною метою шести сигм є вдосконалення процесу, щоб вироблене програмне забезпечення не містило дефектів на 99,76%.

SQA - паралельна діяльність, яка застосовується протягом усього життєвого циклу програмного забезпечення.

Гарантія якості програмного забезпечення дуже важлива для того, щоб ваш програмний продукт або послуга досягли успіху на ринку та виправдали очікування замовника.

Існує різноманітна діяльність, стандарти та методи, яких потрібно дотримуватися, щоб забезпечити якісне програмне забезпечення високої якості та узгоджується з потребами бізнесу.