ЗБПИ20-2 Нечаев Данила экзамен Вариант 24

**1 вопрос**: Объектно-ориентированные метрики. Общие характеристики объектно-ориентированных компонентов

Объектно-ориентированные метрики и характеристики компонентов являются ключевыми аспектами в разработке программного обеспечения, ориентированной на объекты. Они помогают оценивать качество кода, улучшать проектирование и обеспечивать более эффективное управление проектами.

**Общие Характеристики Объектно-Ориентированных Компонентов**

1. **Инкапсуляция**: Скрытие внутренних деталей компонентов программы, что облегчает управление сложностью и повышает безопасность данных.
2. **Наследование**: Позволяет новому классу перенимать (или расширять) функциональность уже существующего класса. Это облегчает повторное использование кода и иерархическую организацию.
3. **Полиморфизм**: Способность методов иметь одно и то же имя, но разную реализацию в разных классах. Это обеспечивает гибкость в обработке различных типов данных.
4. **Связность:** Степень, в которой элементы класса связаны друг с другом. Высокая связность внутри классов желательна, поскольку она указывает на то, что класс выполняет одну задачу.

**Метрики для Оценки Объектно-Ориентированных Компонентов**

1. **Число методов (NOM)**: Количество методов в классе. Индикатор размера и сложности класса.
2. **Глубина наследования (DIT)**: Уровень, на котором класс находится в иерархии наследования. Более высокие значения могут указывать на более сложную архитектуру.
3. **Число детей (NOC)**: Количество подклассов, наследующих от класса. Высокие значения могут указывать на значительное повторное использование кода.
4. **Связность методов в классе**: Показывает, насколько методы класса связаны с данными класса. Низкие значения LCOM указывают на лучшую проектировочную связность в классе.

**2 вопрос:**

Модель Муса основывается на предположении, что количество отказов программного обеспечения со временем уменьшается, а время между отказами увеличивается по мере исправления ошибок.

**Предварительные данные**

1. **Первоначальное количество ошибок в программе** (N) = 100
2. **Выявлено ошибок** = 30
3. **Время испытаний** (T) = 15 часов
4. **Коэффициент сжатия тестов** (C) = 5
5. **Заданная наработка на отказ** = 3 часа
6. **Общее реальное время тестирования** TrealTreal​ = T / C = 15 / 5 = 3 часа
7. **V** = 108
8. **K** = 3 × 10^-7

N = 100 # Первоначальное количество ошибок  
n\_t = 30 # Выявлено ошибок  
T\_real = 15 / 5 # Общее реальное время тестирования в часах  
V = 108 #  
K = 3e-7 # Коэффициент для расчета µ  
  
# Заданная наработка на отказ  
target\_time = 3 # часы  
  
# Расчет µ  
mu = K \* V  
  
# Расчет λ(t) на момент T\_real  
lambda\_t = mu \* (N - n\_t)  
  
# Расчет надежности R(t) для заданной наработка на отказ  
import math  
R\_t = math.exp(-lambda\_t \* target\_time)  
  
print(mu, lambda\_t, R\_t, sep="\n")

**Коэффициент µ (смертности ошибок): 3.24e-05**

**Интенсивность отказов  на момент времени 3 часа**: 0.002268

**Надёжность программы R(t)для заданной наработки на отказ 3 часа**: примерно 0.993 или 99.3%

**3 вопрос:**

**Производительность**

* Метрики: время ответа системы, количество обработанных транзакций за единицу времени, пропускная способность.
* Временные ряды: измерения производительности в разные моменты времени (например, ежедневно, еженедельно) для отслеживания улучшений или ухудшений в ответах системы.

**Надежность**

* Метрики: время до отказа, частота сбоев, скорость исправления ошибок.
* Временные ряды: данные о количестве и серьезности инцидентов со временем, чтобы увидеть, как изменяется надежность продукта.

**Удобство использования**

* Метрики: время обучения новых пользователей, количество запросов поддержки, пользовательские оценки удобства использования.
* Временные ряды: отслеживание этих показателей с течением времени помогает увидеть, насколько продукт становится более или менее удобным для пользователей.

**Функциональность**

* Метрики: количество реализованных функций, степень удовлетворения пользовательских требований.
* Временные ряды: изменения в количестве и качестве функций, предоставляемых продуктом.

**Обновляемость и масштабируемость**

* Метрики: время, необходимое для внедрения изменений или обновлений, способность системы адаптироваться к увеличению объема данных или числа пользователей.
* Временные ряды: измерения, связанные с развертыванием обновлений и изменениями в использовании системы со временем.