ЗБ-ПИ20-2 Шахмурзиева Раяна экзамен B23

**Задание 1**

**Метрики Мартина**, также известные как метрики OO (объектно-ориентированного) проектирования, помогают оценивать качество архитектуры и дизайна программного обеспечения.

Основные метрики

**Метрика абсракции**:

Измеряет соотношение абстрактных и конкретных классов в пакете.

Определяет степень абстрактности пакета, что помогает в оценке гибкости и возможности расширения кода.

**Сцепление** :

Измеряет взаимозависимость между модулями.

Низкое сцепление указывает на то, что модули автономны и могут легко изменяться.

**Сцепление афферентное**

Количество других модулей, зависящих от данного модуля.

Высокое значение может указывать на важность и централизованность модуля.

**Сцепление эфферентное**

Количество модулей, от которых зависит данный модуль.

Высокое значение может указывать на сложность изменения модуля, так как много других модулей будет затронуто.

**Метрика Нестабильности**

Показатель от 0 (полностью стабильный) до 1 (полностью нестабильный).

Нестабильный модуль склонен к частым изменениям, что может быть плохо для устойчивости системы.

**Метрика Абстракции-стабильности**

Измеряет, насколько модуль является абстрактным и стабильным.

Желательно, чтобы стабильные модули были абстрактными (могли расширяться) и нестабильные были конкретными (легко изменяемыми).

Эти метрики помогают в поддержании хорошего дизайна программного обеспечения, минимизируя взаимозависимости и увеличивая гибкость и расширяемость системы.

**Набор метрик Чидамбера и Кемерера**

Метрики Чидамбера и Кемерера используются для оценки качества объектно-ориентированного программного обеспечения. Основные метрики включают:

**Число методов класса :**

Сумма всех методов в классе, возможно с учетом сложности каждого метода.

Высокое значение WMC указывает на сложный и потенциально трудно поддерживаемый класс.

**Глубина наследования:**

Максимальная длина от данного класса до корневого класса.

Большая глубина может указывать на сложность понимания и тестирования класса.

**Число подклассов :**

Количество прямых подклассов данного класса.

Высокое значение может указывать на высокую степень повторного использования, но также может усложнять управление и тестирование.

**Сцепление между объектами :**

Количество других классов, с которыми данный класс взаимодействует.

Высокое значение CBO указывает на высокую связанность, что может усложнять изменения и сопровождение.

**Плотность связей :**

Количество методов, которые могут быть вызваны в ответ на сообщение, отправленное объекту данного класса.

Высокое значение RFC указывает на сложный и трудно тестируемый класс.

**Скрытие данных :**

Измеряет степень, в которой методы класса не используют одни и те же поля.

Высокое значение LCOM указывает на низкую связанность класса, что может означать необходимость его разбиения.

Эти метрики помогают оценивать и улучшать архитектуру и дизайн объектно-ориентированного программного обеспечения, повышая его поддерживаемость и качество.

**Задание 2**

<https://colab.research.google.com/drive/1sGq4dws68hJOP60FvaSY6uA1t4t0Godl?usp=sharing#scrollTo=51d5e1e4>

**Задание 3**

Согласно процессному подходу какие основные характеристики качества выделяют в гибкой методологии разработки ПС? 2. Приведите основные метрики характеристики качества прогресс

В гибкой методологии разработки программного обеспечения (гибкие методологии или Agile) основное внимание уделяется нескольким ключевым характеристикам качества:

**Основные характеристики качества в гибкой методологии:**

**Работающий продукт**

Главное мерило прогресса — работающий продукт, который демонстрирует функциональность и обеспечивает ценность для пользователя.

**Гибкость и адаптивность**

Способность команды быстро реагировать на изменения требований и адаптироваться к новым условиям и потребностям клиента.

**Постоянная интеграция и доставка**

Частое интегрирование и релиз программного обеспечения для обеспечения стабильности и быстрого получения обратной связи.

**Качество кода**

Поддержание высоких стандартов кода через практики код-ревью, парное программирование, автоматизированное тестирование и рефакторинг.

**Сотрудничество с клиентом**

Тесное взаимодействие с клиентом для точного понимания и выполнения его требований.

**Основные метрики для характеристики качества и прогресса:**

**Скорость**

Измеряет объем работы, выполненной за один спринт. Обычно выражается в количестве выполненных сторипоинтов.

**Диаграмма сгорания задач**

Графическое отображение оставшейся работы по сравнению с временем. Помогает следить за прогрессом и предсказывать завершение работы.

**Цикловое время**

Время, необходимое для завершения одной задачи от начала до конца. Помогает оценивать эффективность процессов.

**Время выполнения заказа**

Время от появления задачи до ее завершения. Включает время ожидания и работы. Важная метрика для понимания общей скорости доставки.

**Покрытие кода тестами**

Процент кода, покрытый автоматическими тестами. Чем выше покрытие, тем больше уверенность в качестве и стабильности кода.

**Плотность дефектов**

Количество дефектов на тысячу строк кода (KLOC). Помогает оценивать качество кода и эффективность тестирования.

**Удовлетворенность клиентов**

Оценка удовлетворенности клиентов качеством продукта и выполненными требованиями. Может измеряться через регулярные опросы и отзывы.

Эти характеристики и метрики помогают командам в гибких методологиях обеспечивать высокое качество программного обеспечения и эффективно управлять процессом разработки.