**1. Метрики ПО: Назначение и Применение**

**Определение метрики программного обеспечения**

Метрика ПО — это количественная мера, используемая для оценки различных аспектов программного кода, таких как сложность, качество, производительность и сопровождаемость.

**Ключевые направления применения метрик**

1. **Оценка сложности кода**
   * Помогает выявить трудные для понимания и поддержки участки кода (например, цикломатическая сложность).
2. **Прогнозирование трудозатрат**
   * Используется для оценки времени и ресурсов, необходимых для разработки или доработки функционала (например, метрики размера кода – строки кода, функции).
3. **Контроль качества**
   * Позволяет находить потенциальные уязвимости, ошибки и "код с запахом" (например, высокое количество параметров функции).
4. **Улучшение стиля кода**
   * Обеспечивает соблюдение стандартов кодирования, что повышает читаемость (например, длина строки, именование переменных).

**Пример: Как метрики помогают при ревью кода?**

Метрики позволяют автоматически находить:

* Слишком сложные функции (цикломатическая сложность > 10),
* Длинные методы (> 50 строк),
* Нечитаемые имена переменных (например, x, data1),
* Глубокую вложенность (if/for внутри if/for).

**2. Метрики Сложности**

**Зачем измерять сложность кода?**

Высокая сложность кода приводит к:

* Трудностям в понимании и поддержке,
* Увеличению числа ошибок,
* Затруднениям при тестировании и рефакторинге.

**Дополнительные метрики сложности**

1. **Глубина вложенности управляющих конструкций**
   * Чем больше вложенных if/for/while, тем сложнее код.
2. **Количество параметров функции**
   * Функции с большим числом параметров (например, > 5) сложнее тестировать и использовать.

**3. Метрики Стилистики (Качества Кода)**

**Зачем нужны метрики стилистики?**

Они помогают:

* Улучшить читаемость кода,
* Упростить поддержку и рефакторинг,
* Соблюдать единые стандарты в команде.

**Примеры метрик стилистики**

1. **Длина идентификаторов**
   * Слишком короткие имена (x, tmp) снижают понятность, слишком длинные – усложняют чтение.
2. **Количество комментариев**
   * Слишком мало комментариев → код непонятен, слишком много → "шум".
3. **Длина строки кода**
   * Длинные строки (> 120 символов) ухудшают читаемость.
4. **Глубина вложенности**
   * Вложенность > 3-4 уровней усложняет восприятие логики.
5. **Соблюдение кодстайла**
   * Единые правила именования, отступов, оформления кода.

**Пример: Почему слишком короткие имена переменных — плохо?**

* x, tmp, data не объясняют назначение переменной,
* Усложняют понимание кода для других разработчиков,
* Увеличивают вероятность ошибок при изменении кода.

**Вывод:** Метрики помогают поддерживать баланс между простотой, понятностью и эффективностью кода.