### 1. Определение и цели измерительных методов оценки программ

**Измерительные методы оценки программ** — это систематизированные подходы к количественной оценке характеристик программного обеспечения с использованием специальных метрик. Эти методы позволяют получить объективные данные о различных аспектах кода, таких как его сложность, качество, производительность и сопровождаемость.

**Основные цели применения:**

* Обеспечение объективной оценки качества кода
* Выявление проблемных участков и "запахов кода"
* Контроль технического долга
* Оптимизация процессов разработки и сопровождения
* Сравнение различных версий или компонентов системы
* Принятие обоснованных решений о необходимости рефакторинга

**2. Примеры метрик и их назначение**

**Метрики размера:**

* **Количество строк кода (SLOC)** — оценивает объем программы. Используется для:
  + Планирования работ
  + Оценки сложности проекта
  + Расчетов производительности разработчиков
* **Количество функций/методов** — показывает модульность кода. Помогает:
  + Оценить уровень детализации
  + Выявить избыточно большие модули

**Метрики сложности:**

* **Цикломатическая сложность** (McCabe) — измеряет количество линейно независимых путей в коде. Применяется для:
  + Выявления переусложненных функций
  + Прогнозирования количества тестов
  + Оценки риска появления ошибок

**Метрики качества:**

* **Коэффициент сопровождения (Maintainability Index)** — комплексный показатель, учитывающий:
  + Объем кода
  + Сложность
  + Комментированность
  + Используется для приоритезации рефакторинга
* **Покрытие тестами (Test Coverage)** — процент кода, выполняемого тестами. Позволяет:
  + Оценить надежность кода
  + Выявить непротестированные участки
  + Контролировать качество тестирования

**3. Условия применения и ограничения**

**Типичные случаи применения метрик:**

* Планирование рефакторинга и оптимизации кода
* Контроль качества в процессе разработки
* Сравнение альтернативных реализаций
* Оценка прогресса проекта
* Принятие решений о архитектурных изменениях
* Аудит кода перед выпуском версии

**Основные ограничения:**

1. **Контекстная зависимость:** Одинаковые значения метрик могут иметь разное значение в разных проектах
2. **Ложные срабатывания:** Автоматические инструменты могут выдавать некорректные предупреждения
3. **Неполнота:** Отдельные метрики не дают полной картины качества кода
4. **Субъективность интерпретации:** Требуется опыт для правильной трактовки результатов
5. **Ресурсоемкость:** Некоторые виды анализа требуют значительных вычислительных мощностей
6. **Динамические аспекты:** Статические метрики не учитывают поведение во время выполнения

**Рекомендации по применению:**

* Использовать комплекс метрик, а не отдельные показатели
* Учитывать специфику проекта и доменной области
* Сочетать количественные метрики с качественным анализом
* Регулярно пересматривать набор используемых метрик
* Адаптировать пороговые значения под конкретный проект