**ПРИДНЕСТРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**им. Т.Г.ШЕВЧЕНКО**

**БЕНДЕРСКИЙ ПОЛИТЕХНИЧЕСКИЙ ФИЛИАЛ**

**Практическая работа №8**

по УП 03.01

Тема: «Использование метрик программного продукта»

Выполнил:

студент 3 курса, БК22АР52ИС2 группы,

**«**Информационные системы и программирования**»**

Николаев Валерий Евгеньевич

Проверил:

Лозовский А.В

2025 год.

**Использование метрик программного продукта** помогает оценивать качество, производительность и эффективность разработки. Они применяются для анализа кода, управления проектом и принятия решений.

**Основные категории метрик**

**1. Метрики кода**

Оценивают качество и сложность исходного кода:

Размер кода:

Количество строк кода (SLOC – Source Lines of Code)

Количество модулей/классов/функций

Сложность кода:

Цикломатическая сложность (McCabe) – число независимых путей в программе

Глубина наследования (DIT – Depth of Inheritance Tree)

Связность классов (Coupling Between Objects, CBO)

Качество кода:

Количество ошибок на 1000 строк кода (Defect Density)

Дублирование кода (Code Duplication)

Соответствие стандартам (через статический анализ, например, SonarQube)

**2. Метрики процесса разработки**

Оценивают эффективность разработки и управления проектом:

Скорость разработки (Velocity) – количество задач/стори-поинтов, выполненных за спринт (в Agile)

Время выполнения задач (Lead Time, Cycle Time)

Частота деплоя (Deployment Frequency)

Коэффициент ошибок (Bug Rate) – количество багов на единицу времени

**3. Метрики производительности**

Измеряют эффективность работы приложения:

Время отклика (Response Time)

Пропускная способность (Throughput) – запросов в секунду

Использование ресурсов (CPU, RAM, Disk I/O)

**4. Метрики тестирования**

Оценивают покрытие и качество тестов:

Покрытие кода (Code Coverage) – процент кода, выполняемого тестами

Количество пройденных/проваленных тестов

Эффективность тестов (Defect Escape Rate) – сколько багов нашлось после релиза

**5. Метрики пользовательского опыта (UX)**

Время загрузки страницы

Количество ошибок в UI/UX

Коэффициент оттока пользователей (Churn Rate)

**Как применять метрики?**

1. Определить цели (улучшение качества, ускорение разработки, снижение ошибок).
2. Выбрать ключевые метрики (например, цикломатическая сложность + покрытие тестами).
3. Собирать данные автоматически (CI/CD, статический анализ, мониторинг).
4. Анализировать и принимать решения (рефакторинг, улучшение процессов).

**Популярные инструменты для сбора метрик**

Анализ кода: SonarQube, ESLint, Checkstyle

Производительность: Prometheus, Grafana, New Relic

Тестирование: JaCoCo (покрытие кода), Selenium

Процесс разработки: Jira, Git (анализ коммитов), GitHub Actions

**Вывод**

Метрики помогают объективно оценивать качество ПО, находить узкие места и улучшать процессы. Главное – выбирать показатели, которые действительно полезны для проекта, а не собирать данные ради данных.