**ПРИДНЕСТРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**им. Т.Г.ШЕВЧЕНКО**

**БЕНДЕРСКИЙ ПОЛИТЕХНИЧЕСКИЙ ФИЛИАЛ**

**Кафедра «Промышленность и информационные технологии»**

**Использование метрик программного продукта**

Выполнил:

студент 3 курса, БК22АР52ИС2 группы,

дневной формы обучения, специальность:

«Информационные системы и программирование»

Герасименко Денис Андреевич

Руководитель учебной практики

Лазовский А.В.

Бендеры 2025 год.

**1. Введение в метрики программного продукта**

Метрики программного продукта — это количественные показатели, которые используются для оценки качества, производительности, стабильности и других характеристик программного обеспечения. Эти метрики играют важную роль на протяжении всего жизненного цикла разработки ПО, начиная от планирования и проектирования до тестирования, внедрения и поддержания.

Использование метрик позволяет:

* Оценивать качество программного продукта.
* Повышать эффективность разработки.
* Улучшать процессы тестирования и поддержки.
* Оценивать риски и улучшать принятие решений на всех этапах разработки.

**2. Виды метрик программного продукта**

Метрики можно классифицировать по различным аспектам разработки ПО. Рассмотрим несколько ключевых категорий:

**2.1. Метрики качества кода**Эти метрики используются для оценки чистоты, читаемости и поддерживаемости исходного кода.

* Плотность комментариев: отношение количества комментариев в коде к общему количеству строк. Важная метрика для оценки понятности и документации кода.
* Сложность кода: показатель, который измеряет сложность логики программы. Например, цикломатическая сложность (цикломатическая сложность) определяет количество независимых путей в программе. Чем выше этот показатель, тем сложнее тестировать и поддерживать код.
* Глубина вложенности: максимальное количество вложенных уровней в блоках кода. Чем больше уровней вложенности, тем сложнее понимать и поддерживать код.
* Коэффициент повторяемости: показывает, сколько раз в программе повторяются одни и те же фрагменты кода. Высокий коэффициент повторяемости указывает на потенциальную избыточность, что может привести к ошибкам и затруднить сопровождение.

**2.2. Метрики производительности**

Эти метрики связаны с эффективностью работы программы, включая время отклика, использование ресурсов и масштабируемость.

* Время отклика: время, которое требуется системе для выполнения операции, от начала запроса до получения результата.
* Пропускная способность: количество операций или запросов, которые система может обработать за определённый период времени.
* Использование ресурсов: метрики, которые отслеживают, как программа использует системные ресурсы, такие как процессорное время, память и дисковое пространство.

**2.3. Метрики надежности и стабильности**

Оценка того, насколько программный продукт устойчив к сбоям и как быстро он восстанавливается.

* Частота ошибок: количество дефектов или ошибок, обнаруженных в процессе эксплуатации программного продукта, в расчёте на время или количество операций.
* Среднее время до отказа (MTBF — Mean Time Between Failures): средний интервал времени между сбоями или отказами системы.
* Среднее время восстановления (MTTR — Mean Time to Repair): среднее время, необходимое для восстановления системы после отказа.

**2.4. Метрики тестирования**

Используются для оценки эффективности тестирования и качества обнаружения дефектов.

* Покрытие кода: процент кода, который был протестирован в процессе выполнения тестов. Чем выше покрытие, тем больше частей программы проверено.
* Частота дефектов: количество найденных ошибок в тестах на единицу времени или по отношению к общему количеству функций.
* Плотность дефектов: отношение количества дефектов к количеству строк кода или к числу модулей, что позволяет оценить качество программного продукта на разных уровнях.

**2.5. Метрики поддерживаемости**

Оценка того, насколько легко поддерживать и обновлять программное обеспечение.

* Время на исправление ошибок: сколько времени требуется для поиска и исправления дефектов в продукте.
* Частота обновлений: количество обновлений, исправлений или изменений, которые были внесены в программу в определённый период.
* Простота изменений: мера того, насколько легко добавить новые функции или изменить существующие, не нарушая работоспособности системы.

**3. Применение метрик программного продукта**

Метрики программного продукта широко используются на всех этапах разработки ПО, включая:

* Планирование и проектирование: Метрики помогают разработчикам и менеджерам понять, как сложен проект, какие ресурсы могут потребоваться и как можно улучшить процесс разработки.
* Разработка: Во время написания кода метрики позволяют отслеживать качество кода и вовремя устранять проблемы, которые могут повлиять на дальнейшую работу.
* Тестирование: Метрики тестирования позволяют оценить эффективность тестовых сценариев и стратегии, а также степень покрытия кода.
* Эксплуатация и поддержка: Метрики, связанные с производительностью, надёжностью и стабильностью, позволяют мониторить работу системы в реальных условиях и выявлять слабые места.

**4. Преимущества и недостатки использования метрик**

Преимущества:

* Управление качеством: Метрики позволяют объективно оценить качество программного продукта, помогают выявить и устранить проблемы на ранних стадиях разработки.
* Оценка рисков: Метрики помогают выявлять потенциальные риски, такие как сложности в коде или проблемы с производительностью, которые могут привести к сбоям в будущем.
* Оптимизация процессов: Использование метрик позволяет повысить эффективность разработки, оптимизировать ресурсы и время, а также улучшить координацию между командами.

Недостатки:

* Перегрузка метриками: Использование слишком большого числа метрик может привести к излишней сложности в анализе и затруднить принятие решений.
* Риск фокусирования на числах: Слишком сильное внимание к числовым показателям может привести к игнорированию других важных аспектов, таких как творческий подход или опыт команды.

**5. Заключение**

Метрики программного продукта играют важную роль в процессе разработки и поддержания ПО. Они помогают не только контролировать качество и производительность системы, но и улучшать процессы разработки, тестирования и эксплуатации. Однако важно использовать метрики сбалансированно, чтобы не перегрузить проект лишними показателями и сохранить фокус на достижении конечных целей.