**Метрики для измерения эффективности разрабатываемого программного продукта**

Этапы формирования программы измерений:

1. Идентификация целей.

Цели:

1. Повысить производительность труда, каждого члена команды в частности.
2. Повысить качество разрабатываемой версии продукта каждым программистом.

Вопросы:

1. Сколько было суммарно (всей командой) создано строк кода?

Сколько программистов в команде?

Сколько времени работал каждый программист?

1. Сколько было обнаружено ошибок до выпуска последней версии программного продукта?

Сколько было суммарно (всей командой) создано строк кода для текущей версии?

Сколько программистов в команде работало над текущей версией продукта?

Выбранные метрики:

1. В качестве метрики для оценки средней производительности программиста в команде была выбрана модифицированная метрика производительности труда.

Productivity by Person = (LOC / Рабочее время затраченное на проект)/Количество программистов в команде

Стратегическая цель метрики – Повысить производительность труда, каждого члена команды.

Изучаемый объект метрики – проект, измеряемый атрибут – средняя производительность труда программиста в команде.

Единица измерения – (LOC/человек\*час).

1. В качестве метрики оценки среднего качества разрабатываемого продукта каждым программистом была выбрана модифицированная метрика плотности ошибок в продукте.

In Process Faults by Person = ((Число обнаруженных ошибок до выпуска релиза/Количество программистов работавших над версией.)/LOC)

Стратегическая цель метрики – Повысить качество разрабатываемой версии продукта каждым программистом.

Изучаемый объект метрики – проект, измеряемый атрибут – средняя плотность неполадок на программиста.

Единица измерения – неполадка/LOC\*человека.

1. Планирование процесса измерений

В качестве измерения показателя LOC и количества выявленных ошибок (countOfEr) будет использоваться встроенные возможности сервиса github.com.

Измерения времени работы программиста будет производится путём измерения времени, потраченного на выполнение задач, поставленных перед началом проекта и измеряется в часах.

Количество программистов (countOfP) в команде неизменно и равно 3.

Версия продукта на данный момент одна поэтому количество программистов, работающих над текущей версией продукта 3.

1. Выполнение измерений

LOC = 113916.

countOfEr = 12.

time = 240. Часов.

countOfP = 3.

1. Анализ и интерпретация данных
   1. Productivity by Person

Получаем что в среднем каждый из программистов в нашей команде писал 158 строк кода в час в течении своей работы.

Исходя из полученных результатов можно сделать вывод: во время разработки следующей версии ПО можем ставить задачу перед каждым программистом:

1. Для не уменьшения производительности – писать не менее 160 строк кода в час.
2. Для увеличения производительности – писать более 180 строк кода в час.
   1. In Process Faults by Person

Получаем, что в среднем допускается 4 ошибки каждым программистом на 10 тысяч строк кода.

Исходя из полученных результатов можно сделать вывод: во время разработки следующей версии ПО можем ставить требования перед каждым программистом:

1. Для не увеличения плотности ошибок в конечно продукте – не допускать более 1 ошибки на 2500 строк кода.
2. Для уменьшения плотности ошибок в конечном продукте – не допускать более 1 ошибки на 5000 строк кода.