Болычев Леонид, Бондаренко Денис, Румянцев Никита,

Б9119-02.03.03техпро,4 курс,

02.03.03 Математическое обеспечение

и администрирование информационных систем

**Разработка программы измерений проекта**

Было предложено разработать систему метрик для оценки эффективности процесса разработки и качества программного продукта, создаваемого в проекте. Опираясь примеры используемых в других проектах метриках, услышанных нами на лекциях, мы решили разработать свои метрики.

**Метрики для оценки эффективности процесса разработки:**

1. *эффективность = storypoints / time*,

где *storypoints* – очки, начисляющиеся за выполнение задач,

*time* – время (в днях), затраченное на выполнение задач.

Её стратегическая цель – попытаться оптимизировать время, уходящее на разработку задач.

Эта метрика кажется нам наиболее эффективной за счёт её простоты. Чем больше сторипоинтов в день зарабатывает участник проекта, тем он эффективнее. Единственная проблема – данная метрика подразумевает необходимость разбить весь проект на огромное число маленьких задач, и каждой назначить на глазок число сторипоинтов за её решение.

1. **On project % time (OPPT)**

*OPPT = (затраченное на проект рабочее время / Общее рабочее время) \* 100%*,

где *OPPT* – сама метрика,

*затраченное на проект рабочее время* – это время, в течение которого работа шла только над самим проектом,

*общее рабочее время* – суммарное время работы.

Стратегическая цель метрики – сокращение сроков выполнения проектов по разработке ПО.

Эта метрика кажется нам менее предпочтительной, чем предыдущая, но она позволит оптимизировать затрачиваемое время, чтобы больше его уходило на работу над проектом и меньше пропадало вникуда.

Метрики для оценки качества программного продукта:

1. **In Process Faults (плотность ошибок в продукте)**

*IPF = (Число обнаруженных ошибок в проекте до его релиза) / LOC*,

где *IPF* – сама метрика In Process Faults,

*число обнаруженных ошибок в проекте до его релиза*– суммарное количество ошибок, выясненных на этапе тестирования и на этапе работы над кодом,

*LOC* – число строк кода.

Её стратегическая цель – повысить качество ПО путём выяснения, насколько много в коде ошибок относительно суммарного объёма написанного кода.

Эта метрика позволит нам примерно оценить, насколько качественно мы выполнили работу, на основе числа найденных до релиза ошибок. Возможно, она не слишком глубока, но она универсальна и проста в использовании.

1. **Product Fault Density (PFD)**

*PFD-Req = Число ошибок, допущенных на этапе разработки требований / Размер требований;*

*PFD-Design = Число ошибок, допущенных на этапе дизайна / Размер документов дизайна;*

*PFD-Coding = Число ошибок, допущенных на этапе кодирования / LOC;*

*PFD-Test = Число ошибок, допущенных на этапе тестирования / Количество тестов;*

Стратегическая цель метрики – повысить качество разрабатываемого ПО. Изучаемый объект метрики – продукт, измеряемый атрибут – плотность неполадок (учитывает все ошибки). Единица измерения – неполадка / единица размера.

Эта метрика даст возможность оценить качество на основе числа ошибок с учётом того, на каком этапе было допущено сколько ошибок. Это может быть особенно полезно, так как ошибки на ранних этапах опаснее и намного дороже.