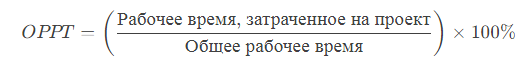
**Метрики для оценки эффективности процесса разработки**

**1. Процент времени, затраченного на проект (On Project % Time, OPPT)**

* **Описание:** Метрика измеряет, какая часть рабочего времени команды тратится непосредственно на разработку проекта.
* **Формула:**

**Пример для игры-платформера:**

* + Если команда из 5 человек работает над проектом 40 часов в неделю, и 30 часов из них тратится на разработку игры, то OPPT = 75%.
* **Цель:** Увеличение процента времени, затрачиваемого на проект, что свидетельствует о повышении эффективности работы команды.

**2. Производительность труда (Productivity)**

* **Описание:** Метрика измеряет, сколько строк кода (LOC) создается за единицу времени.
* **Формула:**



* **Пример для игры-платформера:**
  + Если за неделю команда написала 1000 строк кода, затратив 40 человеко-часов, то Productivity = 25 LOC/человеко-час.
* **Цель:** Повышение производительности труда, что позволяет быстрее завершать задачи.

**Метрики для оценки качества программного продукта**

**1. Плотность ошибок в продукте (In Process Faults, IPF)**

* **Описание:** Метрика измеряет количество ошибок, обнаруженных до выпуска релиза, на 1000 строк кода.
* **Формула:**



* **Пример для игры-платформера:**
  + Если в игре обнаружено 50 ошибок до релиза, а общий объем кода составляет 10 000 строк, то IPF = 5 ошибок на 1000 строк кода.
* **Цель:** Снижение плотности ошибок, что свидетельствует о повышении качества кода.

**2. Эффективность обнаружения ошибок (Phase Containment Effectiveness, PCE)**

* **Описание:** Метрика измеряет, насколько эффективно ошибки обнаруживаются на ранних этапах разработки.
* **Формула:**



**Пример для игры-платформера:**

* + Если на этапе тестирования обнаружено 80 ошибок из 100, внесенных на этапе кодирования, то PCE = 80%.
* **Цель:** Увеличение эффективности обнаружения ошибок на ранних этапах, что снижает затраты на их исправление.