\МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Гжельский государственный университет»** (ГГУ)

Колледж ГГУ

Специальность 09.02.07 Информационные системы и программирование

**Реферат**

**По предмету «Управление проектами»**

**На тему «Метрики сложности программного кода»**

ВЫПОЛНИЛ:

Студент группы ИСП-0-17

Филипович А.А.

ПРОВЕРИЛА:

Прокуронова А.Ю.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

п. Электроизолятор

2019 г.

## Для чего нужны метрики?

## Применение метрик позволяет руководителям проектов и предприятий изучить сложность разработанного или даже разрабатываемого проекта, оценить объем работ, стилистику разрабатываемой программы и усилия, потраченные каждым разработчиком для реализации того или иного решения. Однако метрики могут служить лишь рекомендательными характеристиками, ими нельзя полностью руководствоваться.

## Метрики сложности:

## 1) метрики размера ПО;

## 2) метрики сложности потока управления ПО;

## 3) метрики сложности потока данных ПО

## Виды метрик:

## 1. Количественные метрики - Данная метрика была изначально разработана для оценки трудозатрат по проекту

## 2. Метрики сложности потока управления программы - Следующий большой класс метрик, основанный уже не на количественных показателях, а на анализе управляющего графа программы, называется метрики сложности потока управления программ.

## 3.Метрики сложности потока управления данными - суть метода состоит в оценке информационной прочности отдельно взятого программного модуля с помощью анализа характера использования переменных из списка ввода-вывода.

## 4. Метрики сложности потока управления и данных программы - Данный класс метрик устанавливает сложность структуры программы как на основе количественных подсчетов, так и на основе анализа управляющих структур.

## 5. Объектно-ориентированные метрики

## 6. Метрики надежности - метрики, близкие к количественным, но основанные на количестве ошибок и дефектов в программе.

## 7. Гибридные метрики - В завершении необходимо упомянуть еще один класс метрик, называемых гибридными. Метрики данного класса основываются на более простых метриках и представляют собой их взвешенную сумму.

## Также к группе метрик, основанных на подсчете некоторых единиц в коде программы, относят метрики Холстеда [[3](http://www.viva64.com/go.php?url=243)]. Данные метрики основаны на следующих показателях: n1 — число уникальных операторов программы, включая символы- разделители, имена процедур и знаки операций (словарь операторов), n2 — число уникальных операндов программы (словарь операндов), N1 — общее число операторов в программе, N2 — общее число операндов в программе, n1' — теоретическое число уникальных операторов, n2' — теоретическое число уникальных операндов. Учитывая введенные обозначения, можно определить: n=n1+n2 — словарь программы. N=N1+N2 — длина программы. n'=n1'+n2' — теоретический словарь программы. N'= n1\*log2(n1) + n2\*log2(n2) — теоретическая длина программы (для стилистически корректных программ отклонение N от N' не превышает 10%) V=N\*log2n — объем программы. V'=N'\*log2n' — теоретический объем программы, где n\* — теоретический словарь программы. L=V'/V — уровень качества программирования, для идеальной программы L=1. L'= (2 n2)/ (n1\*N2) — уровень качества программирования, основанный лишь на параметрах реальной программы без учета теоретических параметров. EC=V/(L')2 — сложность понимания программы. D=1/ L' — трудоемкость кодирования программы. y' = V/ D2 — уровень языка выражения. I=V/D — информационное содержание программы, данная характеристика позволяет определить умственные затраты на создание программы E=N' \* log2(n/L) — оценка необходимых интеллектуальных усилий при разработке программы, характеризующая число требуемых элементарных решений при написании программы. При применении метрик Холстеда частично компенсируются недостатки, связанные с возможностью записи одной и той же функциональности разным количеством строк и операторов. Еще одним типом метрик ПО, относящихся к количественным, являются метрики Джилба. Они показывают сложность программного обеспечения на основе насыщенности программы условными операторами или операторами цикла. Данная метрика, не смотря на свою простоту, довольно хорошо отражает сложность написания и понимания программы, а при добавлении такого показателя, как максимальный уровень вложенности условных и циклических операторов, эффективность данной метрики значительно возрастает.

## Заключение

## Подводя итог, хотелось бы отметить, что ни одной универсальной метрики не существует. Любые контролируемые метрические характеристики программы должны контролироваться либо в зависимости друг от друга, либо в зависимости от конкретной задачи, кроме того, можно применять гибридные меры, однако они так же зависят от более простых метрик и также не могут быть универсальными. Строго говоря, любая метрика — это лишь показатель, который сильно зависит от языка и стиля программирования, поэтому ни одну меру нельзя возводить в абсолют и принимать какие-либо решения, основываясь только на ней.