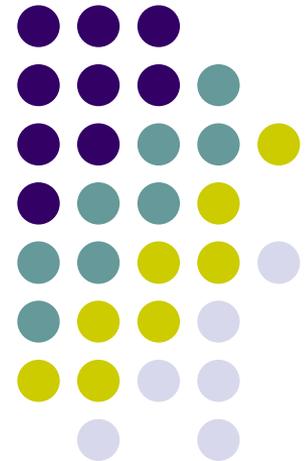


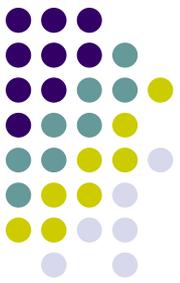
# Автоматизация сбора и анализ метрик программного кода для языка PHP

Научный руководитель: Васючкова Т.С.

Выполнила: Яковлева А.А



# Метрики программного кода.



- Сокращение затрат на оценку качества создаваемого ПО в компании путем автоматизации процесса оценки.
- Оперативный контроль за качеством создаваемого ПО с целью выявления и исправления ошибок и недоработок.
- Мониторинг и планирование профессионального роста сотрудника компании путем регулярной оценки результатов его труда.

# Цель и задачи.



## Цель:

- Автоматизация сбора и анализ метрик программного кода для языка PHP.

## Задачи:

- Разработка концепции методики;
- Анализ потребностей ИТ компании, ожиданий от применения методики оценки качества;
- Определение требований к методике;
- Анализ существующих средств для автоматизации сбора и анализа метрик программного кода;
- Отбор метрик подходящих под требования компании;
- Разработка программного модуля для сбора оценочных метрик разного типа;
- Создание новых метрик целесообразных для использования компанией;
- Создание инструментария для анализа полученных значений метрик.



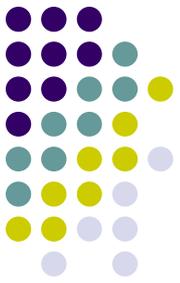
# Выбранные метрики.

- Метрики Холстеда
- Метрики строк кода
- Метрики Джилба
- Метрика Мак-Кейба
- Комплексный показатель качества кода

# Метрики Холстеда



- Число уникальных операторов и операндов программы
- Общее число операторов и операндов
- Теоретическое число уникальных операторов и операндов
- Длина программы
- Размер словаря
- Объем программы
- Уровень сложности
- Усилия для реализации
- Время для понимания программы
- Количество выявленных ошибок.



# Метрики Джилба

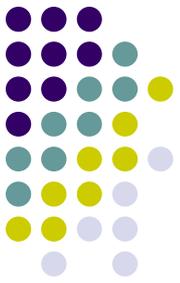
- Показывают сложность программного обеспечения на основе насыщенности программы условными операторами или операторами цикла.
- CL – абсолютная сложность (количество управляющих операторов).
- Относительная сложность программы по Джилбу:
- $cl = CL/n$ ,
- где  $n$  – общее число операторов программы.



# Метрика Мак-Кейба

- Цикломатическая сложность вычисляется для графа потока управления процедуры или функции по формуле:
- $V(G) = e - n + 2p$ ,
- где
- $e$  - количество дуг,
- $n$  - количество вершин,
- $p$  - число компонент связности графа потока управления.

# Комплексный показатель качества кода.



- $MI_{woc} = 171 - 5.2 * \ln(aveV) - 0.23 * aveG - 16.2 * \ln(aveLOC),$

где

$aveV$  – средний объем программы, вычисленный по Холстеду;

$aveG$  – средняя расширенная цикломатическая сложность;

$aveLOC$  – количество строк кода.

- $MI_{cw} = 50 * \sin(\sqrt{2.4 * perCM}),$

где

$perCM$  – средний процент строк комментариев по модулю.

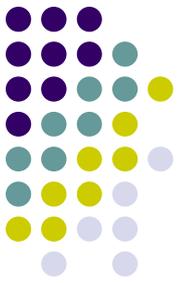
- $MI = MI_{woc} + MI_{cw}$

# Реализация новых метрик.

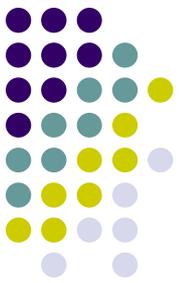


- Метрика эффективность работы конкретного программиста, определяемая по его комплексному показателю качества кода по всем выполняемым им проектам.
- Метрика стоимости поддержки проекта в человеко-часах в месяц.

# Метрика стоимости поддержки проекта (ТСО).



- TCO = training time + maintenance time,
- где
- $TT = K1 * T$ ,
- $MT = B * \text{кол-во файлов} * T$  .
- K1 — сколько раз в месяц файлы проекта подвергаются доработке.
- $K2=0.85$ .
- $1/K2$  (training time + maintenance time) = оценку стоимости поддержки и развития проекта в человеко-часах в месяц.



# Программный модуль.

- В основе работы программного средства лежит парсер, производящий разбор кода на его семантические единицы.
- Программа фиксирует операторы и операнды, условные операторы и т. д., а также запоминает их местонахождение в коде.



```
# PHP Source Code Analyzer version dev
# Run php-sca --help for more info
```

```
File src/Alawar/PHPSourceCodeAnalyzer/Logic/Metrics/ProgramEffort.php : 0-32
```

```
N2 = 49
n2 = 18
N1 = 34
n1 = 16
lines-of-code = 27
lines-of-comments = 4
vG = 2
N = 83
V = 422.25941582378
D = 21.777777777778
L = 0.045918367346939
SAL = 0.045918367346939
E = 9195.8717223845
T = 510.88176235469
PE = 575.18642709348
lines-of-comments-percent = 12
B = 0.1463100015523
MIwoc = 85.710219526771
MI = 52.804238388764
```

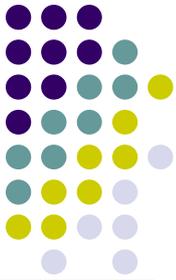
```
Class ProgramEffort : 8-31
```

```
N2 = 46
n2 = 15
N1 = 29
n1 = 12
lines-of-code = 24
lines-of-comments = 4
vG = 2
N = 75
V = 356.61656266226
D = 18.4
L = 0.054347826086957
SAL = 0.054347826086957
E = 6561.7447529856
T = 364.54137516587
PE = 547.2120627488
```

# Инструментарий для анализа полученных значений метрик.



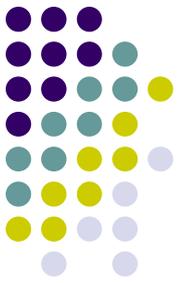
- Строится на основе граничных значений метрик, полученных опытным путем, и указывает, что именно нуждается в доработке программистом для приведения тех или иных метрических показателей в рекомендуемые пределы.



```
# PHP Source Code Analyzer version dev
# Run php-sca --help for more info

src/Alawar/PHPSourceCodeAnalyzer/Console/Formatter/MetricsFormatter.php:0-47 lines-of-comments-percent The file has 8% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Console/Formatter/ViolationsFormatter.php:0-73 lines-of-comments-percent The file has 0% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Console/Command.php:0-161 lines-of-comments-percent The file has 6% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Console/Application.php:0-49 lines-of-comments-percent The file has 0% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Metrics/EffortToImplement.php:0-23 lines-of-comments-percent The file has 18% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Metrics/DifficultyLevel.php:0-33 lines-of-comments-percent The file has 12% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Metrics/ProgramLength.php:0-23 lines-of-comments-percent The file has 18% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Metrics/MaintainabilityIndex.php:0-23 lines-of-comments-percent The file has 18% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Metrics/MaintainabilityIndexWithoutComments.php:0-31 lines-of-comments-percent The file has 12% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Metrics/TimeToImplement.php:0-23 lines-of-comments-percent The file has 18% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Metrics/MaintainabilityIndexCommentWeight.php:0-38 lines-of-comments-percent The file has 10% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Metrics/ProgramVolume.php:0-26 lines-of-comments-percent The file has 16% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Metrics/Metrics.php:0-55 lines-of-comments-percent The file has 0% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Metrics/ProgramLevel.php:0-23 lines-of-comments-percent The file has 18% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Metrics/ProgramEffort.php:0-32 lines-of-comments-percent The file has 12% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Metrics/NumberOfDeliveredBugs.php:0-23 lines-of-comments-percent The file has 18% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Metrics/SpecificationAbstractionLevel.php:0-23 lines-of-comments-percent The file has 18% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Metrics/LinesOfCommentsPercent.php:0-33 lines-of-comments-percent The file has 12% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Violations/Violations.php:0-76 lines-of-comments-percent The file has 4% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Violations/Thresholds.php:0-96 lines-of-comments-percent The file has 0% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Counters/OperandCounter.php:0-62 lines-of-comments-percent The file has 10% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Counters/McCabeCounter.php:0-47 lines-of-comments-percent The file has 14% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Counters/OperandUniqueCounter.php:0-35 lines-of-comments-percent The file has 18% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Counters/OperatorCounter.php:0-74 lines-of-comments-percent The file has 9% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Counters/OperatorUniqueCounter.php:0-34 lines-of-comments-percent The file has 19% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Counters/DummyCounter.php:0-41 lines-of-comments-percent The file has 16% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Counters/FalseCounter.php:0-29 lines-of-comments-percent The file has 22% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Counters/LinesOfCodeCounter.php:0-59 lines-of-comments-percent The file has 11% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Counters/LinesOfCommentsCounter.php:0-50 lines-of-comments-percent The file has 18% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Stats/Stats.php:0-200 lines-of-comments-percent The file has 1% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Logic/Stats/Stats.php:139-174 V The method result() has ProgramVolume of 1003.3694238135. The configured threshold is 1000.
src/Alawar/PHPSourceCodeAnalyzer/Services/Misc/Iterators/SourceFilesIterator.php:0-79 lines-of-comments-percent The file has 21% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Services/Misc/Iterators/FilterIterator.php:0-27 lines-of-comments-percent The file has 20% of comments and is either very trivial or poorly explained.
src/Alawar/PHPSourceCodeAnalyzer/Services/Misc/FileLines/IFileLines.php:11-11 lines-of-code The method lines() has 1 lines of code and too small.
src/Alawar/PHPSourceCodeAnalyzer/Services/Misc/FileLines/FileLines.php:0-28 lines-of-comments-percent The file has 0% of comments and is either very trivial or poorly explained.
```

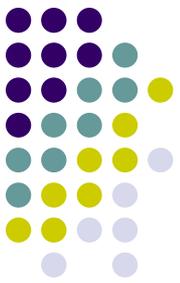
# Результаты.



- В ходе работы выполнена поставленная цель – автоматизация сбора и анализ метрик программного кода для языка PHP.

Были выполнены поставленные задачи:

- Осуществлен анализ потребностей ИТ компании, ожиданий от применения методики оценки качества;
- Определены требования к методике оценки качества;
- Выполнен анализ существующих средств для автоматизации сбора и анализа метрик программного кода;
- Проведен выбор метрик подходящих под требования компании;
- Разработан программный модуль для сбора оценочных метрик разного типа;
- Созданы новые метрики целесообразные для использования компанией;
- Создан инструментарий для анализа полученных значений метрик.



# Публикации

- Автоматизация сбора и анализ метрик программного кода для языка PHP (статья)  
– Вестник магистратуры, научный журнал  
– 2014 г.
- Автоматизация сбора и анализ метрик программного кода для языка PHP (тезисы)  
– МНСК (выступление) – 2014 г.

# Спасибо за внимание

---

