NoVerify: Pretty fast PHP linter

Искандер Шарипов
@quasilyte

PHP Russia 2019

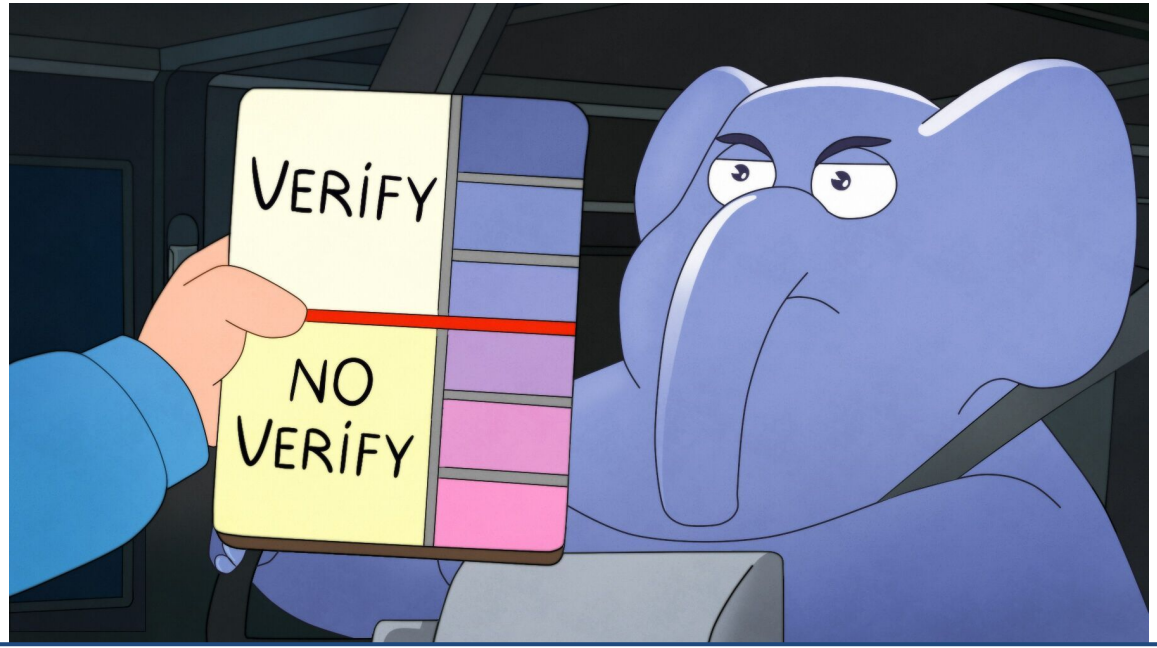
Профессиональная конференция для php-разработчиков

О себе

- Go compiler @ Intel
- Backend infrastructure @ VK
- Go сообщества в Казани и НН
- PHP - мой любимый ЯП*

* как предмет статического анализа

@quasilyte



Статический анализатор NoVerify

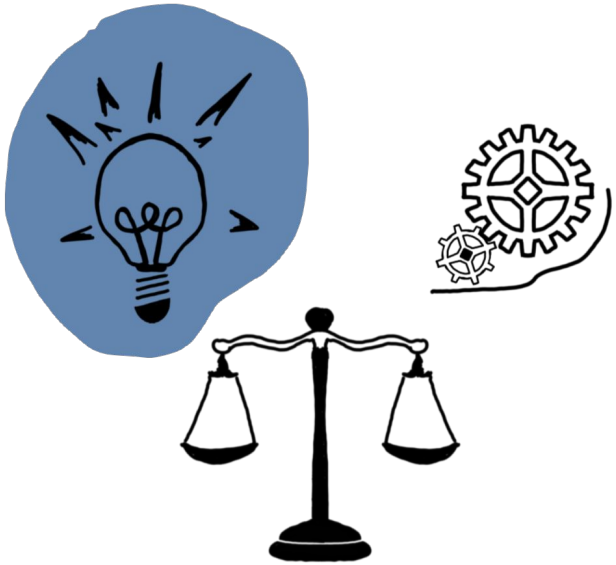
Предыстория

Статья от автора NoVerify на хабре

https://habr.com/ru/company/vk/blog/442284/

Основные темы

- Идея и реализация
- Компромиссы интеграции
- Технические детали



Идея и реализация

Продуктовая разработка

Продуктовая разработка

- Для ВКонтакте важна скорость разработки

Продуктовая разработка

- Для ВКонтакте важна скорость разработки
- Быстрокод может содержать ошибки

Продуктовая разработка

- Для ВКонтакте важна скорость разработки
- Быстрокод может содержать ошибки
- Из-за ошибок страдают наши пользователи

Продуктовая разработка

- Для ВКонтакте важна скорость разработки
- Быстрокод может содержать ошибки
- Из-за ошибок страдают наши пользователи

Контроль качества замедляет разработку.

Этот эффект нужно минимизировать.

Контроль качества

- Больше ревью кода
- Больше тестировщиков и/или тестов

Контроль качества

- Больше ревью кода
- Больше тестировщиков и/или тестов

Контроль качества

- Больше ревью кода
- Больше тестировщиков и/или тестов
- Сбор метрик, быстрая починка
- Более строгий пайплайн

Контроль качества

- Более строгий пайплайн

Здесь на помощь приходят статические анализаторы.

Выбираем линтер

Статический анализ

Сформулируем наши требования.

Статический анализ

- Линтер должен работать быстро

Статический анализ

- Линтер должен работать быстро
- Нужна поддержка “своих” проверок

Статический анализ

- Линтер должен работать быстро
- Нужна поддержка “своих” проверок
- Важна поддерживаемость своих проверок

Статический анализ

- Линтер должен работать быстро

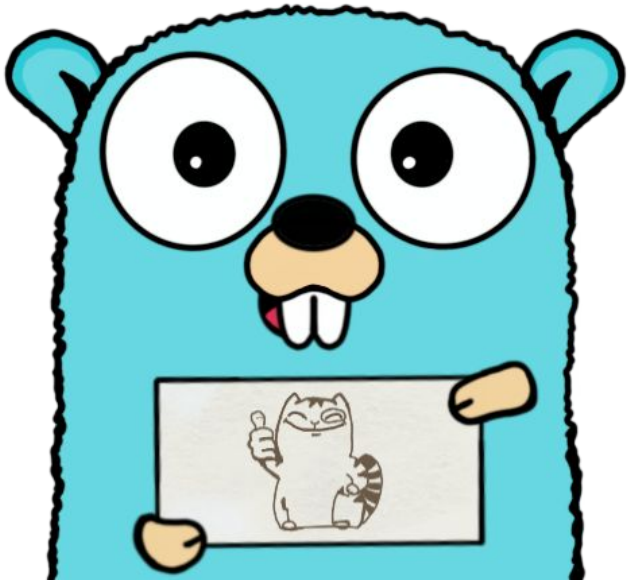
Линтеры на PHP не подходят по скорости.
Слишком сильно замедляют рабочий процесс.

Статический анализ

- Линтер должен работать быстро

Линтеры на PHP не подходят по скорости.
Слишком сильно замедляют рабочий процесс.

> Фатальный недостаток detected!



NoVerify

Мы написали свой PHP линтер на Go

Но почему на Go?

Go как язык для NoVerify

~~Потому что Юра выбрал Go.~~

Go - это хороший компромисс.

Go как язык для NoVerify

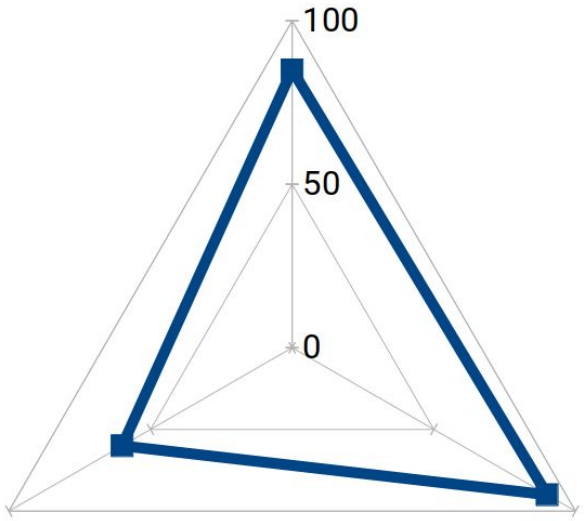
~~Потому что Юра выбрал Go.~~

Go - это хороший компромисс.

> Докажем на инфографике!

Высокоточный анализ

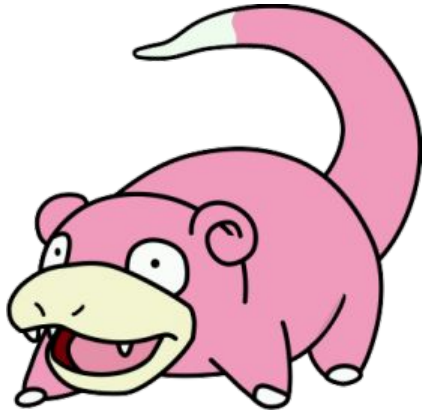
Скорость выполнения



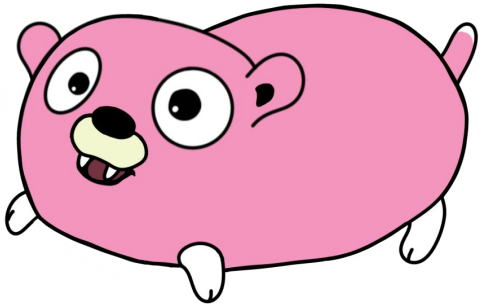
Скорость разработки

Простота поддержки

Высокоточный анализ



PHP - почти быстрый



Go - весьма быстрый

Go как язык для утилит

- Достаточно эффективен

Go как язык для утилит

- Достаточно эффективен
- Легко изучить на базовом уровне

Go как язык для утилит

- Достаточно эффективен
- Легко изучить на базовом уровне
- Проще в поддержке, чем многие ЯП

NoVerify

- Линтер должен работать быстро
- Нужна поддержка “своих” проверок
- Важна поддерживаемость своих проверок

NoVerify

- Линтер должен работать быстро
  - В несколько раз быстрее альтернатив
- Нужна поддержка “своих” проверок

- Важна поддерживаемость своих проверок

NoVerify

- Линтер должен работать быстро
  - В несколько раз быстрее альтернатив
- Нужна поддержка “своих” проверок
  - Можно писать расширения
- Важна поддерживаемость своих проверок

NoVerify

- Линтер должен работать быстро
  - В несколько раз быстрее альтернатив
- Нужна поддержка “своих” проверок
  - Можно писать расширения
- Важна поддерживаемость своих проверок
  - Легко тестировать и разрабатывать

Компромиссы интеграции

Первый запуск

```
$ noverify ~/myproject

...
```

Первый запуск

```
$ noverify ~/myproject

… (тонны текста)

Found 1057280 issues
```

Интегрируем

Что же делать?

Интегрируем

- Запуск в режиме diff

Интегрируем

- Запуск в режиме diff
- Добавление legacy/вендоров в exclude

Интегрируем

- Запуск в режиме diff
- Добавление legacy/вендоров в exclude
- Начать с подмножества проверок

Интегрируем

- Запуск в режиме diff
- Добавление legacy/вендоров в exclude
- Начать с подмножества проверок
- Собираем обратную связь от коллег

Git setup

diff режим подразумевает работу с git репозиториями.

- pre-push hook с линтером
- Опция --no-verify для обхода линтера
- На CI можно выставлять --git-full-analysis

Ложные срабатывания

Ложные срабатывания

- Мешают

Ложные срабатывания

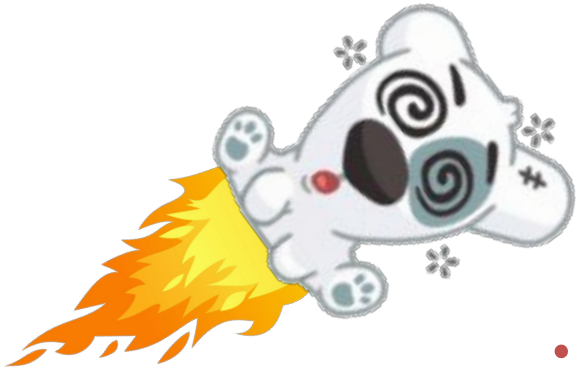
- Мешают
- Очень

Ложные срабатывания

- Мешают
- Очень
- Сильно

Ложные срабатывания

- Мешают
- Очень
- Сильно



git push --no-verify

Ложные срабатывания

```
function get_foo($obj) {
   return $obj->foo;
                ^^^
}

Warning:
Property “foo” does not exist
```

Придирчивость

```
$len = sizeof($x);
       ^^^^^^
```

```
Warning:
use “count” instead of “sizeof”
```

Придирчивость

Что с ней делать?

- Быть строгими и заставлять править всё-всё. Без исключений.

Придирчивость

Что с ней делать?

- Быть строгими и заставлять править всё-всё. Без исключений.

Работает не для любой команды. Некоторых коллег может настроить на бунт.

Придирчивость

Что с ней делать?

- Не блокировать пуш/коммит.
  Выдавать на уровне рекомендаций.

Придирчивость

Что с ней делать?

- Не блокировать пуш/коммит.
  Выдавать на уровне рекомендаций.

Но тогда 99% срабатываний будут игнорироваться.

Придирчивость

Что с ней делать?

- Разрешать некоторый уровень конфигурации для разных команд и разработчиков.

Придирчивость

Что с ней делать?

- Разрешать некоторый уровень конфигурации для разных команд и разработчиков.

Теряем в консистентности.

Придирчивость

Что с ней делать?

- Запускать подобные проверки раз в месяц, в рамках субботников.

Придирчивость

Что с ней делать?

- Запускать подобные проверки раз в месяц, в рамках субботников.

Требует ресурсов на автоматизацию.

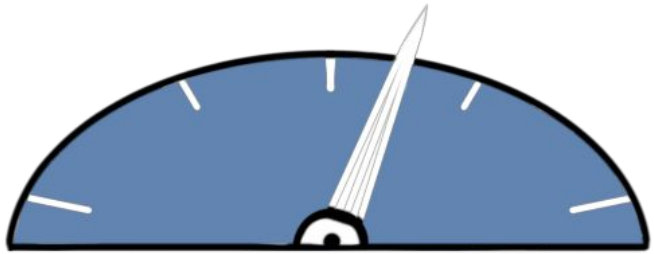
Придирчивость

Что с ней делать?

- А можно ничего не делать!

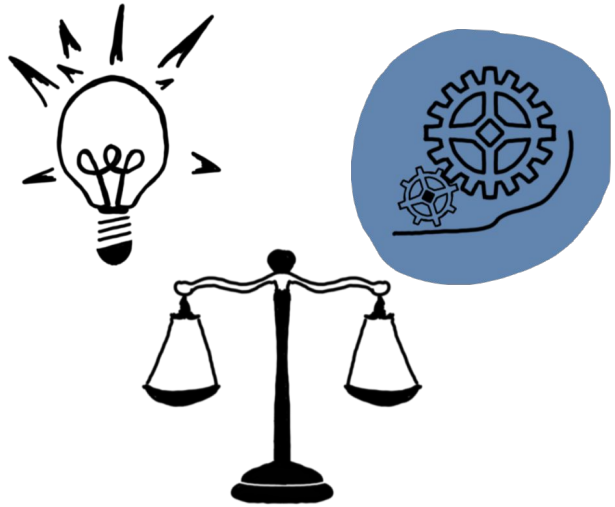
Компромисс

Счастливый линтер



Счастливые разработчики

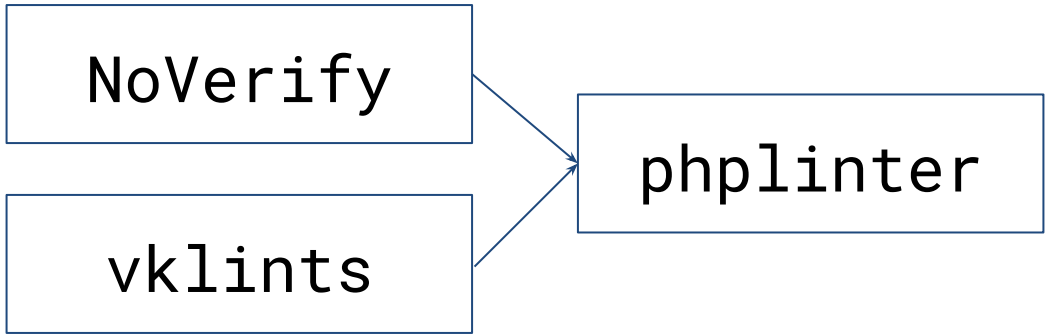
Если линтер мешает работе - это проблема.
Полезное действие должно превышать всё остальное.



Технические детали

Расширяемость

Приватные проверки ВК



vklints

Что представляют из себя vklints?

vklints

Что представляют из себя vklints?

- Проверки специфики использований некоторых наших функций и классов

vklints

Что представляют из себя vklints?

- Дополнительные проверки стиля

vklints

Что представляют из себя vklints?

- Требования использования строгого сравнения для некоторых типов

vklints

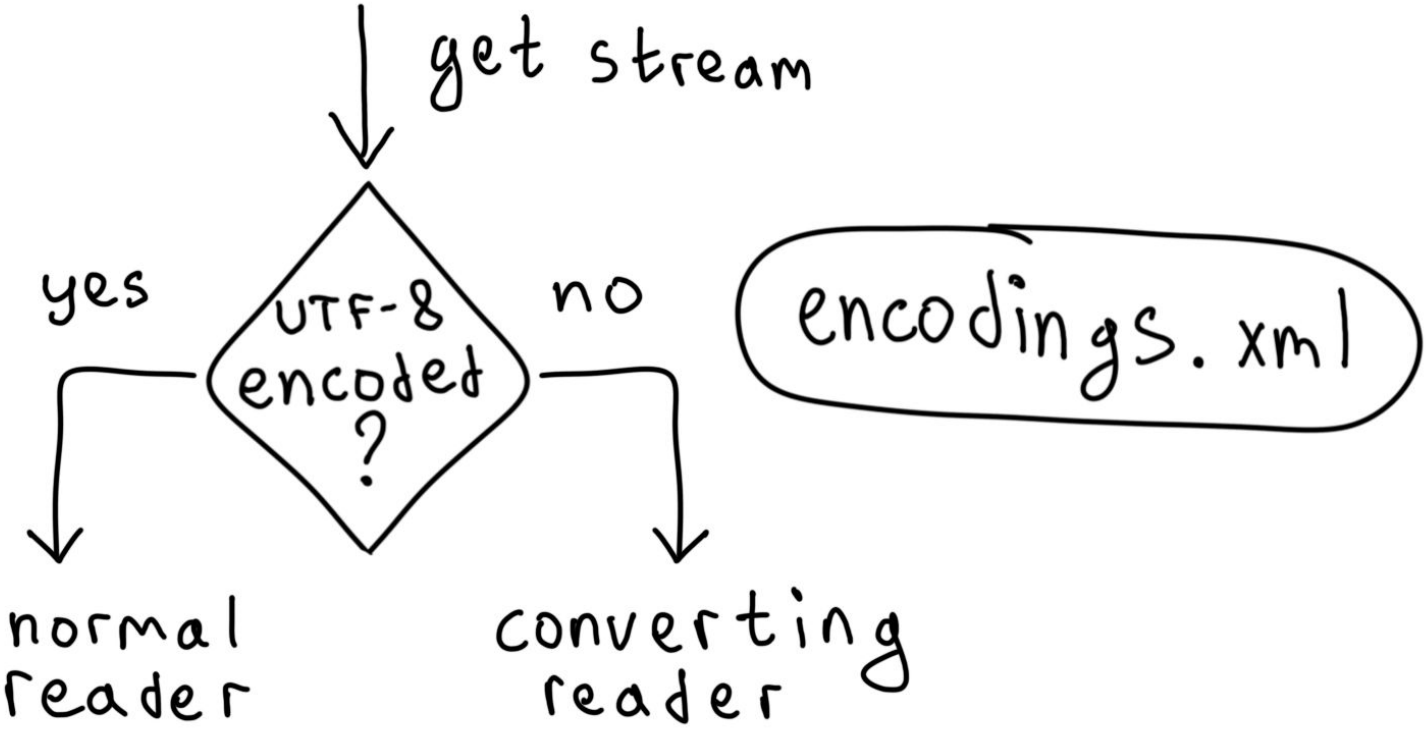
Что представляют из себя vklints?

- Подозрительные ключи массивов

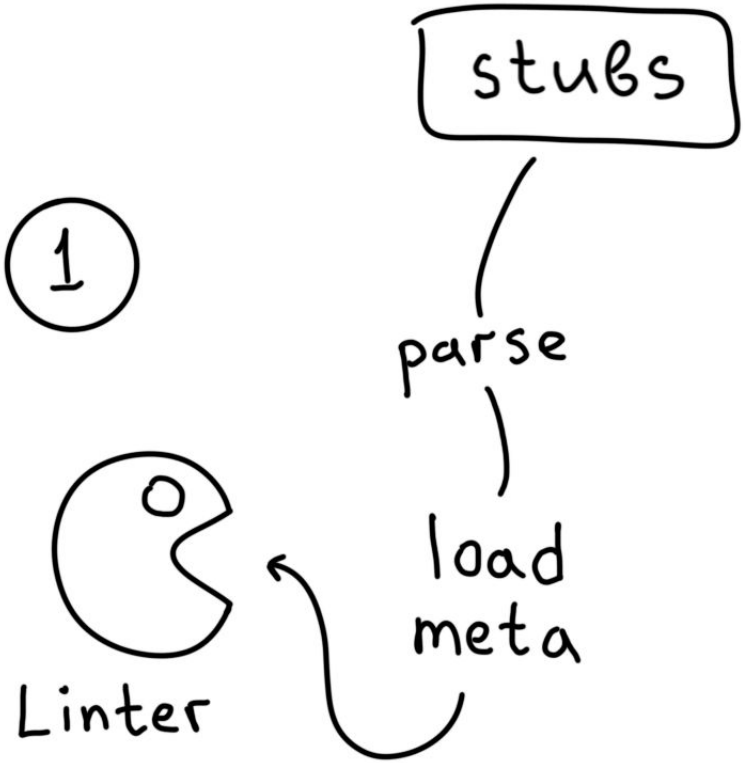
Как работает NoVerify

Go PHP parser

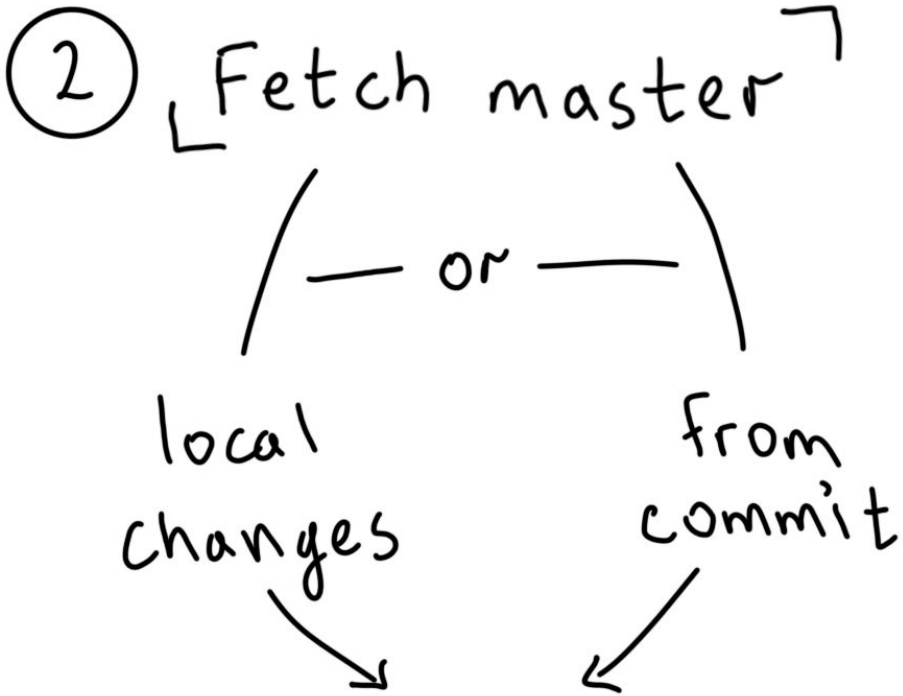
https://github.com/z7zmey/php-parser



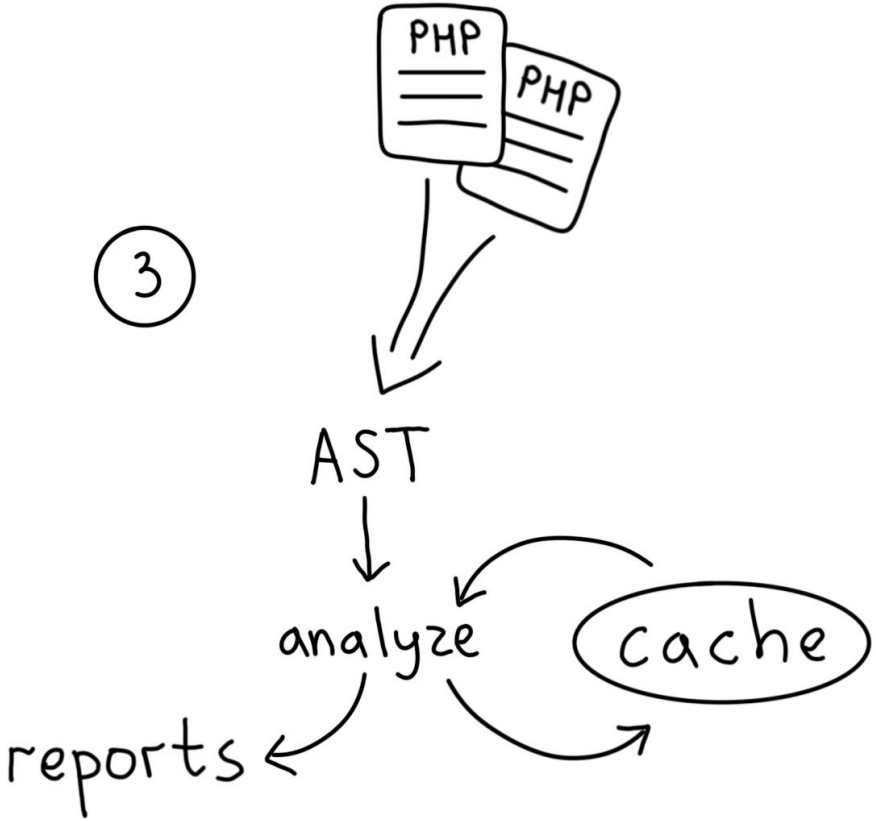
Используемый нами парсер умеет работать только с UTF-8.



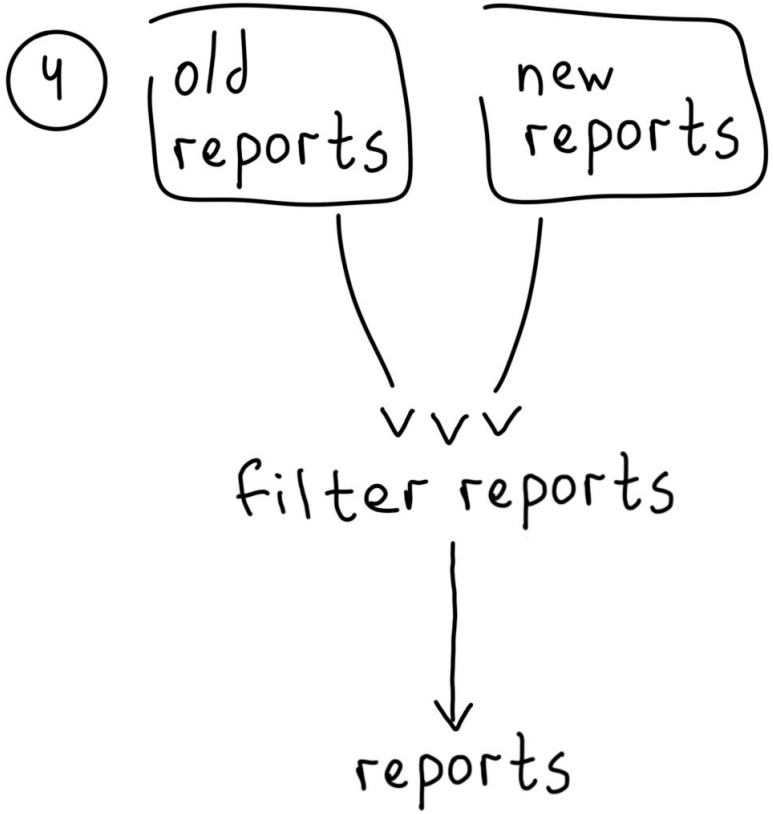
Загрузка метаданных из phpstorm-stubs.
Анализ override директив.



Подгрузка актуального master.
Выбор кода для проверка.



Разбор входных файлов.
Анализ AST, работа с кэшем.

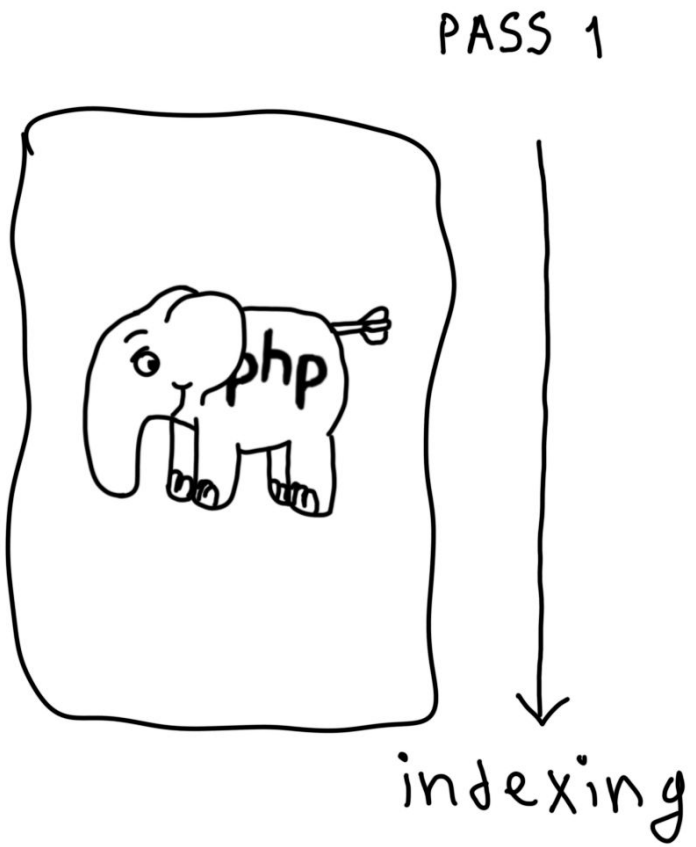


Вычисление разницы отчётов.
Фильтрация отчётов.

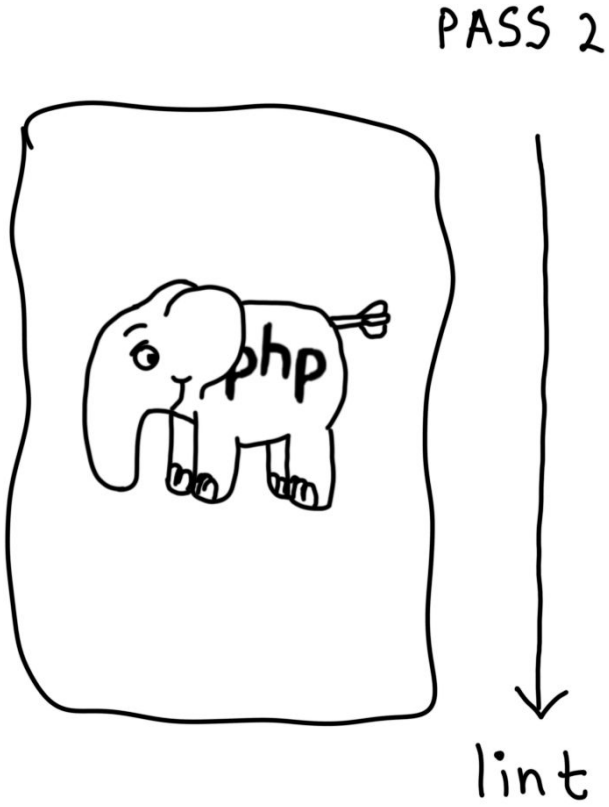
Двойной анализ кода (в режиме diff)

Можем себе это позволить!

Всё равно быстрее линтеров на PHP.

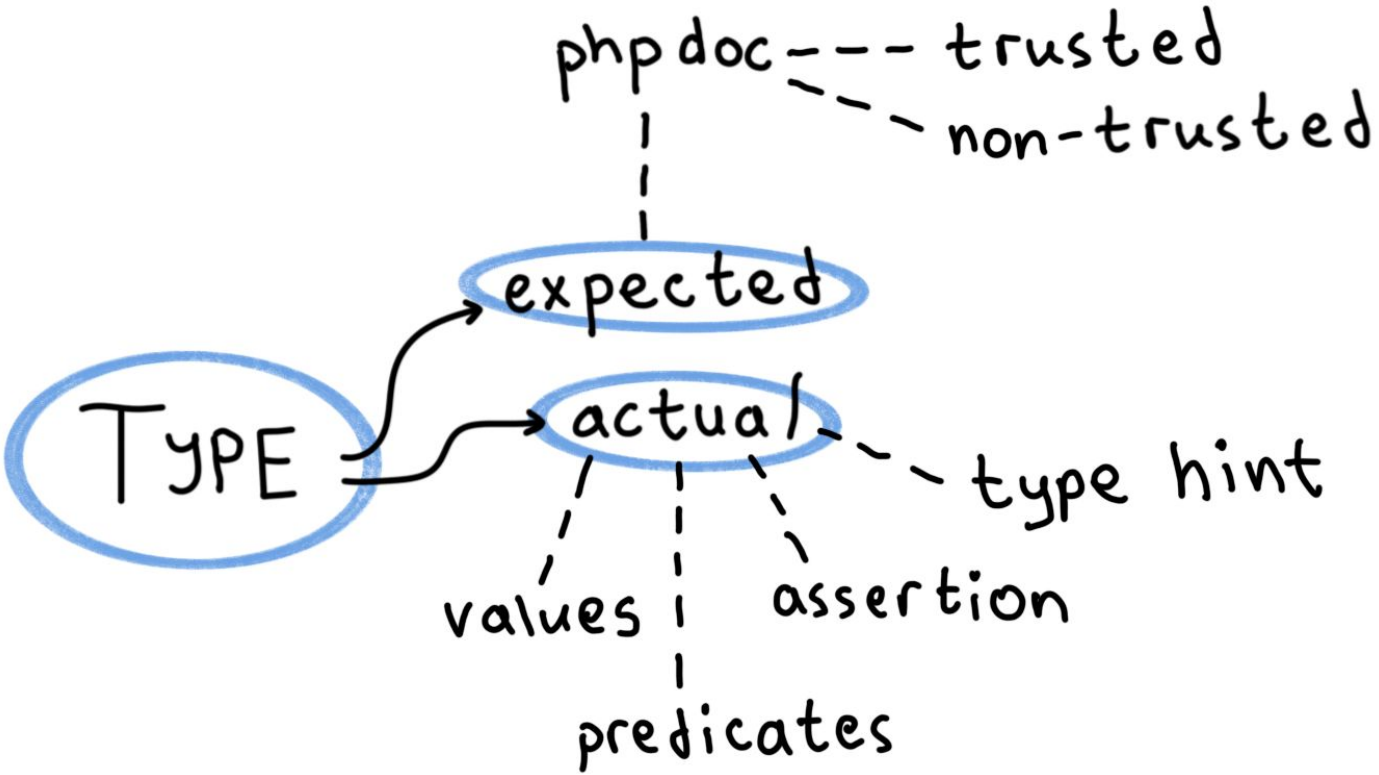


Первый проход по AST, индексация.



Второй проход по AST,
запуск проверок.

Вывод типов



Actual типы “сильнее”, но expected типы имеют больше семантики.

Expected vs actual

```
class Foo {
  /** @return static */
  public function newStatic() : self {
    return new static();
  }
}
// actual = Foo
// expected = static
```

Expected vs actual

```
class Foo {
  /** @return static */
  public function newStatic() : self {
    return new static();
  }
}
// actual - для PHP интерпретатора
// expected - подсказка для IDE/линтера
```

Type hints

```
declare(strict_types=1);
function f(T $foo) {
   $foo = 10; // Присвоили int
   return $foo;
}
// Ошибка ли?
```

Type hints

```
declare(strict_types=1);
function f(T $foo) {
   $foo = 10; // Присвоили int
   return $foo;
}
// Ошибка ли?
// * С точки зрения PHP - нет.
// * Но придраться к этому можно.
```

Type hints

```
declare(strict_types=1);
function f(T $foo) {
   $foo = 10; // Присвоили int
   return $foo;
}
// ? 1. f -> int
// ? 2. f -> T|int
// ? 3. f -> T
```

Type hints

```
declare(strict_types=1);
function f(T $foo) {
   $foo = 10; // Присвоили int
   return $foo;
}
// + 1. f -> int
// + 2. f -> T|int
// - 3. f -> T
```

Аннотации

```
/** @return int */
function f() { return "I lied!"; }

// * Комментарии могут вводить
//   в заблуждение.
// * NoVerify выводит actual return type
//   как string.
```

Наследование аннотаций

```php
interface IFoo {
  /** @return int */
  public function foo();
}
class Fooer implements IFoo {
  /** @inheritdoc */
  public function foo() { return "10"; }
}
```

Наследование аннотаций

```
interface IFoo {
  /** @return int */
  public function foo();
}
class Fooer implements IFoo {
  /** @inheritdoc */
  public function foo() { return 10; }
}
```

Наследование аннотаций

```
interface IFoo {
  /** @return int */
  public function foo();
}
class Fooer implements IFoo {
  /** @return string */
  public function foo() { return "10"; }
}
```

VKCOM/phpstorm-stubs

Fork, в который отправляются правки (затем создаются PR в upstream)

php-consistent & php-critic

Go -> PHP

go-critic -> php-critic

go-consistent -> php-consistent

php-consistent (in development)

Находит нарушения консистентности в коде.

```
is_int vs is_integer
```

php-consistent (in development)

Находит нарушения консистентности в коде.

```
!!$x vs (bool)$x
```

php-consistent (in development)

Находит нарушения консистентности в коде.

```
'x'=>$y  vs  "x"=>$y
```

php-critic (in development)

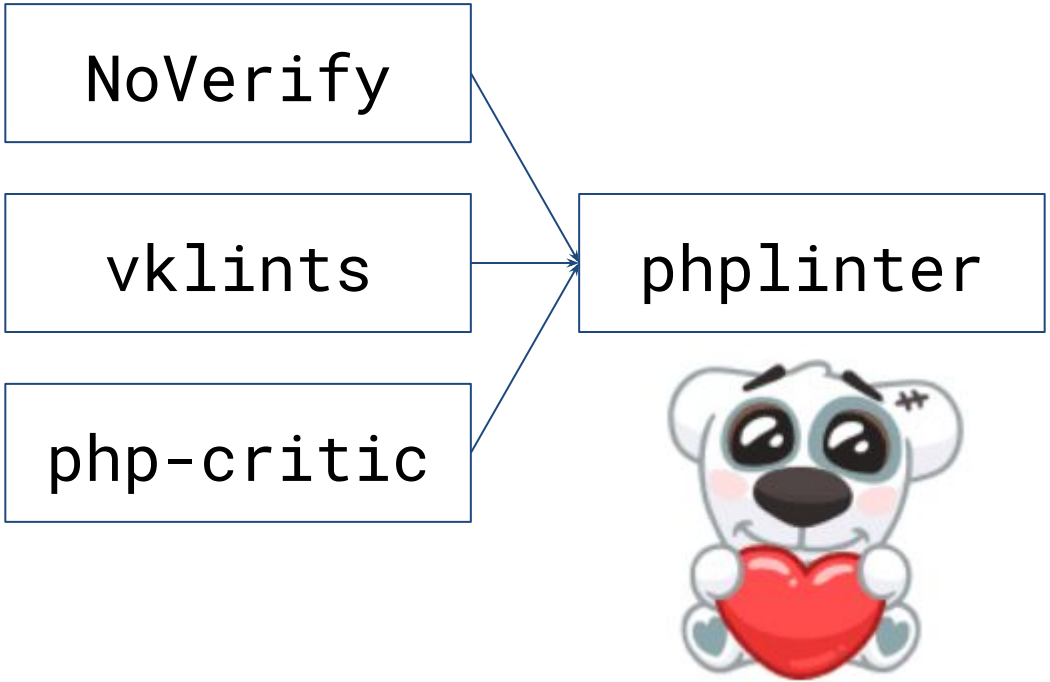
Набор наиболее привередливых проверок.

```
strcmp(s1, s2) > 0
```

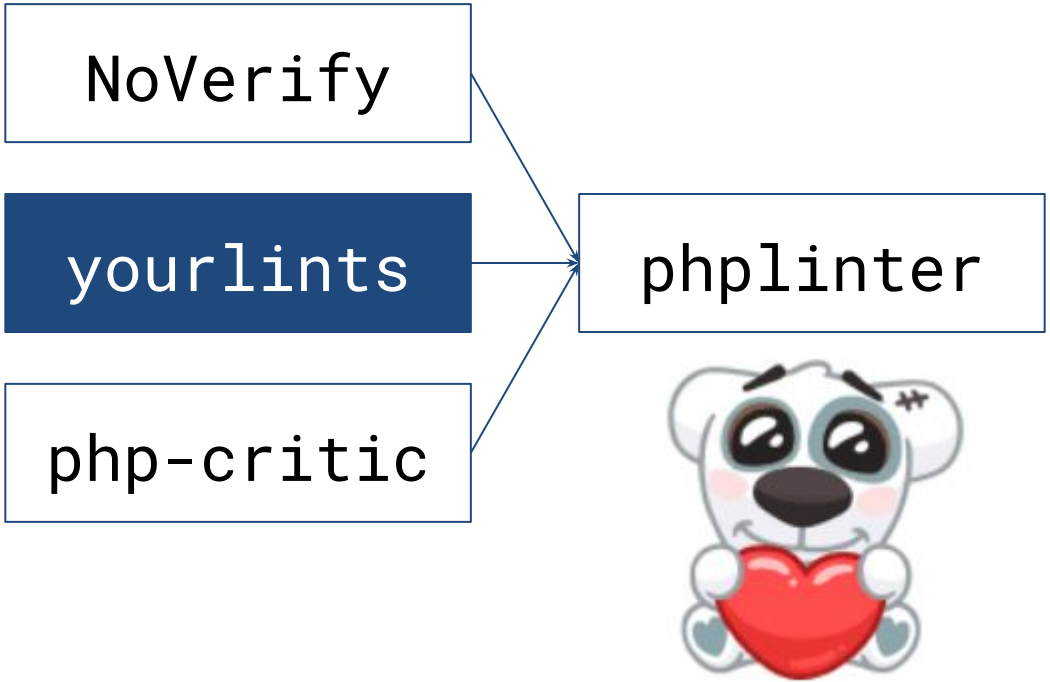
=>

```
s1 > s2
```

3 в 1



3 в 1



Open Source

Open Source

Если:

- Что-то сломалось / не запустилось

Open Source

Если:

- Что-то сломалось / не запустилось

Реакция:

- Негодовать, удалить NoVerify

Open Source

Если:

- Что-то сломалось / не запустилось

~~Реакция:~~

- ~~Негодовать, удалить NoVerify~~

Правильная реакция:

- Оформить тикет на GitHub

Open Source

Если:

- Вам не хватает какой-то фичи

Open Source

Если:

- Вам не хватает какой-то фичи

Реакция:

- Удалить NoVerify и написать свой линтер

Open Source

Если:

- Вам не хватает какой-то фичи

Реакция:

- Удалить NoVerify и написать свой линтер

> Звучит знакомо... Хорошее решение!

Open Source

Если:

- Вам не хватает какой-то фичи

~~Реакция:~~

- ~~Удалить NoVerify и написать свой линтер~~

Правильная реакция:

- Оформить тикет на GitHub

Гоферы

https://github.com/VKCOM/noverify

```
no_noverify or die("fin") ?>
```