



Дебаг

выясняем какая черепашка врет



Сегодня в программе

Экспешены

`gdb`

`time`

очевидные советы



Люди ошибаются



Программы содержат ошибки



Как сообщать об ошибках?
спойлер: подробно



Обычный код

```
value = self.database.get(key)
```

```
if value is not None:
```



Как сообщить что что-то пошло не так?

```
value = self.database.get(key)
```

```
if value is not None:
```

None - нет такого значения в базе?

Или произошла ошибка?



Способы сообщить об ошибке

Аварийно завершиться с подробным сообщением (упасть)



Способы сообщить об ошибке

Аварийно завершиться с подробным сообщением (упасть)

Вернуть специальное значение



Способы сообщить об ошибке

Аварийно завершиться с подробным сообщением (упасть)

Вернуть специальное значение

Вернуть признак ошибки отдельным параметром



Способы сообщить об ошибке

Аварийно завершиться с подробным сообщением (упасть)

Вернуть специальное значение

Вернуть признак ошибки отдельным параметром

Вернуть признак ошибки глобальной переменной



Способы сообщить об ошибке

Аварийно завершиться с подробным сообщением (упасть)

Вернуть специальное значение

Вернуть признак ошибки отдельным параметром

Вернуть признак ошибки глобальной переменной

Бросить исключение



Исключение (Exception) - это класс

Тип

Сообщение об ошибке (message)

Первопричину (cause)

Стек вызовов (stack trace)



Что делать?

Игнорировать



Что делать?

Игнорировать

Завернуть, добавить сообщение об ошибке и пробросить



Что делать?

Игнорировать

Завернуть, добавить сообщение об ошибке и пробросить

Обработать (например `retry`)



Either<Result, Exception>

Обертка над ответом и ошибкой

Может содержать только что-то одно



Что мы уже умеем

Декомпозиция

Фэйл-фаст, Эксершены

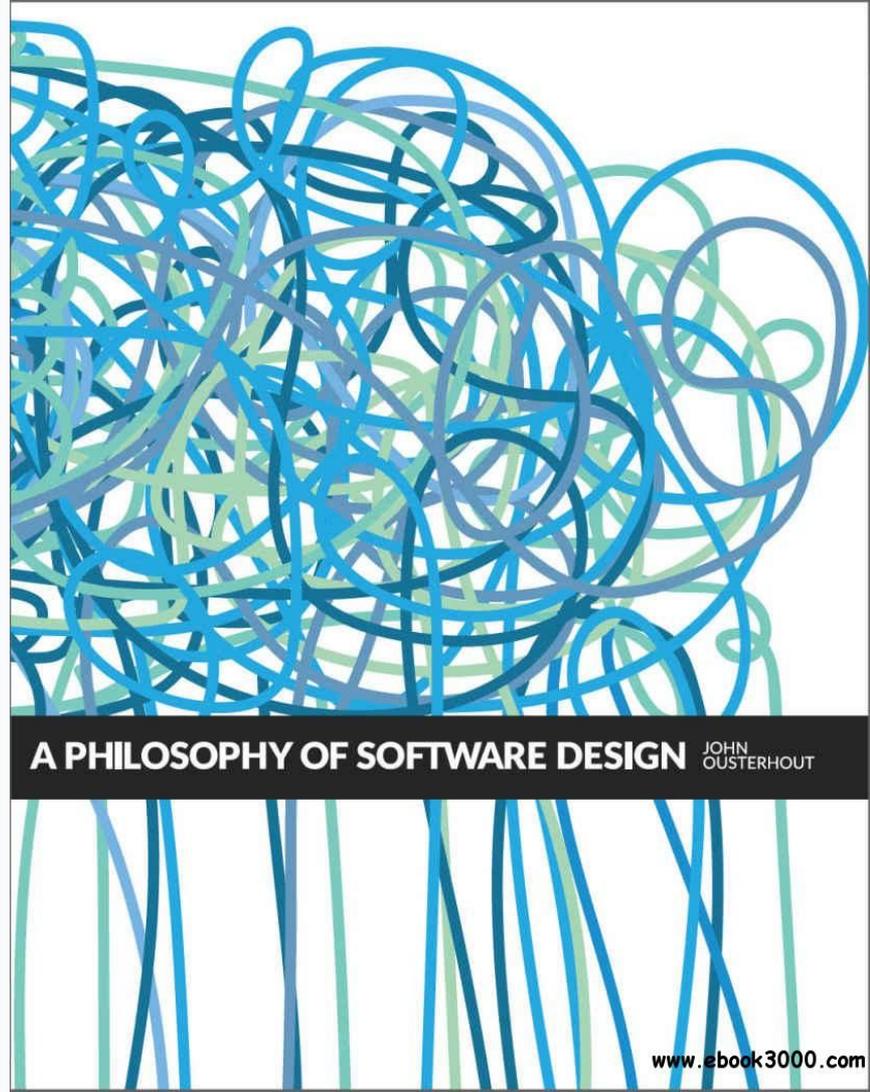
Юнит тесты

Логирование + Мониторинг (или гребание логов)



A philosophy of software design

John Osterhout





Следим за процессом

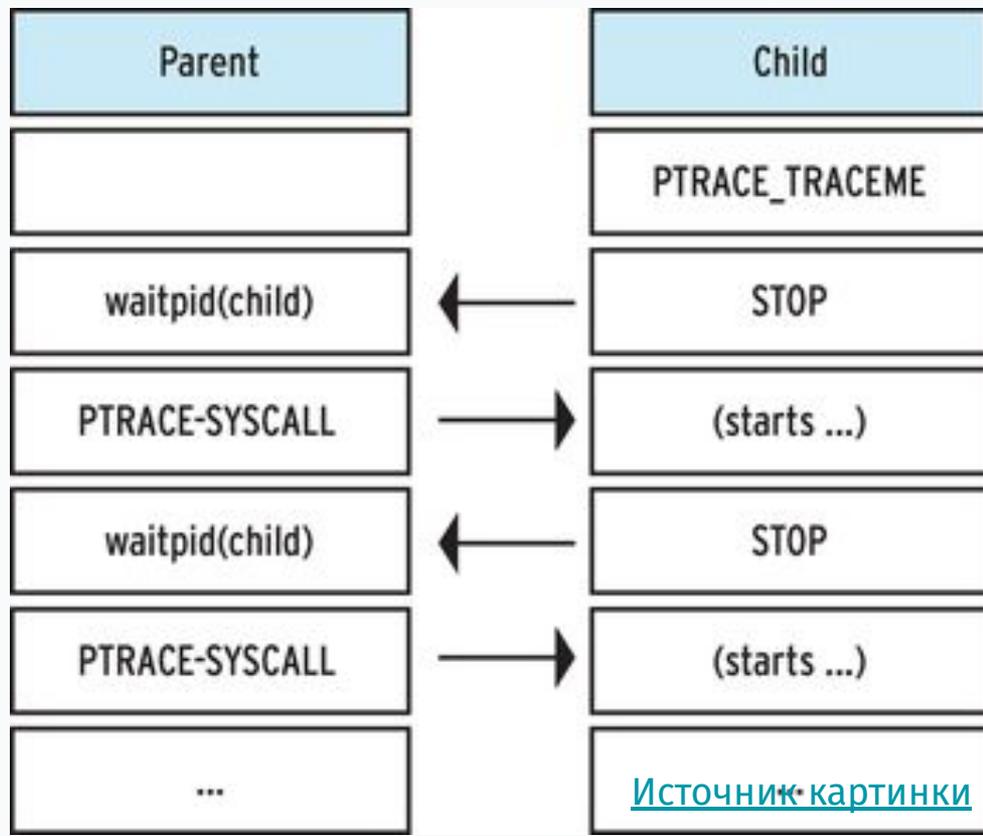


Что делает процесс?

`ptrace()`

`tracer/tracee`

`strace`





Дебагаем



gdb

брекпойнты

-g3

брекпойнты с условием (поток, значение
переменной и т.д)

в зависимости от ключей у нас может быть
разный код

step over. step into

dgb/lldb

потоки, стек, вычисление выражений

watches



ОСНОВНЫЕ КОМАНДЫ

run, start

next, step

info local/threads, frame

print, watch

break, continue

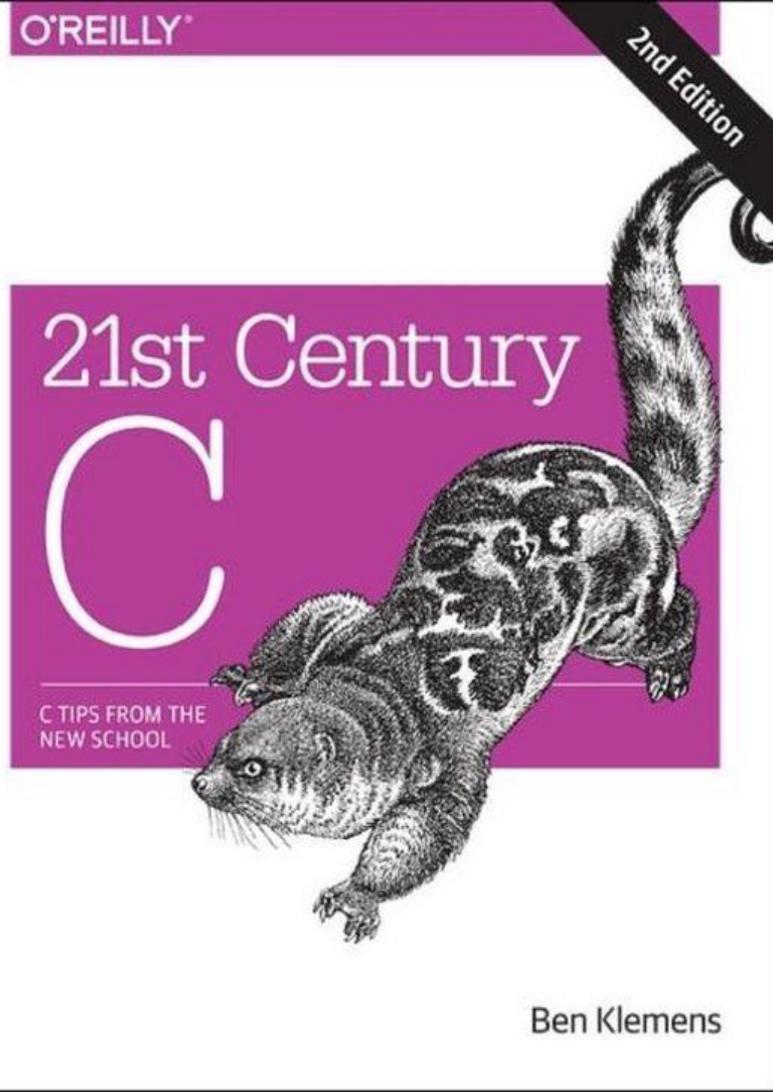
clear

quit



21st Century C

Ben Klemens





remote debug

```
gdb attach $PID
```

Виртуальная машина позволяет подключиться



IDEA

attach to process

analyze stacktrace

watches

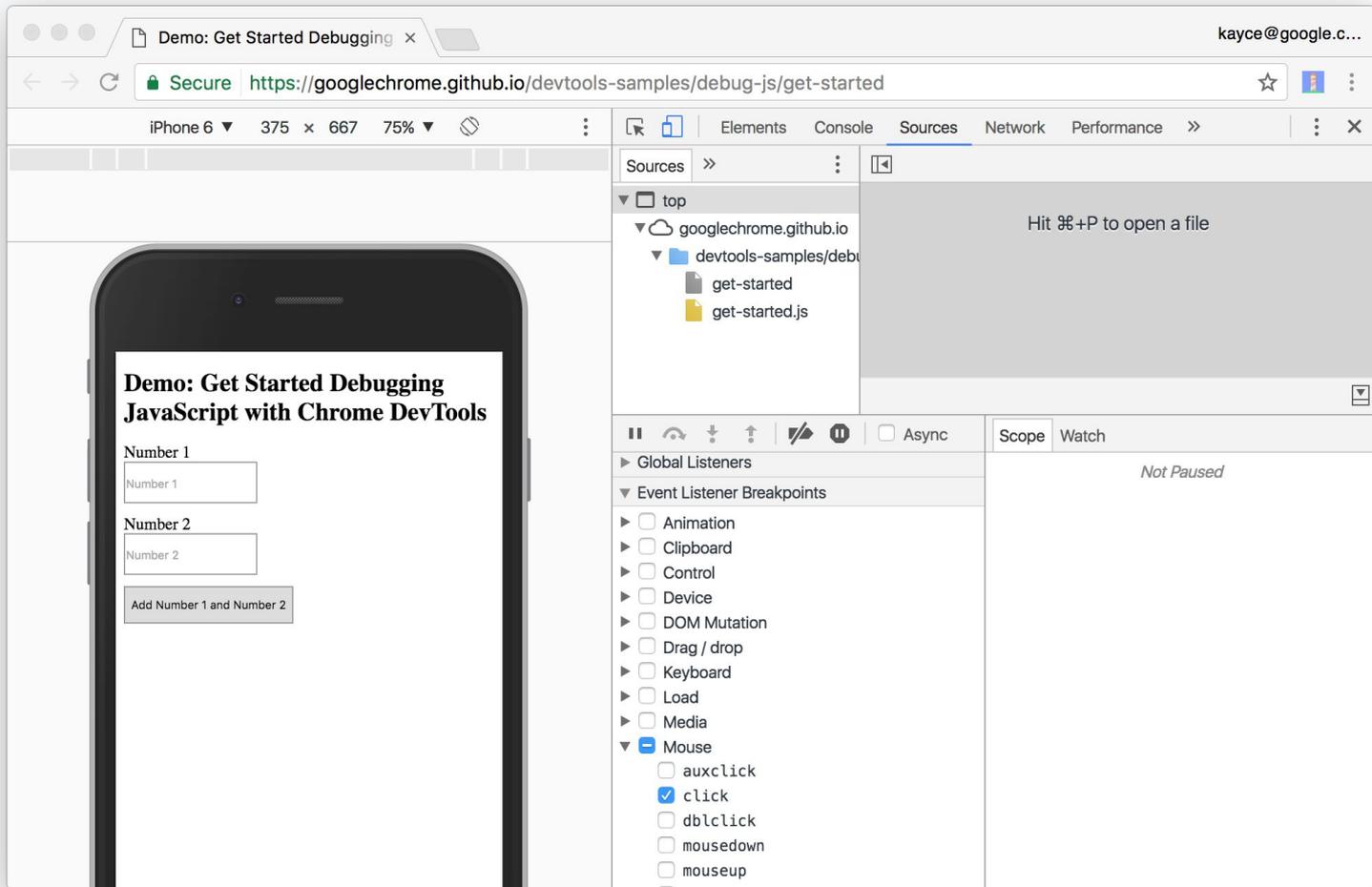
drop frame

break if exception, condition

Evaluate expression



WebConsole



[Источник картинки](#)



Профилируем по рабоче-крестьянски



Профилирование и анализ

time: real, user, sys

EXPLAIN SELECT FROM *

pyflakes

valgrind

flameGraph ([пример](#))

htop

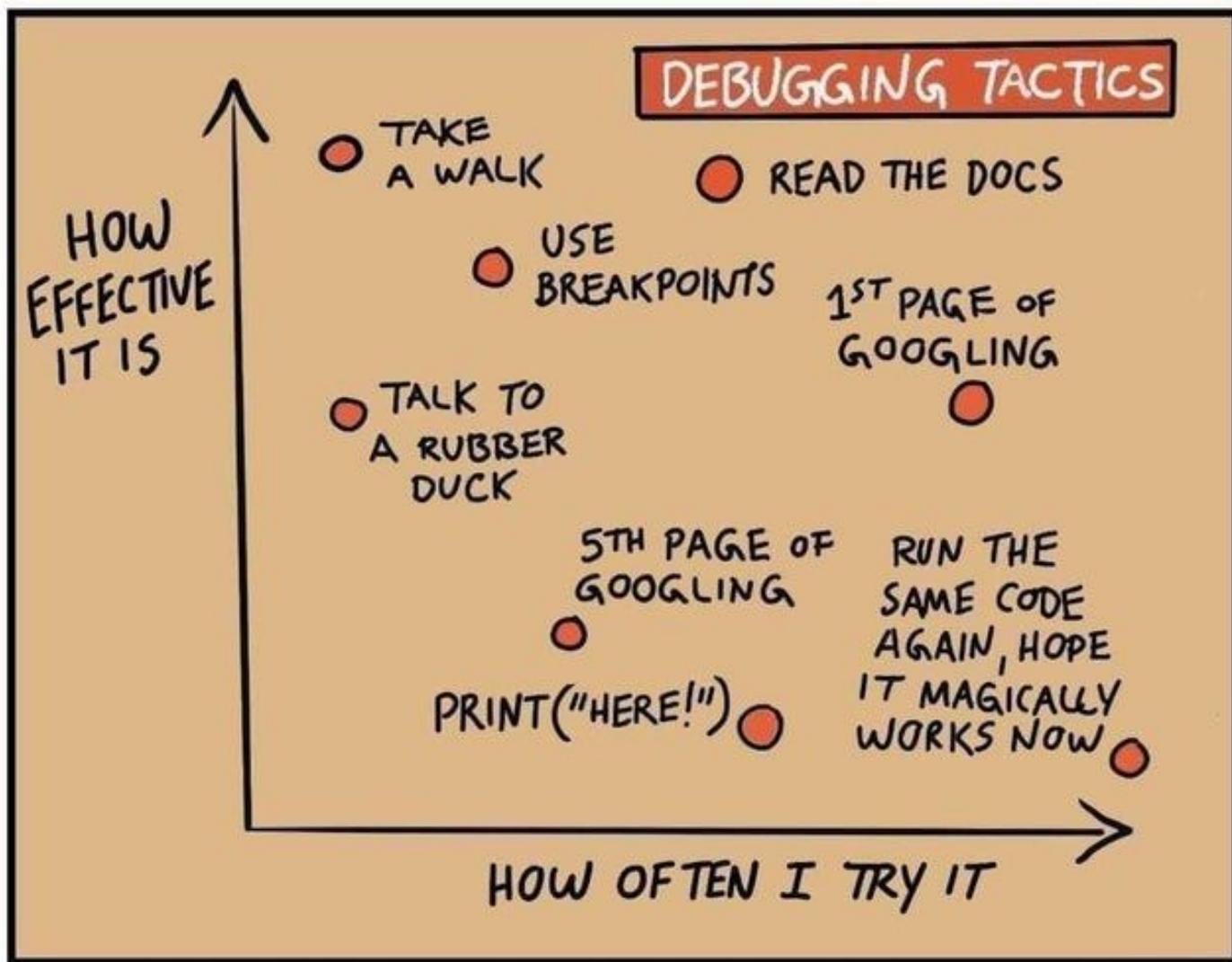
[МНОГО ИНТЕРЕСНОГО ТУТ](#)



Анализ сложности

time func(N)

time func(2*N)





"The most effective debugging tool is still careful thought, coupled with judiciously placed print statements."

— Brian W. Kernighan, *Unix for beginners*
1979



Чтобы понять почему что-то не работает, надо понять почему что-то работает



Ищите паттерны



Смотрите на последние изменения



Не повторяйтесь



Смотрите спектрейс



Читайте код



Расскажите комунибудь



Делайте ошибки воспроизводимыми



Сужайте область поиска



Используйте логгирование или принты



Используйте инструменты
статического анализа



Рисуйте



Конспектируйте



Популярные подходы

Ищите паттерны

Смотрите на последние изменения

Не повторяйтесь

Смотрите стектрейс

Читайте код (внимательно)

Расскажите код кому-нибудь

Поймите как воспроизвести ошибку

Используйте бинарный поиск

Принты / логи

Рисуйте

Используйте инструменты анализа

Записывайте

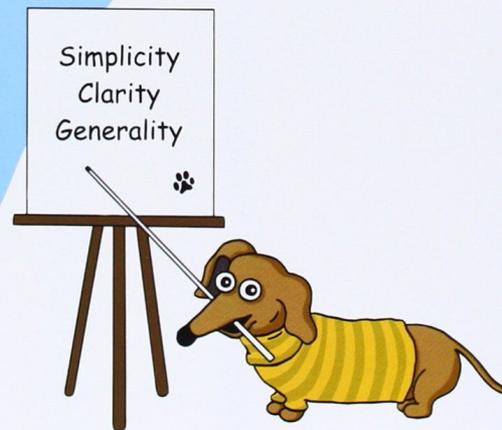


Что почитать

Та самая книжка с собакой

The Practice of Programming

Brian W. Kernighan
Rob Pike





Софт скиллз

Понимать как “это” должно работать

Построение и проверка гипотез

Чтение кода библиотек



Лучше предохраняться

Fail fast (defensive programming)

Юнит тесты

Статический анализ

Практики/паттерны



Спасибо!

Приходите еще