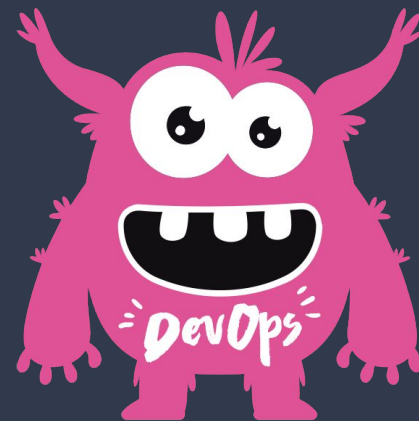


# Job security или как да стана неуволняем

Живко Ангелов

OpenFest 2019,  
3-ти Ноември 2019



---

# Dark Debt

---

# Кой съм аз ?



## ЖИВКО АНГЕЛОВ

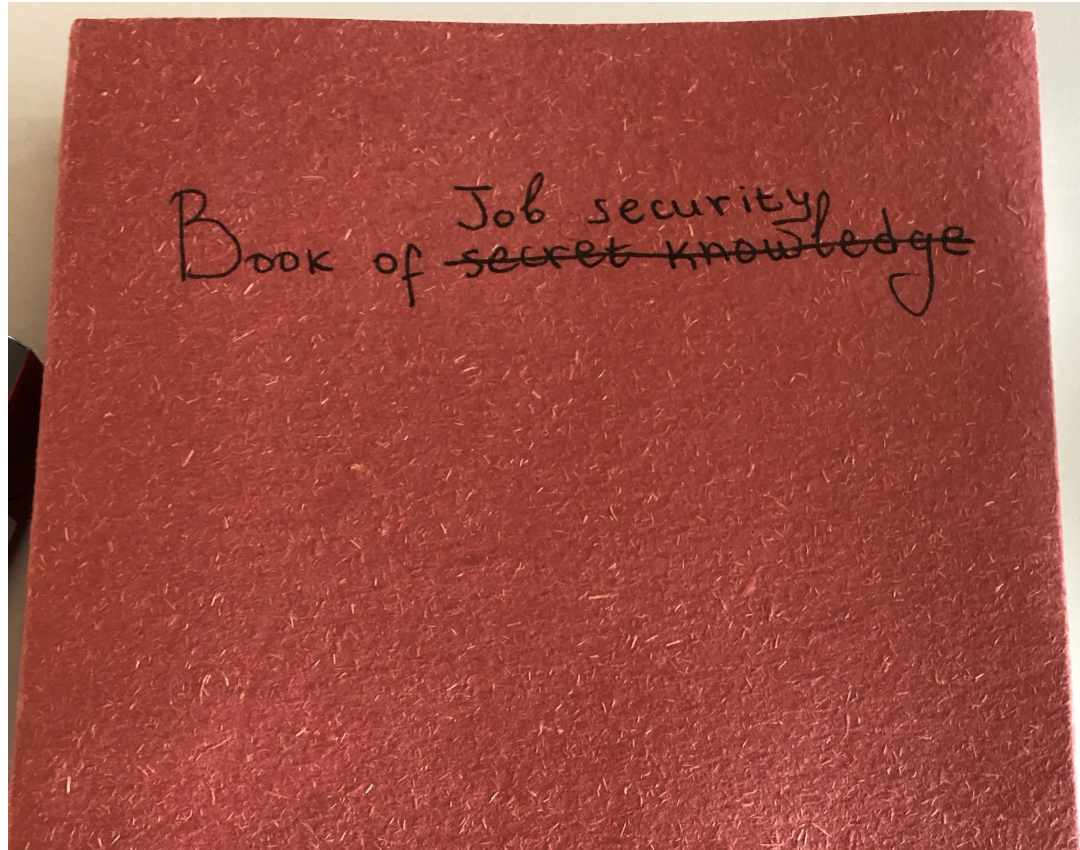
- DevOps @ SiteGround
- ExAdmin @ SiteGround
- 13 years @ SiteGround

---

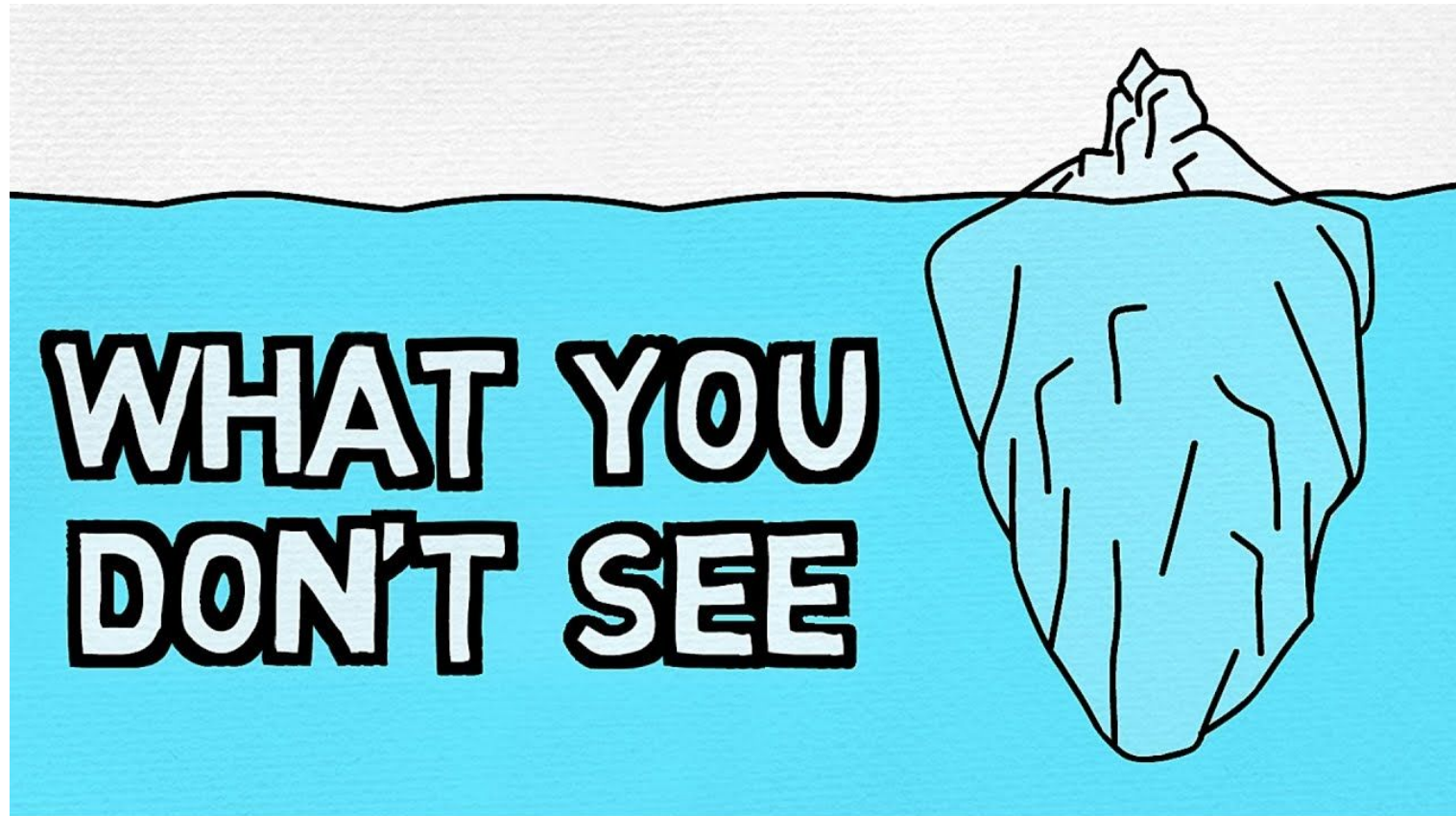
Какво е Job security ?

---

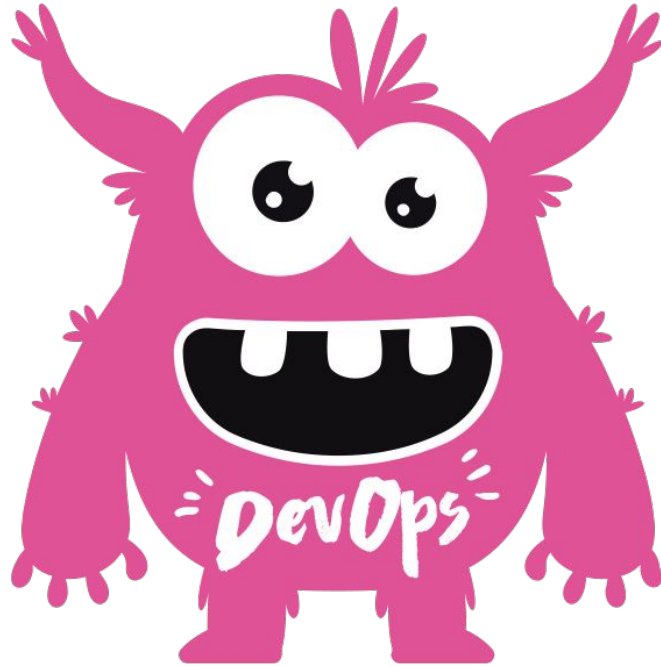
# Job security



# Dark Debt



# Chief Job Security Officer



---

# Job security & Software Development

---

# Променливи

- Списък с имена за бебета
- Променливи от една буква

```
for (let i=0; i<50; i++) {  
    s += (t[i]*4)/5;  
}
```

# Променливи

- Използвайте подобни имена

`parseInt` - `parseInt`, `hashtable` - `hashTable`

- Бъдете абстрактни

`it`, `everything`, `data`, `handle`, `stuff`, `do`,  
`routine`, `perform`

# Променливи

- Използвайте шльокавица

`$hendlyr $statiq $hranilishte`

- Множествено число от различни езици

`$knigi $statii`

# Променливи

- Преизползвайте имената на променливите
- Правете несъответствия

`Postal code -> $zip`

# Променливи

- Термини от математиката

`data = (slash + asterisk) / equals;`



# Променливи

- Кога да използваме  $i$  и кога  $n$
- Многократно задавайте стойност на променлива, не я използвайте

# Променливи

- Използвайте малко l, голямо O в имената на променливите

```
let a=1, O1=5, l1=10;
```

```
if (O=0)
```

```
    a=O1;
```

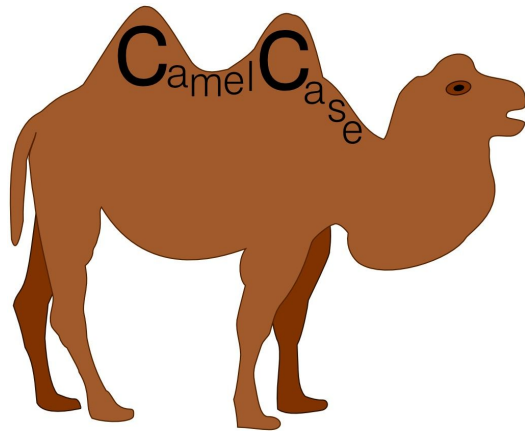
```
else
```

```
    l=l1;
```

# Функции/методи

- БЕЗРАЗБОРНО ГОЛЕМИ И МАЛКИ БУКВИ

`performUnitTest()` -> `PerforMunIttEst()`



# Функции/методи

- Правете повече (или по-малко) от това, което пишете

`isValid(x)`



# Функции/методи

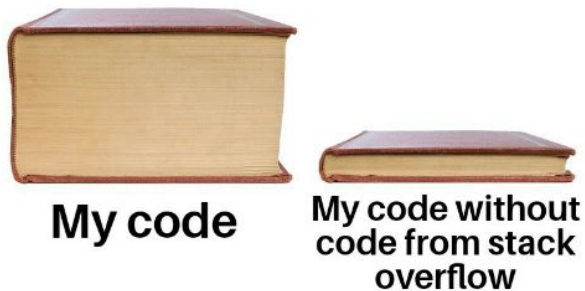
- Професионални правописни грешки

`HandlerOpening()` -> `HandlreClosing()`



# Функции/методи

- Копирайте функции от stackoverflow без да ги използвате



Stack overflow is a  
programming language.

# Функции/методы

- Размествайте параметри

`drawRectangle (height, width) -> drawRectangle (width, height)`



# Функции/методи

- Overloading, overloading, overloading
- Дублицирайте и разхвърлете

`setAlignment(int alignment) -> setLeftAlignment, setRightAlignment, setCenterAlignment`

- Правете големи кюфтета
- Бъдете абстрактни

`doJob() , performIt() , PerformDataFunction()`

# Коментари

- Послъгвайте в коментарите
- Документирайте очевидното

```
i += 1 // add 1 to i
```

- Документирайте Как, а не Защо
- Никога не слагайте коментар при декларация на променлива

# Коментари

- Камофлаж

```
for (j=0; j<array_len; j+=8) {  
    total += array[j+0];  
    total += array[j+1];  
    total += array[j+2]; /* Main body of  
    total += array[j+3]; * loop is unrolled  
    total += array[j+4]; * for greater speed.  
    total += array[j+5]; */  
    total += array[j+6];  
    total += array[j+7];  
}
```

# Генерални техники

- Многомерни масиви
- The Kama Sutra
- Пътуване във времето
- Влагайте възможно най-дълбоко
- Възможно най-дълги линии

# Генерални техники

- **Early returns, goto. Spaghetti code**
- **Избягвайте {} ако е възможно**
- **Изберете език за програмиране, който никой друг не знае във фирмата**
  - **Само вие можете да го оправите**
  - **Няма кой да прави код ревю, може да правете каквото си поискате**

# Генерални техники

- Шефа знае най-добре
- Не прекомпилирайте!
- Правете го така, както само вие си знаете
- Third Party Libraries

# Генерални техники

- Магически масиви

```
data[2][0] = true;
data[2][2] = "OpenFest";
data[2][8] = 2019;
data[2][5] = 11.5;
data[2][6] = data[0][2]
data[2][12] = 2018;

if (data[2][12] > data[2][8] && data[2][2] == "OpenFest" && data[2][0]) {
    data[3] = data[2];
    data[3][8] = 2020;
}
```

# Генерални техники

- Използвайте ;

```
if(a);  
else;  
{  
int d;  
d = c;  
}  
;
```



# Генерални техники

- Вложени switch-case

```
switch(a) {  
  case 1:  
    switch(b) {  
      case 8:  
        switch(c) {  
          case 12:  
            switch(d) {  
              case 18:  
                // something  
                break;  
                ....  
            }  
            break;  
            ....  
          }  
          break;  
          ....  
        }  
        break;  
        ....  
      }  
      break;  
      ....  
    }  
    break;  
    ....  
  }  
}
```

# Генерални техники

- “Unused Variable” грешки или как да бъдем по-умни от компилатора

```
i = i;
```



# Version Control Systems

- **Git ?!**
- **Създайте своя собствена система чрез магически окончания и symlinks**

`api_GEN.pl`

`api_GEN6.pl`

`api_OLD2.pl`

`api_latest-stable8.pl`

`api_latest-stable3.pl -> api_GEN6.pl`

# Тестове

- **Unit tests - никога**
- **Performance tests - никога**



---

# Job security & DBA

---

# Job security & DBA

- Кирилица в името на таблица
- Space в името на таблица
- SQL Aliasing - a-z
- SQL Outer Join
- Дълги PL/SQL функции
- Бизнес логика в PL/SQL функции

---

# Job security & DevOps

---

# БАШОВИЗЪМ

- Използвайте `bash`, много `bash` (по ~3000 реда на скрипт).
- Генерални (фалшиви) имена на скриптове - `fix_everything.sh`
- One liners

```
[[ -d directory ]] && echo yep || echo nope
```

# БАШОВИЗЪМ

- Използвайте уникалността на bash

```
lower() {  
    # Usage: lower "string"  
    printf '%s\n' "${1,,}"  
}
```

```
$ lower "HELLO"  
hello
```

# БАШОВИЗЪМ

- To upper case

```
#{1^^}
```

- Reverse case

```
#{1~~}
```

# БАШОВИЗЪМ

- Trim leading and trailing white space from string

```
trim_string() {  
    # Usage: trim_string " example string "  
    : "${1#"${1%%[![:space:]]*}"}"  
    : "${_%"${_##*[![:space:]]}"}"  
    printf '%s\n' "$_"  
}
```

# Perl гимнастики

- Oneliners

```
# iptables -L -nvx | perl -lane '$_ = $F[0]; $pkts +=  
$_ if /^d/; END { print $pkts }'
```

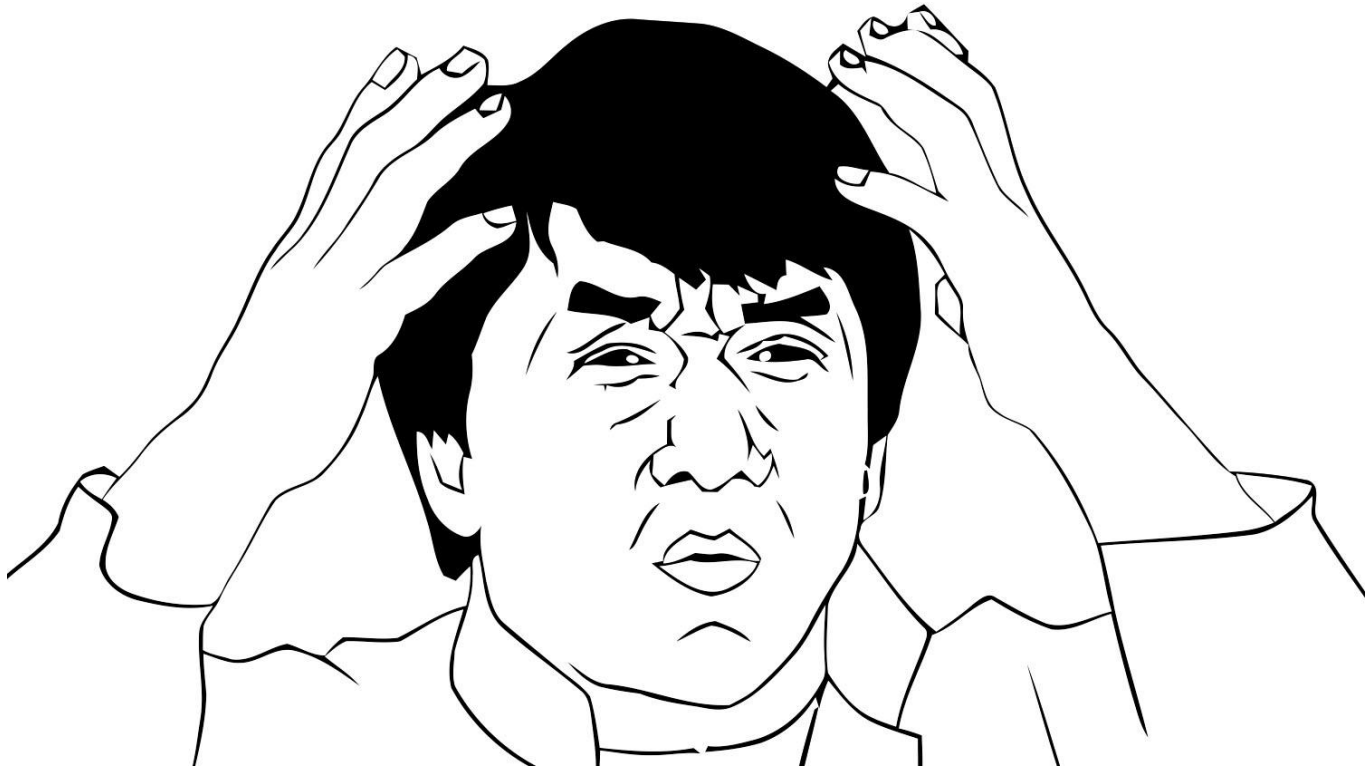
- Използвайте trailing if и unless особено на дълги линии

# Perl гимнастики

- Ternary operators колкото може повече.  
Вложете ги.

```
return ($a<$b) ? ($b<$c) ? $b : ($a<$c) ? $c : $a :  
($a<$c) ? $a : ($b<$c) ? $c : $b;
```

# Перл гимнастики



# Цитати

“не го пипай, ние и без това ще го махаме скоро”

“качи го така, пък после ще го оправим ...”

... 13 години по-късно – кой го е писал това ?!?

---

# ИЗВОД

---

# ИЗВОД



**JOB SECURITY**

We can all be replaced

---

Благодаря!

---

---

Въпроси ?

---

# ИЗТОЧНИЦИ

<https://github.com/Droogans/unmaintainable-code>

<https://github.com/dylananaraps/pure-bash-bible>

# Обратна връзка

