

...ers, Developers, Developers, Developers, Dev...

Кирпиченков Денис

Naumen

# История о разработке одного продукта

- Время разработки 10 лет
- 3 перерождения: Python, Java + JSP, Java + GWT
- 1 000 000 строк java-кода, 60 000 строк скриптов
- Распределенная команда из 20 разработчиков в 3 городах

# Read,Code,Debug,Test... Repeat

## Общие алгоритм работы

- ① прочитать требования;
- ② реализовать;
- ③ показать реализацию;
- ④ отдать в тестирование;
- ⑤ goto 1/goto 2.

## Исправление дефектов

- ① воспроизвести дефект;
- ② проанализировать причины;
- ③ исправить поведение.

## Улучшение (производительности) продукта

- ① проанализировать проблему;
- ② замерить текущие показатели;
- ③ исправить;
- ④ замерить новые показатели;
- ⑤ goto 3.

# Что общего у (почти) всех программистов



# void function career() { while(1) { doCodeALot() } }

общенный путь разработчика

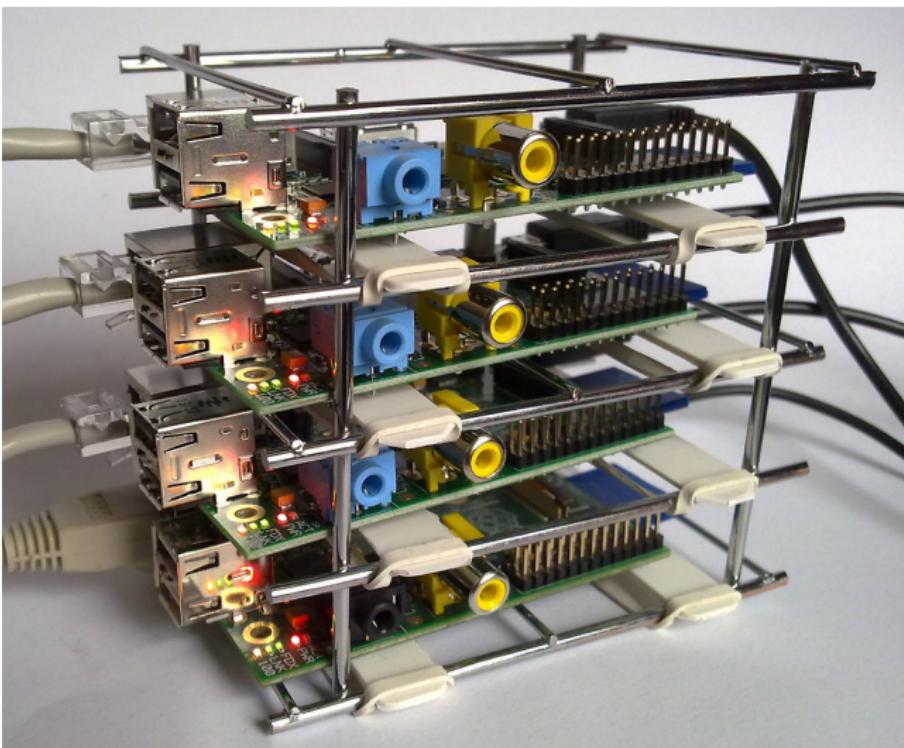
- Стажер
- Разработчик
- Ведущий разработчик



# “Типы” программистов

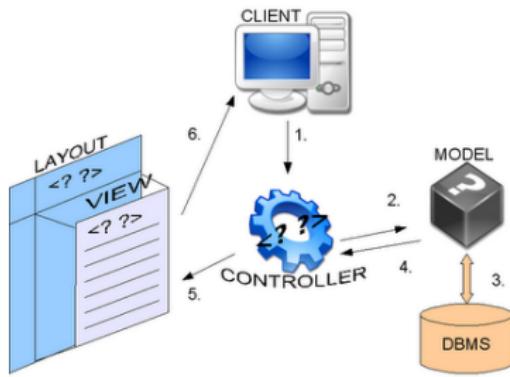
- Embedded разработка
- Enterprise разработка
- Mobile разработка
- *Gamedev*
- *Desktop-приложения*

# Embedded решения

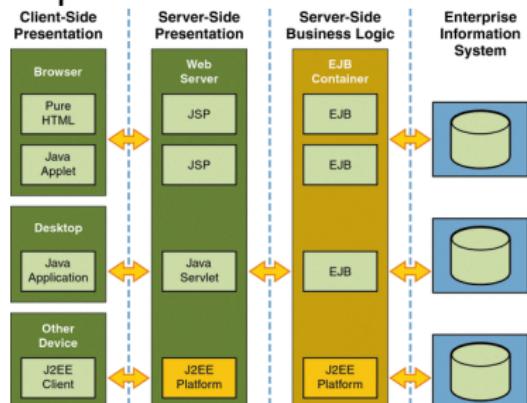


# Enterprise разработка

## Версия 1.0



## Версия 18.5.2.8



# Mobile apps



Hello, world!

Polarization

$$I = I_0 \cos^2 \theta$$

refraction

$$n_1 \sin \theta_1 = n_2 \sin \theta_2$$

critical angle

$$\theta_c = \sin^{-1} \frac{n_2}{n_1}$$

$I = 4I_0 \cos^2 \left( \frac{\pi}{2} \theta \right)$

disangle  $\theta$  and  $(\frac{\pi}{2})$  maxima  
disangle  $= (\pm \frac{\pi}{2})$  minima

higher  $n$

dispersion angle

$$C_d = \frac{d\theta}{d\lambda}$$

thin film

$$2L \left( \frac{m + \frac{1}{2}}{\lambda} \right) \text{ (maxima/minima)}$$

single slit diffraction

$$a \sin \theta = m \lambda \quad (m=1,2,3) \text{ minima}$$

$$I = I_0 \left( \frac{\sin \theta}{\theta} \right)^2 \quad \theta = \frac{1}{2} \pi \sin \theta$$

Gravitational Deflection

$$\theta = 1.72 \frac{R}{c^2} \text{ first minimum}$$

opposite direction

is  $\theta < R/c^2$  it is avoided.

Newton's ring

$$d \sin \theta = m \lambda \text{ (maxima/minima)}$$

signature

$R = N \lambda$  radius of curvature

$\frac{1}{f} = \frac{1}{R} + \frac{1}{z}$  spherical mirror

ratio of curvatures

$$M = -\frac{1}{f}$$

image distance

$$f' = -\frac{h'}{h} = -\frac{1}{M}$$

real image

Plane mirror

$$i = -p$$

$S$  is  $i$  if on same side as incoming light

$S'$  is  $i$  if on same side of outgoing light

$F = p$  if on same side of outgoing light

# Что нужно знать

- Язык программирования
- Типы данных
- Особенности аппаратного обеспечения и сети

# M x N

Есть матрица M x N.

$$M = \begin{bmatrix} 1 & 2 & 3 & 5 \\ 3 & 4 & 5 & 7 \\ 1 & 2 & 3 & 5 \\ 3 & 4 & 5 & 7 \\ 1 & 2 & 3 & 5 \\ 3 & 4 & 5 & 7 \end{bmatrix}$$

Цель: обработать все элементы матрицы наиболее эффективным способом.

Как это более эффективно сделать, с точки зрения быстродействия, по колонкам или по строкам матрицы?

Caveat!



# Finale



# Finale



<https://github.com/d0k1/U-R-DEVELOPER>