

Как мы создавали карту развития для разработчиков

Сергей Черепанов (FSD, ptichka.moscow)



Saint
TeamLead
Conf **2019**

Профессиональная
конференция
про тимлидов
и для тимлидов



Разработчик с 2012 года

Frontend: TypeScript + React

Backend: ~~Python, Elixir~~, теперь Haskell

Страдаю стихийным саморазвитием

Профиль команды — Front-end outsourcing

- JavaScript.
- TypeScript.
- React.
- Redux.



Проект Ptichka.Moscow

- Курьерская служба срочной доставки по городу.
- Front-end: TypeScript, React, Redux, Redux-Saga.
- Back-end: Python, Elixir, Haskell, AWS.
- 250+ доставок в день.

Курьерская доставка по Москве

Войти | Отследить заказ

Сегодня

Срочно

Введите адрес откуда забрать посылку

Большая Дмитровка, 25, офис 31

с 11:00 до 15:00

быть на адресе

Выбрать на карте

Введите адрес, куда доставить

2-я Брестская улица, 8, кв 31

с 11:30 до 15:30

быть на адресе

Выбрать на карте

+

Добавить адрес

📦

Пакет с одеждой

документы

🌿

🌸

🍰

🍷

🍷

🍷

🚗 на автомобиле

+

Поручения курьеру

Принять наличные

Купить без предоплаты

Другое поручение

😊

Добавьте чаевые в знак благодарности. Курьеру будет приятно

-

0

+

👤

Наличными курьеру

Главы доклада

1. Проблемы, боли и решение.
2. С чего мы начинали.
3. К чему мы пришли.
4. Саммари.

Проблемы, боли и решение

Глава 1

№1. Неосознанная некомпетентность

Некомпетентность, которая ещё даже не определена на радаре.

Разработчик в компании уже несколько лет, но всё равно...

Спотыкается на фундаментальных вещах: знание вёрстки или JavaScript

Не знает даже основ проектирования

Проектирует с явным нарушением принятых практик (например, в ООП),
а значит — делает код менее поддерживаемым.

Плодит велосипеды

Не знает про debounce и пишет ad-hoc-решение.

Проблема №2. Беспорядочные повышения

- Кто громче просит, тому доплачивать.

Проблема №2. Беспорядочные повышения

- Кто громче просит, тому доплачивать.
- За теми, кто явно не просит, следить индивидуально и крайне внимательно.

Проблема №2. Беспорядочные повышения

- Кто громче просит, тому доплачивать.
- За теми, кто явно не просит, следить индивидуально и крайне внимательно.
- Разработчикам приходится постоянно идти на “неудобные” разговоры.

Проблема №3. Сложность распределения команды по проектам

- Возможность переключать участников между проектами.

Проблема №3. Сложность распределения команды по проектам

- Возможность переключать участников между проектами.
- Быстрое определение исполнителей по компетенциям.

Проблема №3. Сложность распределения команды по проектам

- Возможность переключать участников между проектами.
- Быстрое определение исполнителей по компетенциям.
- Поиск общего языка в команде.

Варианты решения проблем

- Забить и оставить хаос :)

неэффективно

Варианты решения проблем

- Забить и оставить хаос :)

неэффективно

- One to one + performance review + ИПР

дорого, субъективно

Варианты решения проблем

- Забить и оставить хаос :)

неэффективно

- One to one + performance review + ИПР

дорого, субъективно

- Карта развития

Решение — карта развития

- Список вопросов по теории для каждого грейда.

Решение — карта развития

- Список вопросов по теории для каждого грейда.
- Приём грейда производится теми, кто сам его сдавал недавно.

Решение — карта развития

- Список вопросов по теории для каждого грейда.
- Приём грейда производится теми, кто сам его сдавал недавно.
- Бесконечное количество попыток.


Решение — карта развития

- Список вопросов по теории для каждого грейда.
- Приём грейда производится теми, кто сам его сдавал недавно.
- Бесконечное количество попыток.
- Свободный темп сдачи на уровень.





Все уровни на Github

[Code](#) [Issues 18](#) [Pull requests 7](#) [Projects 0](#) [Wiki](#)

Branch: master ▾ [developers-roadmap](#) / [frontend](#) /

 **Znack** Remove-oop-book (#97)

..

 junior-1	Add questions about 'use strict' (#95)
 junior-2	53 add questions by time junior 2 (#72)
 junior-3	8 add questions by typescript Junior-3 (#48)
 middle-1	Remove-oop-book (#97)

Каждый уровень разбит по темам

 [books.md](#)

jun3: rewrite security topic

 [build.md](#)

jun3: move build to separate file

 [css.md](#)

fix notes

 [data.md](#)

jun3: refactor data.md questions

 [dom.md](#)

Statements are reformulated into questions

 [git.md](#)

jun3/git: add bisect (#89)

Тема раскрывается вопросами

- Что такое асинхронность? Чем она полезна?
- Какие в серверной разработке есть альтернативы?
- Какие есть минусы у асинхронного подхода?
- Что такое Event Loop?
- Что такое Job queue?
- Почему для асинхронных действий не получится использовать `try..catch` ?

```
try {  
    setTimeout(function() { throw new Error('Async error'); }, 1000);  
} catch (e) {  
    console.log(e);  
}
```

Цифры

- **50 часов** на первую версию.
- **750 часов** на текущую версию.
- **0.5 года** — самый быстрый рост от джуна до мидла.
- **500+** вопросов на три уровня джуниоров.


С чего мы начинали


Глава 2

Карту вели в rizzoma.com



Junior 

Старший Junior 

Middle первой категории 

- Работа с данными 

- Асинхронность 

- Что такое асинхронность, чем она полезна и какие в серверной разработке есть альтернативы?  

- Что такое Event Loop? 

Карта развития — простые списки вопросов

- По ± 50 вопросов на уровень.

Карта развития — простые списки вопросов

- По ± 50 вопросов на уровень.
- 2 уровня джуниора (low junior, high junior).

Карта развития — простые списки вопросов

- По ± 50 вопросов на уровень.
- 2 уровня джуниора (low junior, high junior).
- 3 уровня мидла.

Джуниор

- Хорошо владеет базовыми инструментами.
- Решает локальные задачи под присмотром.
- HTML.
- CSS.
- JavaScript.
- React or Angular or Vue or ...
- Пишет unit-тесты.

Мидл

- Понимает бизнес-ценности своих задач.
- Имеет базовые навыки проектирования.
- ООП vs ФП.
- SOLID.
- DDD.
- Основы клиент-серверной архитектуры.
- KISS, YAGNI, etc.
- ...

Сеньор

- Проектирует.
- Обучается сам.
- Силён в коммуникациях.
- Высокоуровневая архитектура:
 - большая система проектов «над» фреймворками;
 - кастомная модульная библиотека (например, сложные графики).
- Может объяснить:
 - бизнес-ценности технарям;
 - технические проблемы нетехнарям.
- Знает, что изучать, как, когда, зачем.

Интервью

1. Найти 2-3 разработчика, последними сдавших уровень.



Алексей
Джун

Интервью

1. Найти 2-3 разработчика, последними сдавших уровень.



Максим
Мидл



Дмитрий
Мидл



Алексей
Джун

Интервью

1. Найти 2-3 разработчика, последними сдавших уровень.
2. Ответить по всем пунктам чеклиста (± 3 часа).



Максим
Мидл



Дмитрий
Мидл



Алексей
Джун

Интервью

1. Найти 2-3 разработчика, последними сдавших уровень.
2. Ответить по всем пунктам чеклиста (± 3 часа).
3. Иногда отвечать на вопросы “между строк”.



Максим
Мидл



Дмитрий
Мидл



Алексей
Джун

Интервью

1. Найти 2-3 разработчика, последними сдавших уровень.
2. Ответить по всем пунктам чеклиста (± 3 часа).
3. Иногда отвечать на вопросы “между строк”.
4. Получить фидбэк и доизучать.



Максим
Мидл



Дмитрий
Мидл



Алексей
Джун

Проблема — общие формулировки большинства вопросов

— *В чём сила, брат?*

— *Хм...*

Привязали уровни к вознаграждению

Подводные камни, осторожно!

Почему это сработало у нас:

- Узкая специализация.

Привязали уровни к вознаграждению

Подводные камни, осторожно!

Почему это сработало у нас:

- Узкая специализация.
- Качественно проработаны уровни развития.

Привязали уровни к вознаграждению

Подводные камни, осторожно!

Почему это сработало у нас:

- Узкая специализация.
- Качественно проработаны уровни развития.
- Открытая атмосфера в команде.

Мы забыли погуглить :)

- Могли взять за основу аналоги.
- Делали всё с нуля.

Сдача уровня разом

- Подготовка каждого уровня занимала 4+ месяцев.
- Начальные темы забывались, пока готовились последующие.

Самостоятельный поиск источников

- Для каждого уровня каждый искал подходящие источники.
- Не было никаких рекомендаций, кроме комментариев коллег.

Создание карты одним человеком

90% вопросов в карте написано силами одного разработчика

К чему мы пришли

спустя 3 года

Глава 3

Содержимое темы

До

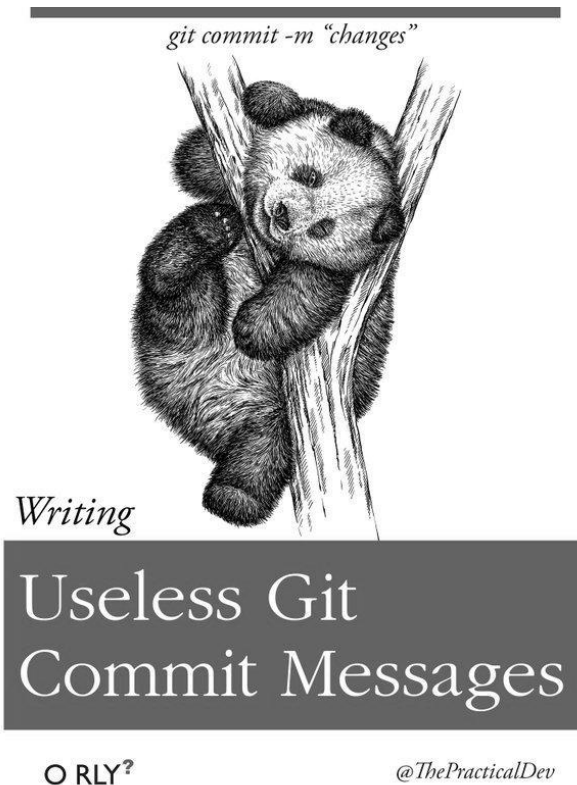
- Список вопросов.

После

- Список вопросов.
- Рекомендуемые источники.
- Единое форматирование.

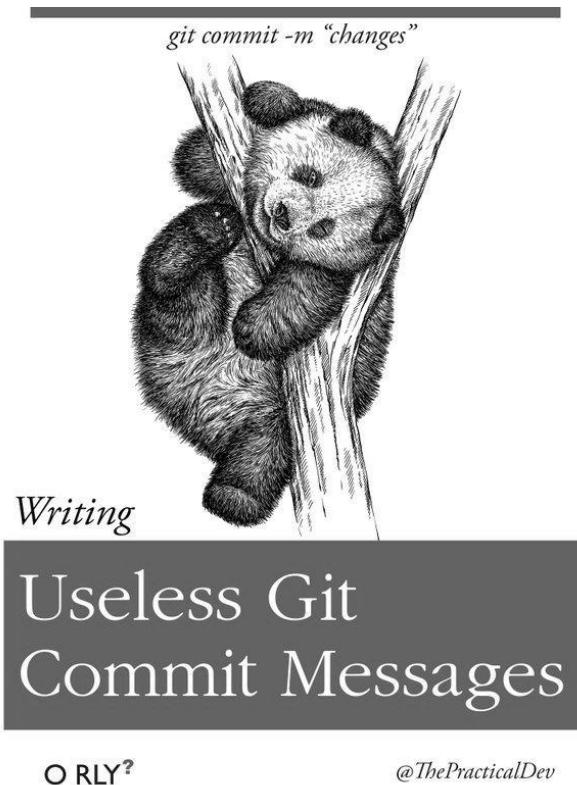
Не только разработка

- Деловая переписка: структура текста по Б. Минто.



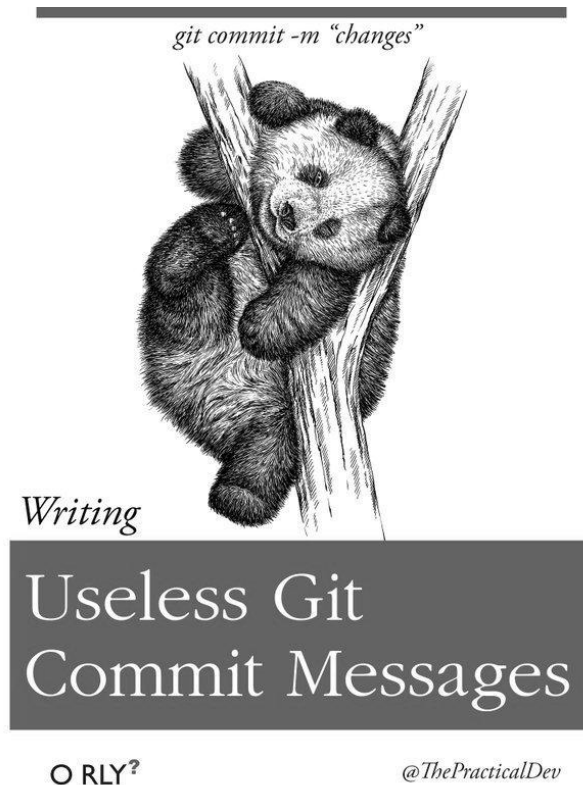
Не только разработка

- Деловая переписка: структура текста по Б. Минто.
- Основы методов управления Scrum/Kanban.



Не только разработка

- Деловая переписка: структура текста по Б. Минто.
- Основы методов управления Scrum/Kanban.
- Планирование:
 - декомпозиция задач;
 - управление временем;
 - приоритизация.



Совместное создание уровней

- Создаётся Issue. Возможен полный отказ от темы.

Отказались от вопросов по Redux-Saga.

Совместное создание уровней

- Создаётся Issue. Возможен полный отказ от темы.

Отказались от вопросов по Redux-Saga.

- Пишется Pull Request. Возможна полная переработка темы.

Полностью переписали вопросы по TypeScript.

Review от 2-8 человек

Помогает

- находить компромисс через общие принципы;

Review от 2-8 человек

Помогает

- находить компромисс через общие принципы;
- приводить к общему знаменателю;

Review от 2-8 человек

Помогает

- находить компромисс через общие принципы;
- приводить к общему знаменателю;
- завязать холивар на 100+ комментариев;

8 add questions by typescript Junior-3 #48

Edit

Merged

Znack merged 44 commits into `fullstack-development:master` from `GoodDayToday0key:8-add-questions-by-typescript-Junior-3` 20 days ago

Conversation 142

Commits 44

Checks 0

Files changed 2

+65 -0

Интервью — не только оценка

- Принимающие закрепляют изученное.
- Рассказывающий расширяет ассоциативные связи.

Изученная, но не вынесенная на интервью книга, быстро забывается.

Рефакторинг карты

Убираем темы и источники, которые себя не оправдали, например:

- «Чистый код» Р. Мартина.
- «Объектно-ориентированное мышление» М. Вайсфельда.

Повторяем для back-end



Совмещаем с программой обучения

- **3 года** работает бесперебойно.
- **10.000 человек** приступило.
- **4-7 месяцев** длится обучения.

Полный цикл от зелёного юнца до могущественного бойца.

Планируем ошибки

- Не используем систему как инструмент увольнения сотрудников.
- Склоняемся к ошибкам первого рода (false positive), не второго (false negative).
- Соблюдаем баланс между объективностью и субъективностью.

Мотивация роста по карте

 Прокачка скиллов.

Мотивация роста по карте

 Прокачка скиллов.

 Повышение грейда и денежного вознаграждения.

Мотивация роста по карте

 Прокачка скиллов.

 Повышение грейда и денежного вознаграждения.

 Fun!

Фишка для найма

- Привлекаем страждущих, а они сами к нам тянутся.

Фишка для найма

- Привлекаем страждущих, а они сами к нам тянутся.
- Объясняем ценности.

Фишка для найма

- Привлекаем страждущих, а они сами к нам тянутся.
- Объясняем ценности.
- Договариваемся об условиях.

Инструмент онбординга

- Адаптация новых сотрудников.

Инструмент онбординга

- Адаптация новых сотрудников.
- Быстрый переход к продолжению изучения.

Инструмент онбординга

- Адаптация новых сотрудников.
- Быстрый переход к продолжению изучения.
- Знакомство с командой.

Эксперимент: ежемесячный one-to-one

Начали с 3-х сотрудников:

- Только пришедший разработчик.

Эксперимент: ежемесячный one-to-one

Начали с 3-х сотрудников:

- Только пришедший разработчик.
- «Пинатель», который не продвигался по карте больше 2 лет.

Эксперимент: ежемесячный one-to-one

Начали с 3-х сотрудников:

- Только пришедший разработчик.
- «Пинатель», который не продвигался по карте больше 2 лет.
- Опытный удалёнщик.

Планы: практические упражнения

Пример

На тему ООП дать разработку Service Worker'а с задачами:

- Хранение данных.
- Синхронизация вкладок.
- Поддержка offline-режима.

Планы: open-source

Некоторые практические упражнения могут превратиться в open-source-библиотеки.

Пример: парсинг и манипулирование URL.

План: трекинг

- Кто на каком уровне?
- Кто сдавал последним middle-1?

План: интернационализация

Уровни старше middle-2 уже создаются на английском.

Саммари

Глава 4

Саммари

- Начинайте с простого.

Саммари

- Начинайте с простого.
- Заимствуйте.

Саммари

- Начинайте с простого.
- Заимствуйте.
- Ищите пассионарных к развитию энтузиастов.

Саммари

- Начинайте с простого.
- Заимствуйте.
- Ищите пассионарных к развитию энтузиастов.
- Собирайте фидбэк у команды.

Саммари

- Начинайте с простого.
- Заимствуйте.
- Ищите пассионарных к развитию энтузиастов.
- Собирайте фидбэк у команды.
- Повышайте эффективность с P2P-обучением.

Саммари

- Начинайте с простого.
- Заимствуйте.
- Ищите пассионарных к развитию энтузиастов.
- Собирайте фидбэк у команды.
- Повышайте эффективность с P2P-обучением.
- Формализуйте социальный лифт.

Саммари

- Начинайте с простого.
- Заимствуйте.
- Ищите пассионарных к развитию энтузиастов.
- Собирайте фидбэк у команды.
- Повышайте эффективность с P2P-обучением.
- Формализуйте социальный лифт.
- Склоняйтесь в сторону ошибок первого рода.

Как мы создавали карту развития для разработчиков



[Карта развития на Github](#)

Черепанов Сергей

Cofounder FSD, cofounder & CTO ptichka.moscow

<https://t.me/znack>, sergey@fullstack-development.com

Потенциальные вопросы

1. Почему после интервью проще запоминать материал?
2. Что сеньор может спроектировать, а мидл нет?
3. Если карта изменилась, а мидл уже сдал уровень?
4. А если в компании всё строго с планированием времени?
5. А если кому-то хочется глубоко закопаться в узкую область?
6. Как помогать команде резко прокачать конкретные знания по какой-то теме по текущему заданию?

Доступные карты развития/ матрицы компетенций

Глава пасхальная

Programmer Competency Matrix

Самый гуглабельный
системный обзор

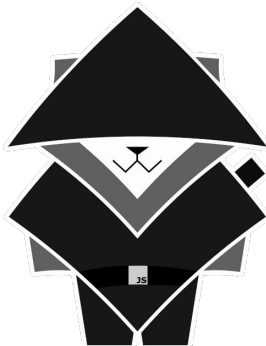


[Источник](#)

- Computer Science
- Code organization
- API design
- Language Paradigms
- Testing
- Problem decomposition
- Communications
- ...

Матрица от JavaScript.Ninja

Сообщество Javascript-разработчиков родом из Харькова



[Источник](#)

- Специфичные технологии
 - понимание принципов работы движка JavaScript;
 - знание нюансов отрисовки слоёв в браузере;
 - методы математической оценки сложности алгоритмов;
 - ...
- Концептуальные знания
 - SOLID;
 - DDD;
 - как совмещать ООП и ФП;
 - как грамотно логировать;
 - как грамотно кэшировать;
 - ...

Hexlet

Краткая матрица с мощными требованиями



- Специфичные знания
 - работа сетей
 - инициализация в *nix-мирах
 - ...
- Концептуальные знания
 - SOLID;
 - DDD;
 - виды полиморфизмов;
 - теория графов, теория категорий;
 - ...

Front-End Developer Handbook 2018

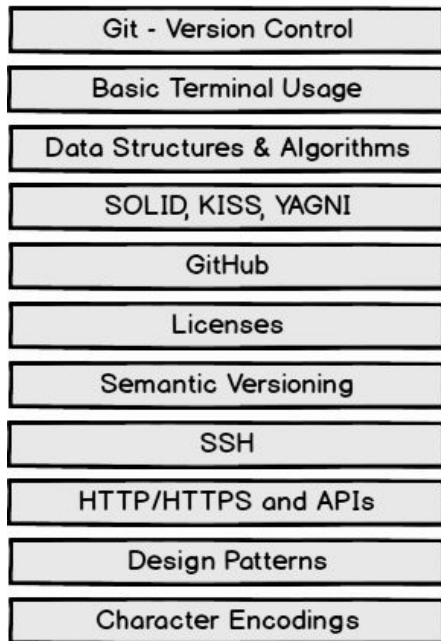
Много фронтальной специфики и поверхностно проработаны концептуальные вещи

Frontend *Masters*

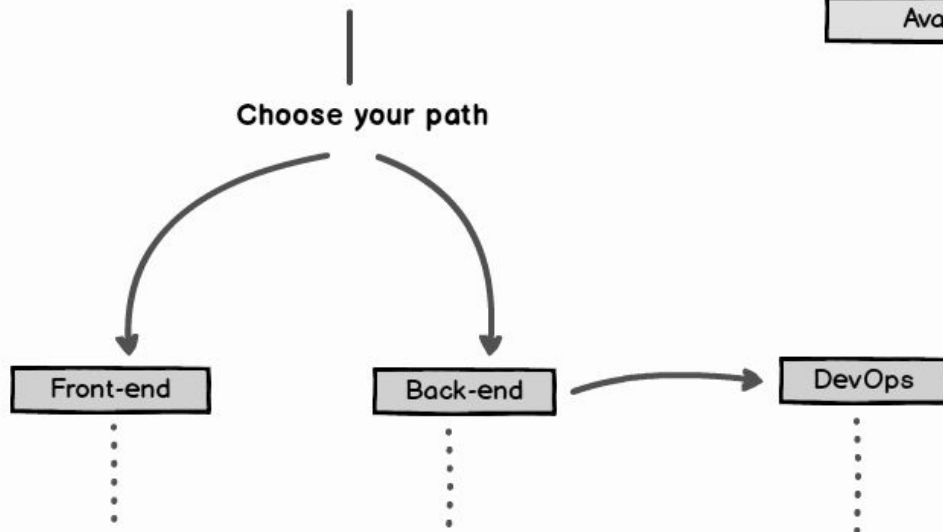
- HTML/CSS/JS.
- SEO, Accessibility, Progressive Web Apps.
- Build and Task Automation.
- Testing.
- HTTP/Networks.
- ...

Web developer roadmap

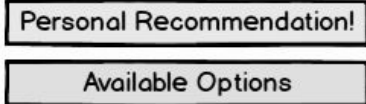
Required for any path



Web Developer in 2019



Legends



Матрица от CircleCI

Гайд и матрица по уровням навыков



[Источник](#)

Крайне хорошо проработаны
Soft Skills.

Базово проработаны
концептуальные вещи,
наподобие дебагинга или
безопасности.

Medium Engineering Growth Framework

*Серия статей: Матрица по уровням навыков.
Инструмент визуализации*



Методология с глубокой проработкой самого формата развития.

Много внимания уделено Soft Skills.

[Источник](#)

Grow.by OT EPAM

HARD SKILLS

SOFT SKILLS

ENGLISH

L1

L2

L3

L4

L5

FRONT-END AND JAVASCRIPT, JAVASC...

Skills

Expected

Software

−

Craftmanship and Development Process

...

Software

−

Engineering Practices

...

Code Standards

Novice

...

Code Review Process

Novice

...

Core JavaScript

Expected for selected level:

Arrays

▼

Novice

Closures Basics

^

Novice

Success criteria:

– Nested functions

– Able to create nested functions