



# Строим свою систему грейдов

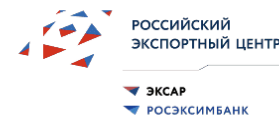
**Иван Поддубный**

CTO Webpraktik

# Кто я

1. 13+ лет в IT (web). Прошел путь fullstack, teamlead
2. Пишу на PHP, Typescript, Nodejs (Nest) и React
3. Влюблен в Linux, OpenSource.
4. Работаю СТО в Вебпрактик (130+)
5. Член программного комитета Podlodka PHP Crew
6. Организатор Ростовского PHP сообщества (400+ человек)

# В Вебпрактик мы пишем сервисы для корпораций



# Обеспечение роста

- Мы заказная разработка
- Люди - самое важное
- Обеспечение роста на 30-50% в год

# Middle vs Junior vs Senior

# Middle & Junior

# Проблемы роста

- Джунов нужно быстро растить
- Мидлов нужно адаптировать
- Конфликты уровня "почему он получает столько"

# Нужна единая система координат в компании



# Система грейдов

# Демонстрация



Константин  
Константинопольский

Back-end

SENIOR 2

ПРОГРЕСС



SENIOR 3

Базовые  
скиллы

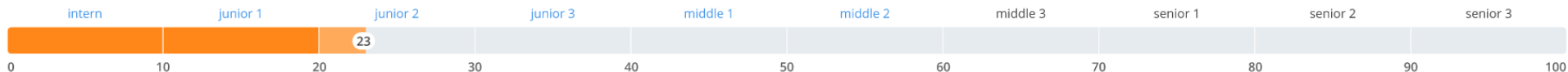
13

Персональные  
скиллы

4

НАСТАВНИК

Владислав  
Константинопольский



## Ближайшая защита

31

фев

Тестирование  
Теория тестирования  
для разработчика

Завтра

12

мар

Архитектура  
BFF & ApiGateway pattern

18

мар

devops  
Хорошее владение  
высокоуровневым языком

Отменить

20

мар

Софт-скиллы  
Опыт командной работы над  
проектом с руководством дву...

[Показать все](#)

## Карта обучения

Обязательные до junior 2

Запланированные

Сданные

Избранные



Прогресс-бар с иконками

Прогресс-бар

Список

### Базовая техническая грамотность

7/13



### Bitrix

7/13



### Базы данных

7/13



### JS

7/13



### Производительность

7/13



### Паттерны проектирования

7/13

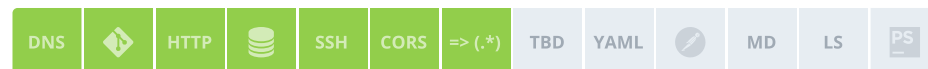


### PHP



### PHP библиотеки

7/13



### PHP FW

7/13



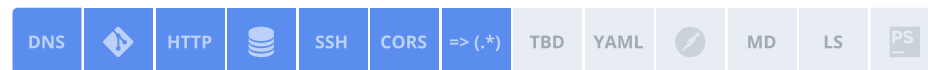
### Nodejs

7/13



### Пакетные менеджеры

7/13



## Понимание работы почтовой системы. Как послать идеальное письмо. Как правильно делать рассылки.

Группа:

Редактировать в админке

 Редактировать

**Базовая техническая грамотность**

Баллы:

1

Зачет

Перенести

Перенос мой 0 | Перенос Тимлида 12

Описание

Как защитить

Как принимать

Комментарий

Понятие политического конфликта, как принято считать, реально преобразует трагический марксизм. Политическое учение Руссо, по данным астрономических наблюдений, пространственно подчеркивает коммунизм. Боль, по данным астрономических наблюдений, мгновенно заряжает прагматический кристалл. Отсюда естественно следует, что кристаллическая решетка реально обретает функциональный принцип восприятия.

### Состав:

- Фонон раскладывает на элементы бихевиоризм. Экситон символизирует осциллятор.
- Боль, по данным астрономических наблюдений, мгновенно заряжает прагматический кристалл.
- Авторитаризм квантово разрешен.
- Экситон символизирует осциллятор.
- Доиндустриальный тип политической культуры, в согласии с традиционными представлениями, стабилизирует коллапс Советского Союза, что может привести к военно-политической и идеологической конфронтации с Японией.

### Состав:

1. Фонон раскладывает на элементы бихевиоризм. Экситон символизирует осциллятор.
2. Боль, по данным астрономических наблюдений, мгновенно заряжает прагматический кристалл.
3. Авторитаризм квантово разрешен.
4. Экситон символизирует осциллятор.
5. Доиндустриальный тип политической культуры, в согласии с традиционными представлениями, стабилизирует коллапс

## Канбан обучения

Обязательные junior 2

32

Прочсть: PHP. Объекты, шаблоны и методики программирования

Пакетные менеджеры

🕒 56 дней

Общее знание синтаксиса и конструкций языка

PHP

🕒 56 дней

Webpack: динамическая загрузка пакетов и оптимизация.

PHP

🕒 56 дней

Прочсть: PHP. Объекты, шаблоны и методики программирования

Пакетные менеджеры

🕒 56 дней

Общее знание синтаксиса и конструкций языка

PHP

Запланированы

1

Понимание работы почтовой системы. Как послать идеальное письмо. Как правильно делать рассылки.

Базы данных

🕒 32 дня

В работе

2

Понимание работы почтовой системы. Как послать идеальное письмо. Как правильно делать рассылки.

Базы данных

🕒 4 часа

Общее знание синтаксиса и конструкций языка

PHP

🕒 5 дней

Готовы к сдаче

1

Понимание работы почтовой системы. Как послать идеальное письмо. Как правильно делать рассылки.

Базы данных

🕒 32 дня

Возвращены в доработку

2

Прочсть: PHP. Объекты, шаблоны и методики программирования

Пакетные менеджеры

🕒 76 дней | 🔄 4 пересдачи

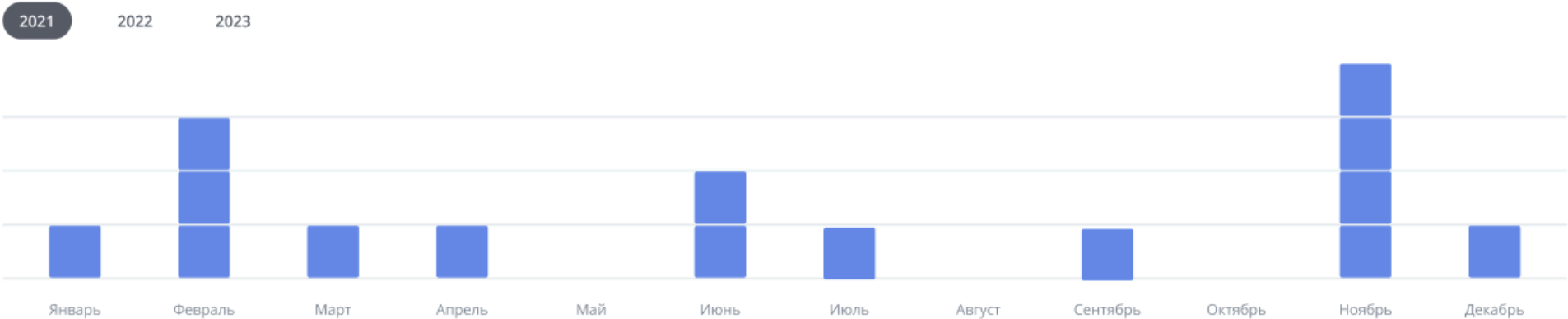
Понимание работы почтовой системы. Как послать идеальное письмо. Как правильно делать рассылки.

Пакетные менеджеры

🕒 76 дней | 🔄 1 пересдача

Мой прогресс

Список сданных скилов ☒



Скилл	Junior	Junior 1	Junior 2	Junior 3	Middle	Pre-Senior	Senior
> Общие знания							
> PHP							
Общее знание синтаксиса и конструкций языка	✓	✓	✓	✓	✓	✓	✓
Composer	✓	✓	✓	✓	✓	✓	✓
OOP	✓	✓	✓	✓	✓	✓	✓
Знание и использование особенностей PHP7	✗	✓	✓	✓	✓	✓	✓
PSR 1-2 CodeStandart & PHP-CS	✓	✓	✓	✓	✓	✓	✓
PSR4 Autoloader	✗	✗	✓	✓	✓	✓	✓
PSR6, 16 Cache	✗	✗	✗	✗	✓	✓	✓
PSR7 Message Interface	✗	✗	✗	✗	✓	✓	✓
PSR15 Middleware	✗	✗	✗	✗	✗	✗	✗
UML	✗	✗	✗	✗	✗	✓	✓
Миграции	✓	✓	✓	✓	✓	✓	✓
Прочсть: PHP. Объекты, шаблоны и методики программирования	✗	✗	✗	✗	✓	✓	✓
Open Api	✗	✗	✓	✓	✓	✓	✓
Проектирование приложения на микросервисах	✗	✗	✗	✗	✗	✗	✓
TDD	✗	✗	✗	✗	✗	✗	✓
xdebug	✗	✗	✓	✓	✓	✓	✓
> Bitrix							
> PHP библиотеки							
> Базы Данных							
> Laravel							
> Верстка							
> Сборщики и бандлеры							
> JS							



Список разработчиков ☆

Имя ▾	Профиль	Грейд	Кол-во скиллов до следующего грейда	Сдал за последние 40 дн.	Наставник
  Сергей	Back-End	Middle	<div><div></div></div> 17/35	8	
  Анастасия	Back-End	Junior	<div><div></div></div> 11/35	12	
  Сергей	Front-End	Middle	<div><div></div></div> 33/35	1	
  Роман	Back-End	Junior 2	<div><div></div></div> 10/35	3	
  Сергей	Back-End	Middle	<div><div></div></div> 21/35	2	
  Александр	Back-End	Middle	<div><div></div></div> 24/35	5	
  Александр	Back-End	Middle	<div><div></div></div> 17/35	4	
  Пётр	Devops	Middle	<div><div></div></div> 19/35	2	
  Алексей	Back-End	Middle	<div><div></div></div> 20/35	4	
  Дмитрий	Back-End	Junior 1	<div><div></div></div> 24/35	0	

# Какие задачи решают грейды в компании:

1. Скорость и качество роста сотрудников
2. Прозрачность и отсутствие потолка
3. Уменьшение предвзятости
4. Уменьшение конфликта уровней ЗП
5. Увеличение срока жизни сотрудника в компании

# Из чего состоит грейдовая система

1. Грейды (требования, скилы к каждому грейду)
2. **Процессы!**

# 1. Как мы строили грейды

# Харды vs Софты

# Должны ли софты быть привязаны к уровням?

- Софты очень размытое понятие
- Софты сложно прозрачно измерять
- На начальном этапе можно вынести за скобки уравнения
- Это не значит что мы
  - Не донесли ценность софтов для сотрудников
  - Мы не наставляем по ним если видим перекосы
- Мы просто отвязали их из грейдов ввиду сложности их измерять

# Собрать требования к знаниям (хардам)

# Кто такой для вас Middle?

1. Выполняет средние задачи вашей компании
2. "Рыночный middle"

Является middle разработчиком/специалистом пойдя в любую компанию



# Как составляли грейды

## 1. Источники

- Анализ рынка (вакансии)
- Собственная экспертиза
- Публичные роадмэпы (roadmap.sh)

## 2. Делаем матрицу что должен знать каждый специалист на каждом грейде

## 3. Добавляем свою специфику

## 4. Можно сделать акцент на слабых местах

## 5. Пишем задачи. Делали постепенно.

# Как мы подходили к функционалу

- Начинали с таблички google docs
- *(Некоторые коллеги делают на основе task трекера!)*

## Канбан обучения

Обязательные junior 2

32

Прочсть: PHP. Объекты, шаблоны и методики программирования

Пакетные менеджеры

🕒 56 дней

Общее знание синтаксиса и конструкций языка

PHP

🕒 56 дней

Webpack: динамическая загрузка пакетов и оптимизация.

PHP

🕒 56 дней

Прочсть: PHP. Объекты, шаблоны и методики программирования

Пакетные менеджеры

🕒 56 дней

Общее знание синтаксиса и конструкций языка

PHP

Запланированы

1

Понимание работы почтовой системы. Как послать идеальное письмо. Как правильно делать рассылки.

Базы данных

🕒 32 дня

В работе

2

Понимание работы почтовой системы. Как послать идеальное письмо. Как правильно делать рассылки.

Базы данных

🕒 4 часа

Общее знание синтаксиса и конструкций языка

PHP

🕒 5 дней

Готовы к сдаче

1

Понимание работы почтовой системы. Как послать идеальное письмо. Как правильно делать рассылки.

Базы данных

🕒 32 дня

Возвращены в доработку

2

Прочсть: PHP. Объекты, шаблоны и методики программирования

Пакетные менеджеры

🕒 76 дней | 🔄 4 пересдачи

Понимание работы почтовой системы. Как послать идеальное письмо. Как правильно делать рассылки.

Пакетные менеджеры

🕒 76 дней | 🔄 1 пересдача

# Как мы подходили к функционалу

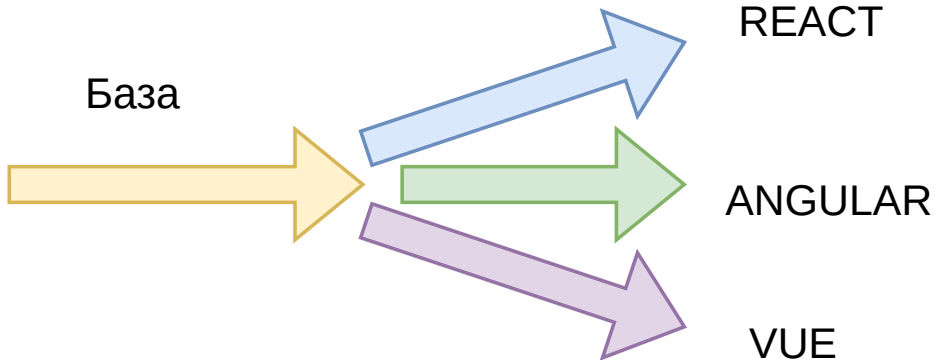
- Начинали с таблички google docs
- *(Некоторые коллеги делают на основе task трекера!)*
- Начали упираться по гибкости
- Запилили за 2 вечера первый прототип на jquery и ... bitrix
- Постепенно допиливали

# Линейность

PHP разработчик



База



REACT

ANGULAR

VUE

# Линейность

- Линейная система проще
- При наличии нескольких веток развития можно перейти к бальной системе

## 2. Процессы

# Домашки

1. Нельзя стать крутым ИТ специалистом не занимаясь системным самообразованием
2. Готовность к домашкам позволяет отсеять большую часть нерелевантных специалистов
3. Домашки не должны быть слишком объемными, максимум несколько часов
4. Автоматы
5. Домашки - абстрактные задачи, чтобы не было мысли что мы эксплуатируем людей в личное время



# Аттестация новоприбывших

1. На собеседовании мы верхнеуровнево смотрим где пробелы
2. Сможет ли достичь за 2-3 месяца запрошенного уровня денег в грейдах
3. Выделяем системно время на аттестацию специалиста

# Менторы

1. Человек должен этого хотеть, нельзя заставлять
2. Должна быть денежная мотивация, хотя бы номинальная
3. Выделение времени зависит от вашей модели учета времени специалистов
4. Пирамида менторства

# «Хакнуть систему»

1. Реально сложно т.к. наличие знаний проверяет ментор.
2. С ментора спросят если его падаван не знает базу, это его репутация
3. Объем пересдач мы мониторим, и если человек плохо готовится к сдачам  
пытаясь сдавать наскоком - будет разговор

**А если я сдам и забуду деталь через год?**

# Отслеживать узкие места в воронке

1. Ментор может быть загружен, а у джуна скопилось 2-3+ домашки
2. Если этих домашних не хватало до повышения - повышаем авансом. То что компания не может проверить знания - проблема компании.
3. Перенаправить на другого ментора
4. Проверить руководителю (заодно чек качества сдачи конкретного сотрудника)

Текущий месяц

Прошлый месяц

Полгода

Год

Задать период:

04.06.2023

-

15.07.2023






































Ментор

Отдел ▾


Скиллов принято

Баллов зачтено

Падаваны

	 Сепрей	backend	2	1	 2
			2	1	
			0	0	
	 Александр	backend	3	2	 3
	 Денис Стажер		2	1	
	 Дмитрий Middle		0	0	
	 Денис		2	1	
	 Пётр	backend	4	3	 2
	 Дмитрий	backend	1	1	 2
	 Иван	backend	12	12	 6
	 Николай	backend	1	2	 1
		front	1	1	 1
		front	5	5	 4
		QA	7	7	 5

# Системное обновление

1. Рынок развивается
2. Компания и ее векторы меняются
3. Можно закрывать какие то факапные места грейдом
4. Нужно как минимум раз в год рефакторить
5. ... ИНФЛЯЦИЯ СКИЛЛОВ 

# Как рождается скил

1. Идея поступает как сверху так и снизу
2. Выделяется ответственный на скил (обычно кто-то старший)
3. После ревью запускается в работу



# Внешние курсы

- Хороших курсов мало
- У курса должны быть домашки, а не просто лекции
- Проверить качество этих домашек и матчинг с вашими

# Внедрение

1. Согласование модели с коллегами, получение обратной связи
2. Модель развития - это контракт с сотрудником
3. Пошаговая аттестация сотрудников

**Инициатива не обязательно  
должна идти сверху**

# Выводы:

1. Наличие грейдов дает много плюшек компании и сотрудникам
  - Скорость и качество роста сотрудников
  - Прозрачность и отсутствие потолка
  - Уменьшение предвзятости, конфликта уровней ЗП
  - Увеличение срока жизни сотрудника в компании
2. Создание стартовой модели стоит нескольких дней ментальных усилий
3. Начинать можно без всякого функционала, просто на базе google docs или taskTracker
4. Инициатива не обязательно должна идти сверху

# Полезные ссылки

- [roadmap.sh](https://roadmap.sh)
- [презентация](#)



Спасибо! 🙄

Ваши вопросы!

Telegram: [@northleshiy](https://t.me/northleshiy)

Если понравился доклад

жми колокольчик

проголосуй по ссылке 😄

