

Матрица компетенций

и оценка Java-разработчиков

Александр **Логинов**

12 лет

в разработке

5 лет

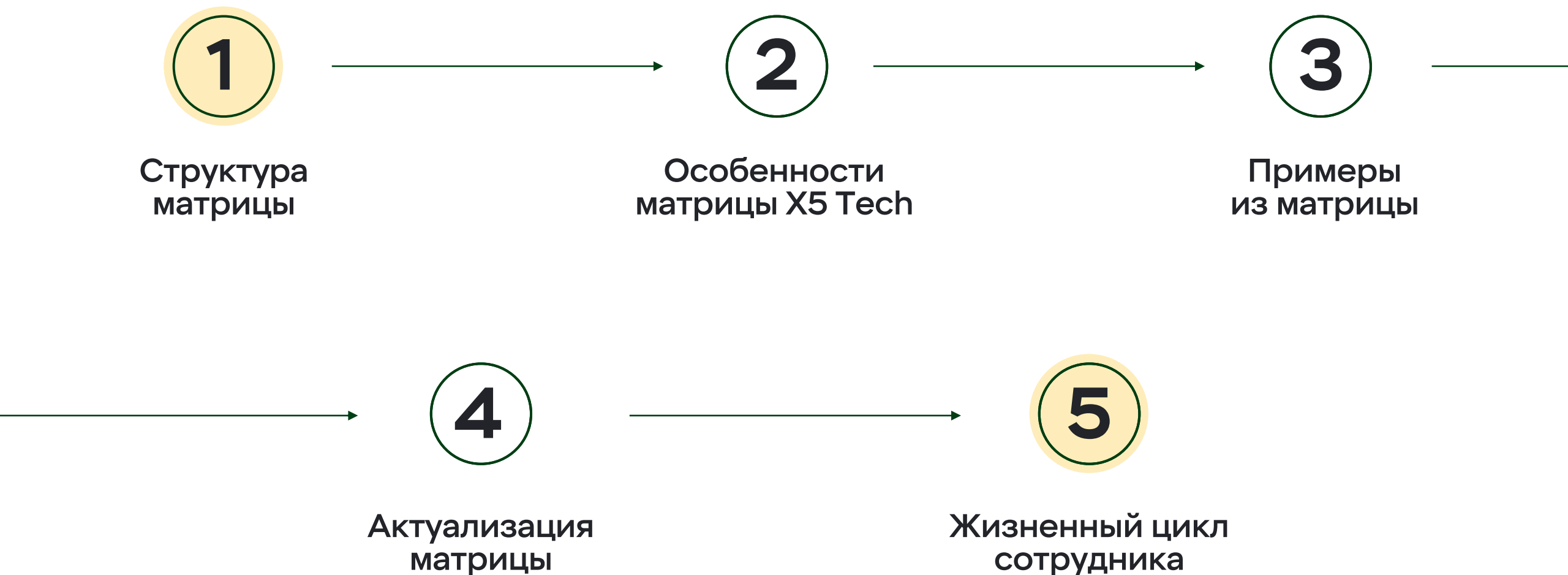
Работаю с людьми

3 года

В X5 Tech



О чем поговорим



Зачем матрица компетенций?



Понять **уровень** сотрудника



Принять решение о **повышении**



Сделать **развитие** прозрачным для сотрудника



Сделать оценку **справедливой**



Полу-милордсиньор, полу-купец,
Полу-мудрец, полу-невежда,

ПРОСТАЯ МАТРИЦА

Грейд	Что должен уметь
Junior	<ul style="list-style-type: none">✓ Знает структуры данных✓ Умеет отлаживать код✓ Умеет писать запросы в БД
Middle	<ul style="list-style-type: none">✓ Умеет настраивать сборку проекта в maven✓ Знает примитивы синхронизации
Senior	<ul style="list-style-type: none">✓ Умеет применять ExecutorService✓ Использует Docker Compose в локальной разработке

ПРОСТАЯ МАТРИЦА



Легка в составлении



Неформализована



Сложно построить
путь сотрудника

РАБОЧАЯ МАТРИЦА

Навык	Basic	Good	Strong
Java	Знает структуры данных	Знает примитивы синхронизации	Умеет применять ExecutorService
Сборка	Знает фазы maven	Умеет управлять зависимостями и решать конфликты	Умеет полностью настраивать сборочный цикл приложения
Docker	Знает команды ps, run, exec	Умеет писать Dockerfile и Docker Compose	Знает устройство volume, network

РАБОЧАЯ МАТРИЦА

Навык	Basic	Good	Strong
Java	Знает структуры данных	Знает примитивы синхронизации	Умеет применять ExecutorService
Сборка	Знает фазы maven	Умеет управлять зависимостями и решать конфликты	Умеет полностью настраивать сборочный цикл приложения
Docker	Знает команды ps, run, exec	Умеет писать Dockerfile и Docker Compose	Знает устройство volume, network

РАБОЧАЯ МАТРИЦА

Навык	Basic	Good	Strong
Java	Знает структуры данных	Знает примитивы синхронизации	Умеет применять ExecutorService
Сборка	Знает фазы maven	Умеет управлять зависимостями и решать конфликты	Умеет полностью настраивать сборочный цикл приложения
Docker	Знает команды ps, run, exec	Умеет писать Dockerfile и Docker Compose	Знает устройство volume, network

РАБОЧАЯ МАТРИЦА

Навык	Basic	Good	Strong
Java	Знает структуры данных	Знает примитивы синхронизации	Умеет применять ExecutorService
Сборка	Знает фазы maven	Умеет управлять зависимостями и решать конфликты	Умеет полностью настраивать сборочный цикл приложения
Docker	Знает команды ps, run, exec	Умеет писать Dockerfile и Docker Compose	Знает устройство volume, network

РАБОЧАЯ МАТРИЦА

Навык	Junior	Middle	Senior
Java	Basic	Good	Strong
Сборка	Basic	Good	Strong
Docker	N\A	Basic	Good

РАБОЧАЯ МАТРИЦА

Навык	Junior	Middle	Senior
Java	Basic	Good	Strong
Сборка	Basic	Good	Strong
Docker	N\A	Basic	Good

РАБОЧАЯ МАТРИЦА

Навык	Junior	Middle	Senior
Java	Basic	Good	Strong
Сборка	Basic	Good	Strong
Docker	N\A	Basic	Good

РАБОЧАЯ МАТРИЦА

Навык	Junior	Middle	Senior
Java	Basic	Good	Strong
Сборка	Basic	Good	Strong
Docker	N\A	Basic	Good

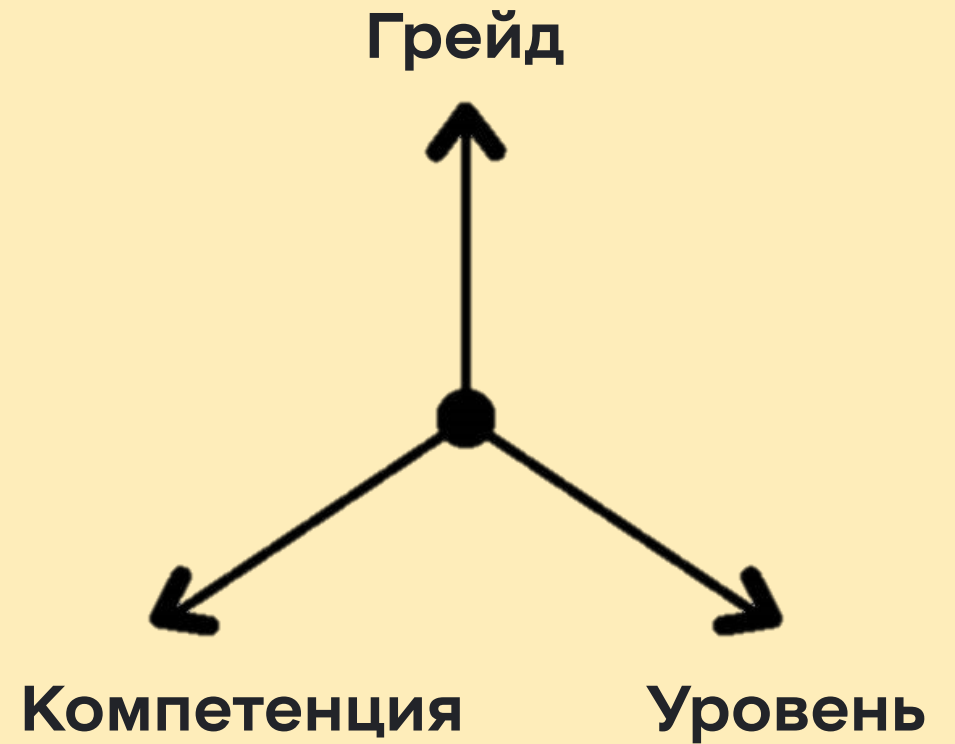
РАБОЧАЯ МАТРИЦА



Достаточно
формализована



Позволяет построить
путь сотрудника



X5 Tech



IT компания



Центры компетенций
и домены

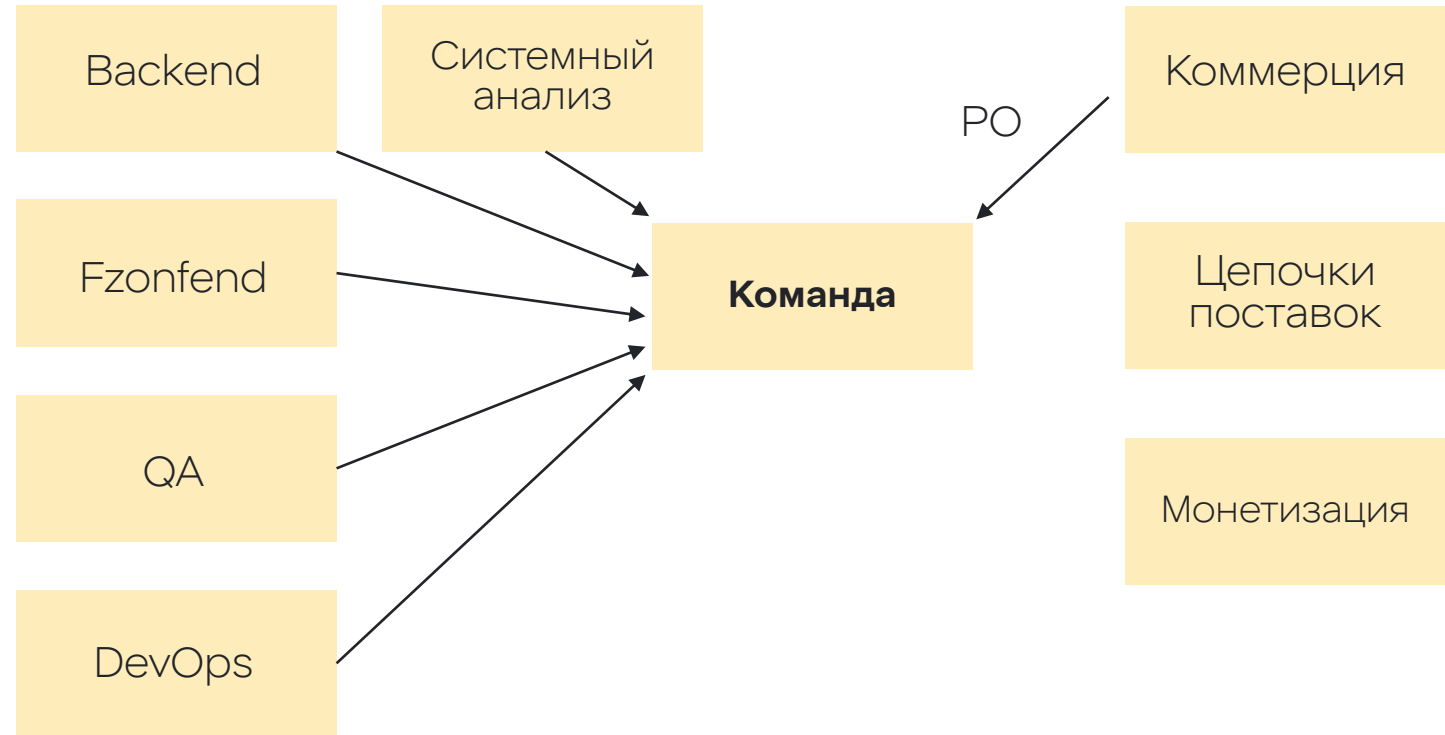


Общий штат ~ 5000



200+ backend
разработчиков

Центры компетенций



X5 Tech - уровни

Старший разработчик



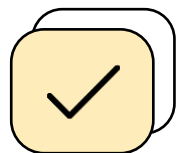
Разработчик



Младший разработчик

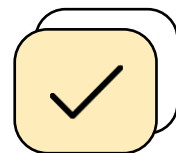


Матрица X5 Tech

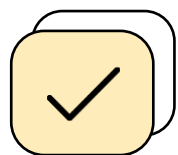


9 грейдов

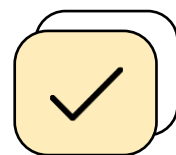
от jun- до sen+



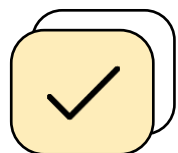
На каждом уровне набор
критериев



**Соответствие
грейд-компетенция**



Критерии **дискретны**



9 уровней владения

от jun- до sen+



Аналог bit-mask

Легенда работы с таблицами

Компетенция	jun-	jun	jun+	Набор критериев
	1			
		1		
			1	

Компетенция	mid-	mid	mid+	Набор критериев
	1			
		1		
			1	

Таблица № 1

Java Core	jun-	jun	jun+	
	1			Понимает синтаксис Java (class, abstract class, interface, record*)
		1		Понимает отличия между классами-обертками и примитивами
		1		Понимает контракт equals/hashCode и знает варианты применения этих методов в Java Core
			1	Знает, что такое enum, и понимает его отличия от констант

Таблица № 2

Java Core	mid-	mid	mid+	
		1		Понимает иерархию Throwable (checked/unchecked, exception/error)
	1			Понимает иерархию коллекций Java
		1		Применяет в работе обобщения в классах и коллекциях
	1			Понимает работу java string pool, java integer const pool
			1	Умеет применять битовые операции

Таблица № 3

Java Concurrency	mid-	mid	mid+	
		1		Понимает базовые примитивы синхронизации (java.util.concurrent.atomic, Collections.synchronized*, synchronized, Lock)
		1		Умеет читать чужой код, содержащий примитивы синхронизации, и понимать, что он делает
		1		Может рассказать, как потоки работают с общими объектами
		1		Имеет представление про volatile/final/synchronized при работе с Java Memory Model
			1	Понимает и может рассказать, что такое compare-and-swap (CAS), reordering, happens-before

Таблица № 4

Java Concurrency	sen-	sen	sen+	
		1		Использует CompletableFuture, понимает отличия в реализациях Future / CompletableFuture
		1		Применяет сторонние неблокирующие фреймворки (RX, Reactor, Cats, FS2)
			1	Умеет оптимизировать конкурентный код
			1	Понимает модель исполнения неблокирующих фреймворков
			1	Понимает особенности распределения памяти в многопоточной среде

Таблица № 5

Spring Core	jun-	jun	jun+	
		1		Понимает базовые строительные объекты Spring: bean, BeanDefinition, Configuration, BeanPostProcessor, BeanFactoryPostProcessor, BeanFactory/ApplicationContext
		1		Умеет конфигурировать приложение через файлы настроек
		1		Знает основные scope bean-ов (singleton, prototype) и их отличия
		1		Понимает в общих чертах об инверсии зависимостей и DI
		1		Знает ЖЦ bean и методы управления им: @PostConstruct, @PreDestroy или аналоги на интерфейсах (InitializingBean, DisposableBean)

Таблица № 6

Spring Core	mid-	mid	mid+	
	1			Использует (или хотя бы знает) про профили приложения
		1		Знает как решать циклические зависимости
	1			Знает какие методы DI есть и какие лучше использовать
		1		Понимает что такое контекст и принцип наследования/ограничения контекстов
		1		Знает как работает Enable-аннотации, понимает концепцию экспорта конфигурации

Таблица № 7

Kafka	mid-	mid	mid+	
	1			Общие знания об инструменте, его концепции и возможностях, отличиях от других инструментах Понимает принципы работы лога данных и его отличия от очереди
	1			Базовые знание основных концепций Kafka (topics, partitions, producers, consumers, consumers groups)
	1			Знания библиотек для работы с Kafka
		1		Умеет считать данные из лога и записать данные в лог, не обращая внимание на успешность обработки
		1		Знает, как масштабируются клиенты, подключающиеся к одному топику

Таблица № 8

Kafka	sen-	sen	sen+	
	1			Глубокие знание основных концепций Kafka (topics, partitions, producers, consumers, consumers groups)
	1			Проектирование топиков, распределения сообщений по ним
	1			Работа с кластером, обработка ошибок в кластерной конфигурации
			1	Знание параметров конфигурирования для настройки кластера
	1			Умеет обрабатывать повторное считывание сообщения из очереди, понимает причины возникновения таких ситуации и методы борьбы с ними

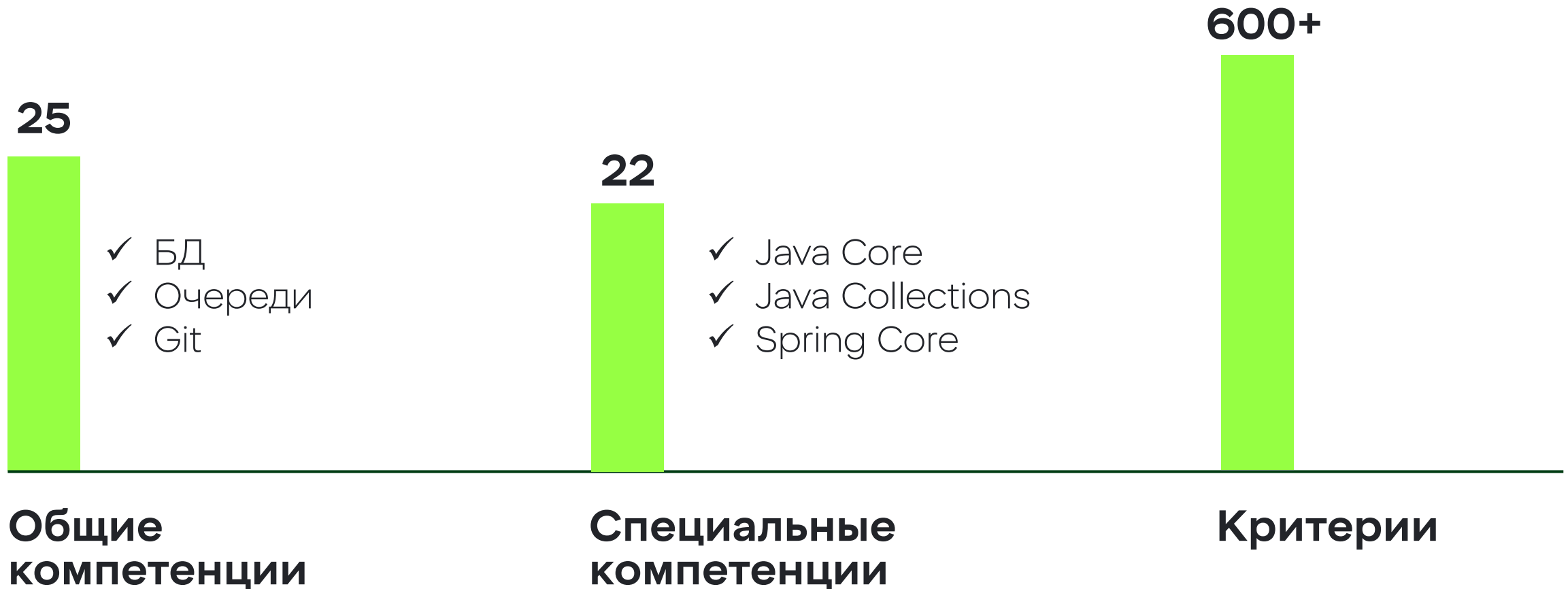
Таблица № 9

PostgreSQL	mid-	mid	mid+	
	1			EXPLAIN vs EXPLAIN ANALYZE Понимает стоимость исполнения SQL запросов для применяемой СУБД
		1		Transaction Isolation в PostgreSQL
	1			Устройство B-Tree индекса, алгоритмы
		1		Partial index
		1		Сложные запросы, подзапросы, CTE

Таблица № 10

PostgreSQL	sen-	sen	sen+	
	1			Внутреннее устройство (память, буферы, хранение на диске). Сбору статистики и использованию VACUUM
		1		Администрирование (поднять, настроить буферы, разрешенные подключения и тп), postgresql.conf, pg_hba.conf
	1			Служебный каталог и его назначение, статистика, планировщик запросов
		1		Знание psql, pg_dump, ... и их возможностей
		1		Конкурентное построение индексов

Матрица X5 Tech - статистика



Актуализация матрицы



Новые
технологии
появляются
каждый день



Существующие
технологии
развиваются



«Составил
и забыл»
нельзя!



Актуализация матрицы



**Добавляем/
убираем
компетенции**

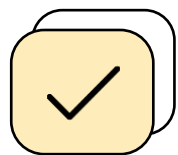


**Добавляем/
убираем
критерии**

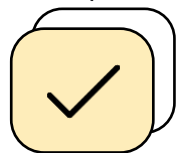


**Мигрируем
критерии**

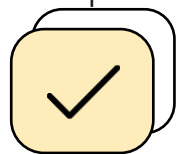
Источники пополнения матрицы



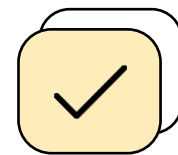
Компетенция попадает
в матрицу



Технология попадает
в список разрешенных



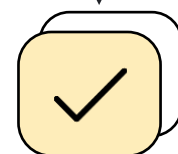
Команда использует
новую технологию



Следим за рынком



Исследуем
новые технологии



Пополняем матрицу

Матрица → Тех. радар

1

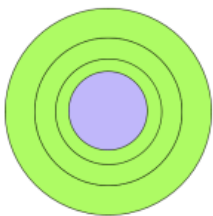
Поддерживается
архитекторами
X5 Tech

1

Регулярно
актуализируется
из центров
компетенций

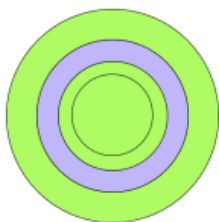
1

Содержит
не только back-end
технологии



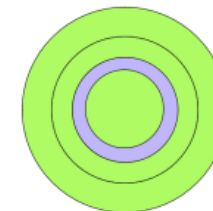
ADOPT

Технологии и инструменты, которые внедрены и активно используются в командах. Технологии, по которым в X5 Tech есть экспертиза



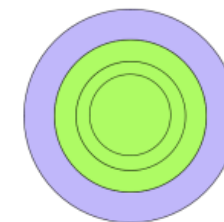
ASSESS

Пробные технологии и инструменты, которые в данный момент оцениваются. С их участием реализуются только тестовые проекты, и они не используются в реальных задачах



TRIAL

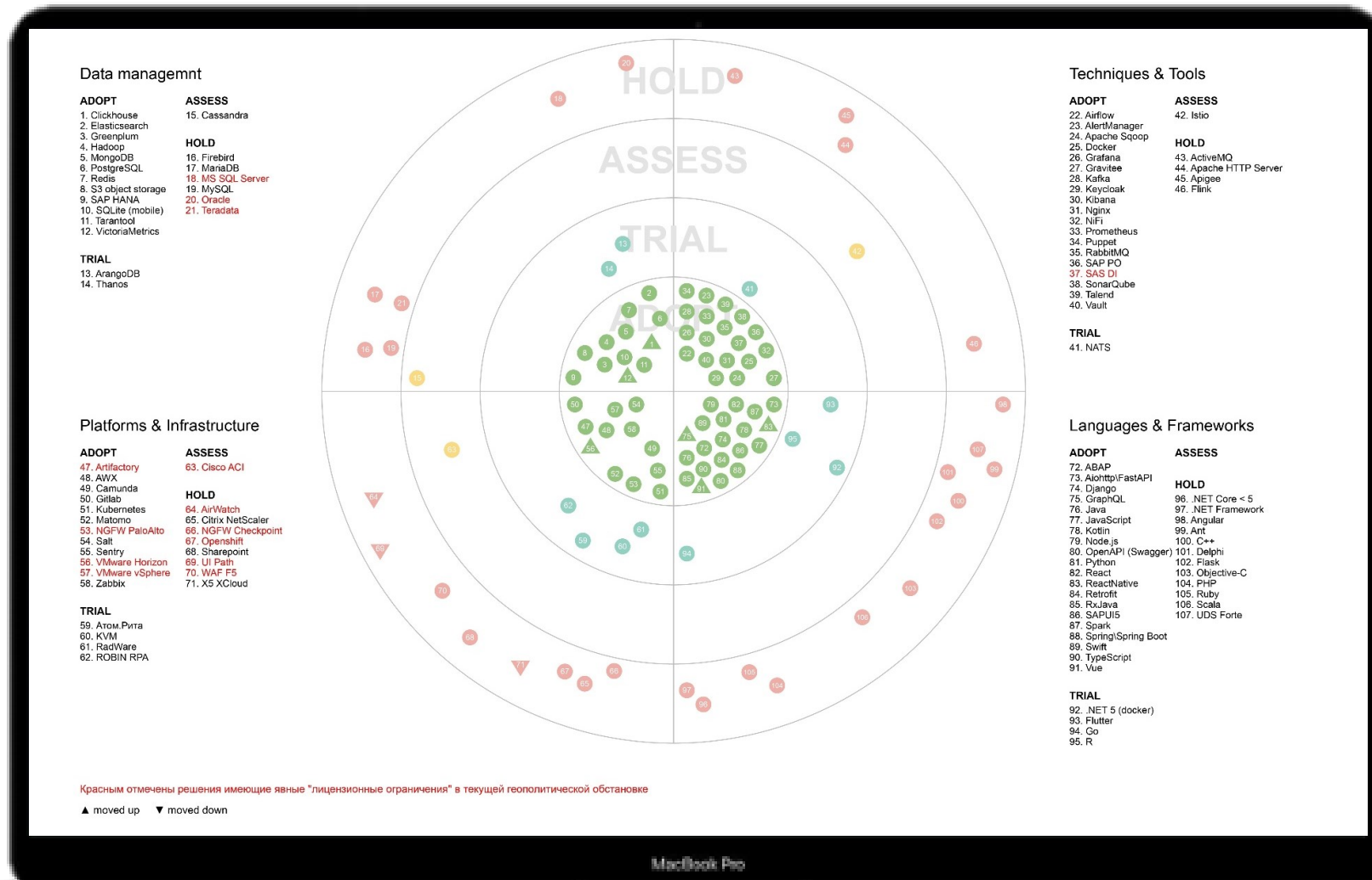
Технологии и инструменты, которые уже прошли этап тестирования и готовятся к тому, чтобы работать в продакшн (или даже уже работают там), но не имеют центров компетенции



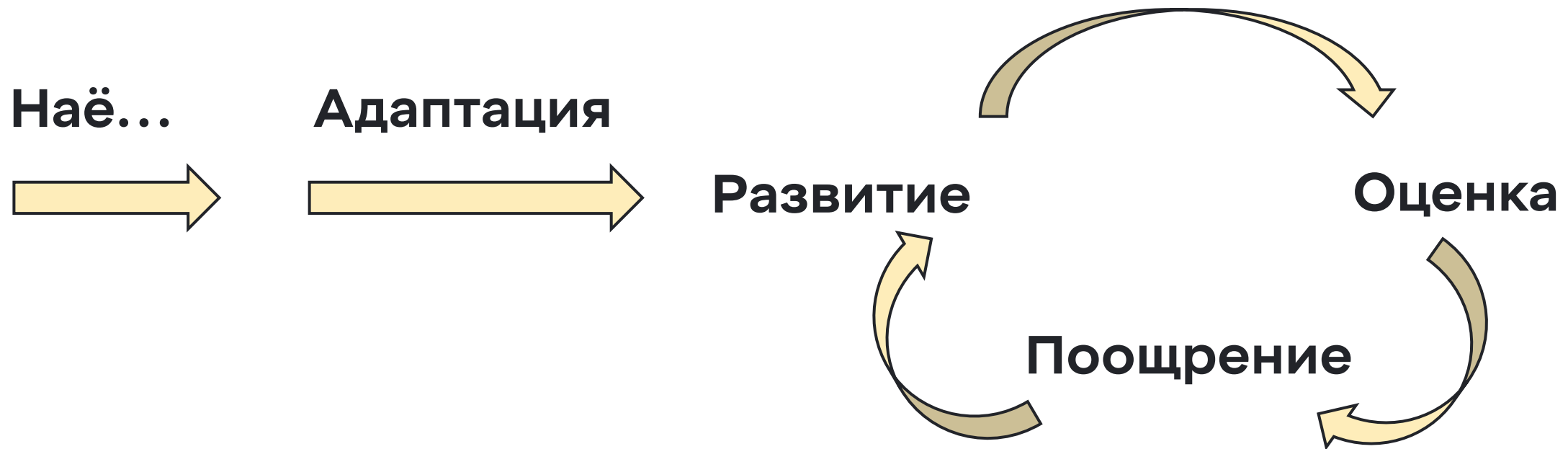
HOLD

В этой категории у нас есть экспертиза, но упомянутые инструменты используются при поддержке существующих систем — новые проекты на них не запускаются

Tech Radar – 2022.10 v3.4



Жизненный цикл сотрудника - обзор



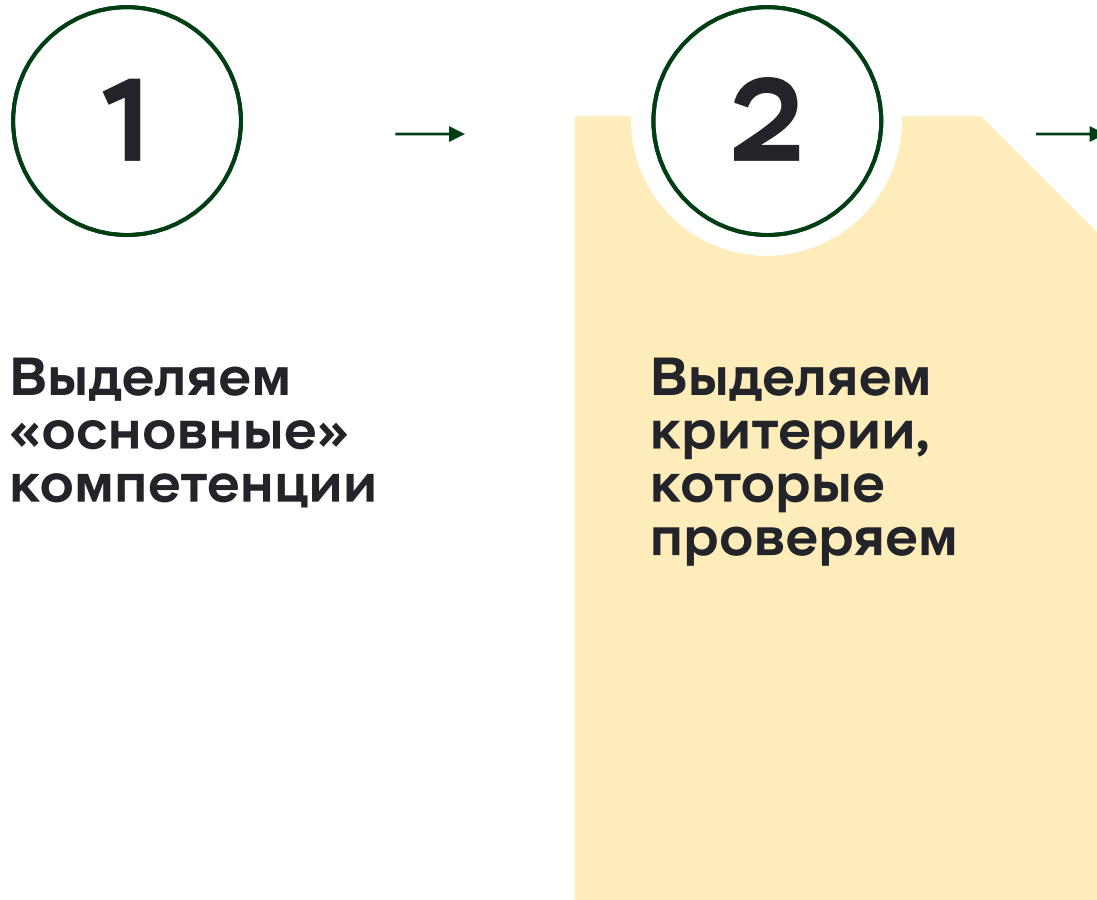
Жизненный цикл - наём

1



**Выделяем
«основные»
компетенции**

Жизненный цикл - наём



Жизненный цикл - наём

1



2



3



**Выделяем
«основные»
компетенции**

**Выделяем
критерии,
которые
проверяем**

**Составляем
перечень
вопросов**

Жизненный цикл - наём

1

Выделяем
«основные»
компетенции



2

Выделяем
критерии,
которые
проверяем



3

Составляем
перечень
вопросов



4

Проверяем
глубину
знаний

```
/**
 * Вопрос на группировку - кандидату необходимо выбрать наилучшую структуру под решение данной задачи и
 обосновать свой выбор
 */
public class Question7 {

    public interface Category {
        Long getId();
        String getName();
    }

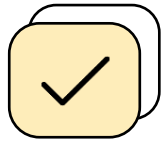
    public interface Product {
        Long getId();
        String getName();
        List<Category> getCategories();
    }

    /**
     * Метод вызывается 1 раз
     */
    public void init(List<Product> products, List<Category> categories) {
        //TODO implement
    }
}
```

```
/**
 * Проверяет принадлежность товара к категории
 * @param productId идентификатор товара
 * @param categoryId идентификатор категории
 * @return {@code true} если товар принадлежит к категории, в остальных случаях {@code false}
 */
public boolean isProductInCategory(Long productId, Long categoryId) {
    //TODO implement
    return false;
}

/**
 * Возвращает список товаров по категории. Товары должны быть отсортированы по идентификатору от
 * меньшему к большему
 * @param categoryId идентификатор категории
 * @return список товаров, принадлежащих к категории или пустой список, если к категории не
 * принадлежит ни один товар
 */
public Collection<Product> getProductsByCategory(Long categoryId) {
    //TODO implement
    return null;
}
}
```

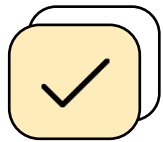
Жизненный цикл - наём



Выделяем «основные» компетенции



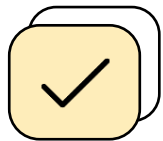
Проверяем глубину знаний



Выделяем критерии, которые проверяем



Выставляем уровень каждой компетенции

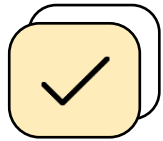


Составляем перечень вопросов

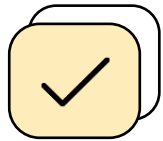


Задачи составлены на основе критериев

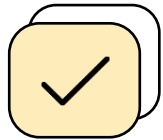
Жизненный цикл - наём



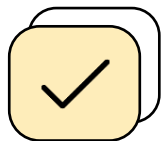
Выделяем «**основные**» компетенции



Выделяем критерии, которые **проверяем**



Составляем **перечень** вопросов



Задачи составлены **на основе** критериев



Проверяем **глубину** знаний

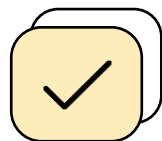


Выставляем уровень **каждой** компетенции



Формируем **итоговый** уровень кандидата

Жизненный цикл - наём



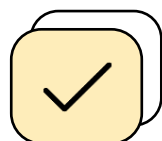
Выделяем «основные» компетенции



Выделяем критерии, которые проверяем



Составляем перечень вопросов



Задачи составлены на основе критериев



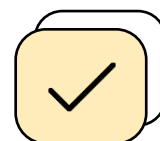
Проверяем глубину знаний



Выставляем уровень каждой компетенции



Формируем итоговый уровень кандидата



Все оценки в едином формате

Жизненный цикл - карточка сотрудника



Готова после
собеседования



Компетенции
из **матрицы**



Доступна
менеджеру
и сотруднику

Карточка сотрудника

Раздел компетенций	Категорий компетенций	Уровень владения
Java	Java*	
	Collections (ArrayList, LinkedList, HashMap, TreeMap...)	mid+
	Сборка мусора (общие принципы, поколения)	mid
	Работа с ресурсами (try-with-resource, IO, JDBC)	
	Lambda, Stream API	mid
	Многопоточка (потoki, синхронизация, volatile, ExecutorService...)	
	Spring*	
	Core (конфигурация, Scope, Qualifier, Primary...)	mid
	Web (Controller, RequestMapping, ExceptionHandler, ControllerAdvice...)	mid+
	Data (Transactional, Data JPA, Hibernate, OSIV)	mid-
Форматы данных	Форматы данных	jun
Базы Данных	Общая теория РСУБД*	mid+
	PostgreSQL*	mid
Клиент-серверное взаимодействие	HTTP, RESTful*	mid

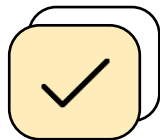
Карточка сотрудника

Раздел компетенций	Категорий компетенций	Уровень владения
Java	Java*	sen
	Collections (ArrayList, LinkedList, HashMap, TreeMap...)	sen
	Сборка мусора (общие принципы, поколения)	
	Работа с ресурсами (try-with-resource, IO, JDBC)	sen
	Lambda, Stream API	
	Многопоточка (потoki, синхронизация, volatile, ExecutorService...)	sen-
	Spring*	sen
	Core (конфигурация, Scope, Qualifier, Primary...)	sen
	Web (Controller, RequestMapping, ExceptionHandler, ControllerAdvice...)	sen-
	Data (Transactional, Data JPA, Hibernate, OSIV)	mid
Базы Данных	Общая теория РСУБД*	sen
	PostgreSQL*	mid

Карточка сотрудника

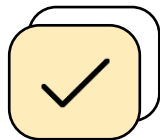
Раздел компетенций	Категорий компетенций	Уровень владения
Архитектура и шаблоны	Архитектура и шаблоны	sen
Клиент-серверное взаимодействие	API*	sen
	HTTP, RESTful*	mid+
Брокеры Сообщений	Общее понимание работы брокеров сообщений и очередей*	sen
	Kafka	mid
	RabbitMQ	mid
Инструменты	GitLab	
	Docker	mid+

Жизненный цикл - индивидуальный план

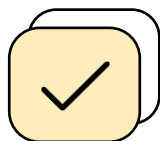


**Составляем
на несколько месяцев**

Жизненный цикл - индивидуальный план

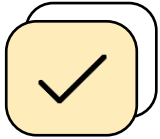


**Составляем
на несколько месяцев**

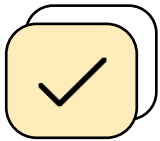


Компетенции À la carte

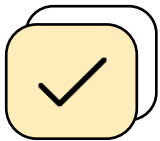
Жизненный цикл - индивидуальный план



**Составляем
на несколько месяцев**

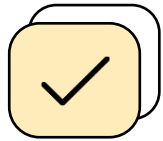


Компетенции À la carte

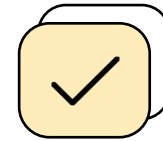


2-3 цели на изучение

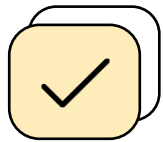
Жизненный цикл - индивидуальный план



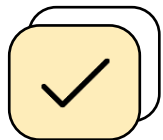
**Составляем
на несколько месяцев**



**Часть карточки
сотрудника**

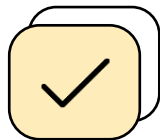


Компетенции À la carte



2-3 цели на изучение

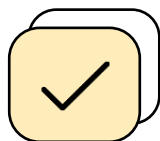
Жизненный цикл - индивидуальный план



**Составляем
на несколько месяцев**



**Часть карточки
сотрудника**

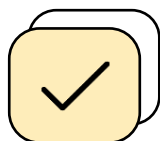


Компетенции À la carte



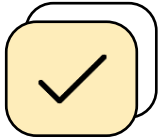
Обязательно

- ✓ Сроки
- ✓ Приоритет
- ✓ Критерии успешности
- ✓ Материалы



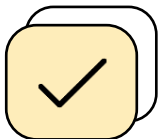
2-3 цели на изучение

Жизненный цикл - выполнение плана развития

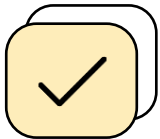


Сотрудник выполняет самостоятельно

- ✓ На рабочих задачах
- ✓ На разовых задачах другого продукта
- ✓ На pet-проекте



Помогает наставник



Прогресс оценивается на 1-1

Жизненный цикл - проблемы учебных материалов



**Не учитывают
специфику
X5 Tech**

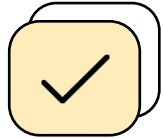


**Содержат
лишнюю
информацию**

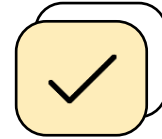


**Сложно оценить
соответствие
нашим
критериям**

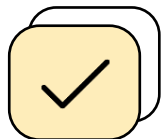
Учебные материалы - Школа разработчиков



Образовательные
результаты = критерии
из матрицы



Начали с «горячих»
компетенций -
проанализировали ИПР



Модули готовят
наши разработчики



Цель - закрыть **80%**
потребностей в обучении



Нет лишнего

Жизненный цикл - итоги развития



Tech review
с другим
сотрудником

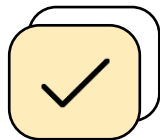


Дружеская
беседа

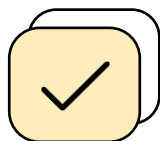


Проверка
по матрице

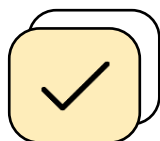
Жизненный цикл - поощрение



Пересмотр з/п



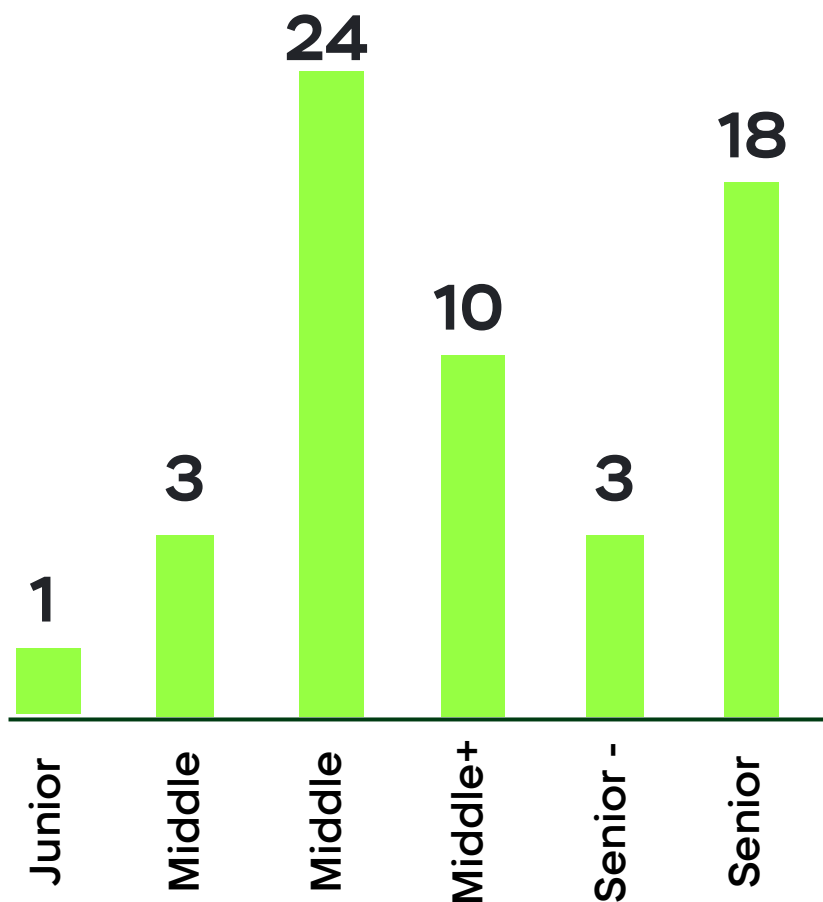
Перевод в другую команду



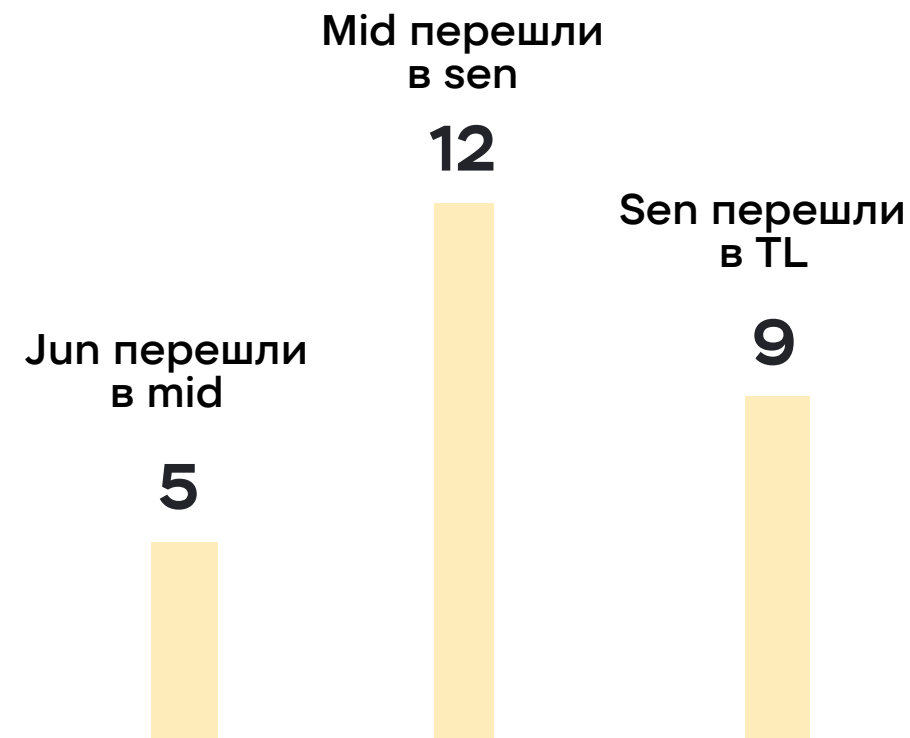
Повышение грейда



Уровни сотрудников



Повышения сотрудников



Матрица - элемент удержания



**Сотрудники
заинтересованы
в развитии**

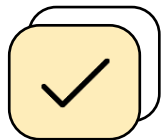


**Развитие
прозрачно**



**Награды
за развитие
ПОНЯТНЫ**

Матрица компетенций - ИТОГИ



Начинайте

с цели - справедливая оценка



Выбирайте

учебные материалы по матрице



Стройте

собеседование по матрице



Удерживайте

сотрудников прозрачным ростом



Развивайте

людей по матрице

Остались вопросы?

Александр Логинов



@saphka

