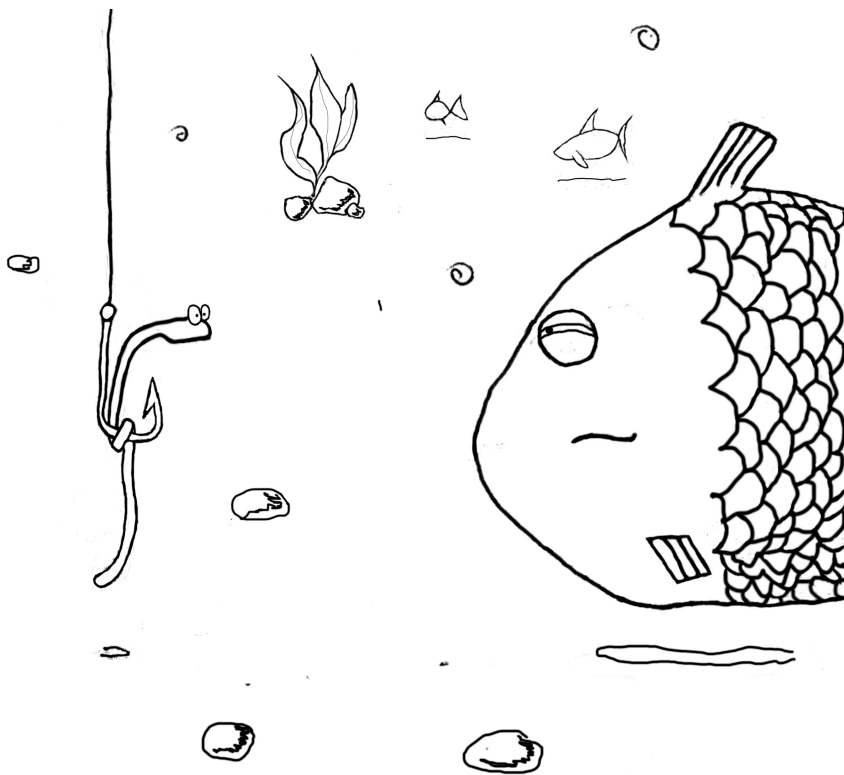


# Братъ или не братъ?

или

## Как собеседовать разработчика

Константин Борисов



УДК 658.3.07

ББК 65.240

Б82

**Борисов, К.**

**Б82**      Брать или не брать? или Как собеседовать разработчика / К. Борисов. – М. : Перо, 2018. – 112 с.

**ISBN 978-5-00122-801-1**

В книге рассказывается о целостном подходе к оценке кандидатов при приёме на работу в сфере ИТ. Большое внимание уделено оценке личностных качеств кандидатов. Также рассказано о методах саморазвития качеств самих интервьюеров, необходимых для успешной работы в этой роли. Описанные в книге концепции проиллюстрированы реальными историями из богатой практики автора.

Книга будет полезна тем, кто занимается собеседованиями разработчиков, а также всем, интересующимся темой найма в области ИТ.

**УДК 658.3.07**

**ББК 65.240**

# Оглавление

<b>1</b>	<b>Введение</b>	<b>7</b>
1.1	Какова цель собеседования? . . . . .	8
	Про непринятие на работу Васи . . . . .	10
1.2	Моральные аспекты собеседования . . . . .	12
1.3	Исследование резюме . . . . .	16
1.4	Начало интервью . . . . .	20
1.5	Резюме и самооценка . . . . .	23
1.6	Структура собеседования . . . . .	26
<b>2</b>	<b>Техническое собеседование</b>	<b>31</b>
2.1	Открытые вопросы . . . . .	32
2.2	Уровни знания . . . . .	35
	Про то, как шесть проектов собеседовали Машеньку . . . . .	38
2.3	Задачи на собеседовании: FizzBuzz . . . . .	41
2.4	Задачи на собеседовании: Цикл в списке . . . . .	45
2.5	Проверка тестовых заданий . . . . .	47
	Про красавицу Марину . . . . .	48
2.6	Технический проект . . . . .	50
2.7	Проверка знания английского . . . . .	53
<b>3</b>	<b>Менеджерское собеседование</b>	<b>57</b>
3.1	Дедуктивный метод . . . . .	57
	Про печального Олега . . . . .	62
3.2	Анализ коннотаций . . . . .	64

Про агрессивного Алексея . . . . .	67
3.3 Технические вопросы от менеджера . . . . .	70
Побег из курятника . . . . .	73
3.4 Вопросы на собеседовании: Изучение нового . . . .	74
3.5 Вопросы на собеседовании: Сложная задача . . . .	79
3.6 Вопросы на собеседовании: Болевые точки в ком- муникациях . . . . .	82
Про ненадёжного разработчика . . . . .	85
3.7 Никто не считает себя плохим . . . . .	87
3.8 Социальное одобрение . . . . .	89
Про тренировку безоценочности . . . . .	90
<b>4 Выведение</b> . . . . .	<b>95</b>
4.1 Офер . . . . .	95
4.2 Разработчик, который не хочет менять работу . . .	98
4.3 Как давать обратную связь кандидатам . . . . .	102
4.4 Основное экономическое соотношение найма . . . .	105
Про неожиданные встречи со старыми знакомыми . . .	108

# Об этой книге

С собеседованиями в сфере разработки ПО сложилась парадоксальная ситуация: все компании сильно озабочены наймом лучших сотрудников, но никто не знает, как это делать.

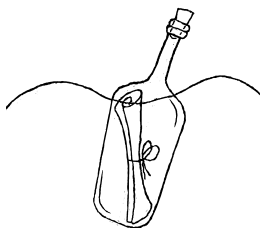
В IT очень много уделяется внимания тренингам. Есть тренинги для разработчиков, тестировщиков, аналитиков и менеджеров. Но тренингов для интервьюеров никто не проводит. Книг, учитывающих специфику индустрии и отечественного менталитета, исчезающе мало. В результате собеседующие двигаются методом проб и ошибок, что часто имеет катастрофические последствия.

За свою карьеру я провёл несколько сотен самых разных собеседований в разных компаниях. Я собеседовал разработчиков, тестировщиков и менеджеров. Я собеседовал на русском и на английском, вживую и через скайп. Я собеседовал студентов и специалистов с сороколетним стажем. И каждое из этих собеседований давало очередной кусочек мозаики, пока в моей голове не сложилась картина, которую я хочу вам, читателям, показать.

Хотя книга, которую вы сейчас читаете, описывает многие технические моменты собеседований, но в целом она сфокусирована на психологических моментах и приёмах интервью. Она будет полезна для лидов и руководителей, которые уже проводят собеседования, и для тех, кто планирует проводить бесе-

дования в будущем. Эта книга будет полезна для кандидатов, которые ищут работу, и хотят оценить, насколько качественно их собеседуют.

Также книга будет полезна любому, кто хочет разобраться в себе и определить для себя направление собственной карьеры. Иногда полезно прособеседовать самого себя.



# Глава 1

## Введение

Многим знакома ситуация, когда проводя собеседование, вы вдруг понимаете, что не знаете, какие вопросы нужно задавать кандидату, о чём и как с ним говорить. Интернет пестрит статьями типа «1000 лучших вопросов для собеседования», но те, кто пытаются использовать эти статьи, разочаровываются.

Ещё неуютней, когда после интервью, разложив перед собой заметки об ответах кандидата, вы понимаете, что не знаете ответов на важные вопросы: Стоит ли нанимать этого кандидата или предыдущий был лучше? Какие проблемы будут в работе с ним? Какие действительно сильные качества выделяют этого кандидата из других?

В этой главе мы затронем самые основы интервью, чтобы лучше понимать, какая задача перед нами стоит, и как надо начинать её решать.

## 1.1 Какова цель собеседования?

Если спросить случайного человека о том, какова же цель собеседования, то он скорее всего ответит фразой, вынесенной в название этой книги: «Брать или не брать?» На самом же деле решение о приёме на работе или отказе делается не на интервью. Оно делается гораздо позднее, после бесед с другими кандидатами, а иногда после повторного интервью с этим же кандидатом.

Вопрос «Брать или не брать?» никак не помогает проводить интервью. Такой узкий взгляд на собеседования загоняет в угол. Например, кандидат вам очень понравился, и вы решили: «Берём!» А через полчаса пришёл следующий кандидат, который тоже произвёл отличное впечатление. И кого из них нанимать? Часто в такой ситуации собеседующий обнаруживает, что у него нет никакой базы, чтобы сделать выбор между двумя хорошими кандидатами.

А иногда наоборот, прособеседовав десяток кандидатов, вы твёрдо решаете не брать никого. Но рынок пуст, и ваше руководство задаёт вопрос: «А если выделить бюджет на 3 месяца обучения, можно ли кого-то из этих кандидатов нанять?» И снова узкая постановка вопроса «брать-не брать» не даёт вам шанса дать осмысленный ответ.

Так давайте разберёмся, каковы же цели собеседования. Вот самые важные из них:

**Составить профиль кандидата.** Речь не только о техническом профиле. Нужно узнать, что за человек перед нами, что его интересует, какие у него жизненные планы, как он может помочь компании, и как компания может ему помочь. Как он будет себя вести в критических ситуациях? Как у него сложатся отношения с коллегами, руководством и заказчиком? Все эти вопросы нужно выяснить. Это очень

сложная задача и последующие главы этой книги написаны про её решение.

### **Оставить положительное впечатление о компании.**

Очень обидно, когда кандидат отвергает ваш offer. А чаще всего это происходит именно оттого, что пытаясь оценить кандидата, вы забыли, что кандидат вас также оценивает. Не только вы принимаете решение, кандидат принимает решение и со своей стороны.

Впечатление от интервью часто является основной информацией о компании. Я сам в своё время перешёл в компанию с потерей зарплаты именно потому, что мне очень захотелось работать с теми людьми, которые меня собеседовали. Оставить такое впечатление, чтобы кандидат захотел работать в вашей компании — это ваша задача.

К тому же не стоит забывать о непринятых кандидатах. Даже не смотря на жуткий кадровый голод в IT, на каждого принятого кандидата приходится более десяти кандидатов отвергнутых. Не стоит считать, что если вы отказываете кандидату, то он просто исчезает. Он продолжает работать, общаться и распространять информацию о вашей компании. Стоит приложить усилия, чтобы эта информация была положительной. Даже если вы уверены, что этот кандидат никогда не будет работать у вас, то будут работать его знакомые. Да и люди меняются и набираются опыта. Через несколько лет вы можете увидеть того же кандидата совсем с другим набором знаний. А вот если кандидат уйдёт с негативным впечатлением, все шансы за то, что к вам он никогда не вернётся.

**Прокачать собственные soft skills.** Многие жалуются мне, что в IT очень редко выпадает шанс потренироваться в soft skills: ведению переговоров, разрешению конфликтов, использованию эмоционального интеллекта, искусству правильной обратной связи. IT компании заполнены интро-

вертами, работающими с машинами, поэтому развивать свои техники общения бывает трудно. Часто недостаток soft skills мешает развитию карьеры, а тренинги дают знания, но не дают достаточно навыков. Так вот собеседования являются хорошим решением этой проблемы.

Собеседования являются переговорами и задачи сторон различные (выяснение целей кандидата является одной из проблем, которую вы будете решать). На собеседованиях нередко конфликты, но в целом обстановка очень дружелюбная. Ваша роль позволяет вам спокойно вести беседу, и кандидат настроен к вам положительно. Для тренировки это практически идеальный расклад. Навыки, выработанные на собеседованиях, очень вам пригодятся когда-нибудь, когда на кону переговоров будет пара миллионов долларов.

Среди этих пунктов составление профиля кандидата является самым важным. После собеседования вы должны очень хорошо представлять, что за человек перед вами. Говорят, чтобы узнать человека, надо прожить с ним рядом много лет. Вам для выполнения этой задачи даётся одно интервью.

## Про непринятие на работу Васи

Как-то мне на собеседовании попался очень вредный кандидат, Вася. Резюме у него было хорошее, и техническое интервью он прошёл весьма успешно. Ко мне он попал на менеджерское собеседование и прямо с порога начал вести себя так, что брать его на работу было абсолютно невозможно. На грани, а иногда и за гранью хамства он рассказал, что компания моя ужасная, интервью у нас отвратительные, да и вообще во всём городе нет ничего его достойного. Причём Вася не то чтобы указывал на какие-то реальные недостатки, которые можно бы-

ло бы исправить, а скорее выказывал своё очень эмоциональное «Фи».

Очень хотелось ответить Васе в том же тоне и выпроводить, но надо же стараться делать мир лучше и добрее! Да и было интересно, что с Василием такое происходит. Очень было любопытно, зачем он пришёл, если считает, что всё плохо. Поэтому я начал упражнение на soft skills, извинился за то, что у него сложилось такое негативное впечатление, сказал, что мы исправимся, поблагодарил за обратную связь и предложил всё-таки поговорить.

И, конечно, мне пришлось сообщить Васе, что речь о его найме идти не может, но сделать это я постарался мягко. Примерно моя речь был следующей: «Мне очень жаль, что у вас сложилось такое негативное впечатление о нашей компании. Спасибо вам большое, что вы потратили время и специально пришли ко мне, чтобы рассказать обо всём, что вы заметили. Решение таких вопросов как раз входит в мои обязанности, и я постараюсь исправить наши процессы, чтобы другие кандидаты не натыкались на те же проблемы. Конечно, с таким впечатлением вы не будете рассматривать наш offer в любом случае. Поэтому о найме говорить дальше смысла нет. Но раз вы всё равно у нас, не согласились бы вы потратить ещё немного своего времени, просто чтобы мы с вами познакомились?»

Собственно на этом вся агрессивность Васи и закончилась. Трудно продолжать кричать на человека, которого видишь в первый раз в жизни, который только что извинился перед тобой и даже поблагодарил. К тому же выглядело так, будто кандидат понял, что и сам не знает, зачем он пришёл. Явно уж не чтоб улучшить наши процессы собеседования. Поэтому мы просто спокойно поговорили.

История Васи оказалась проста. Он хотел переехать в Москву, но всё никак не решался. И вот почти решился. Но, чтобы удостовериться, что в Москве лучше, чем в родном Ом-

ске, он планомерно начал обходить все IT-конторы со строгой ревизией. У омских контор шансов не было. Вася придирчиво относился к беседам и смеялся над любой суммой, которую ему предлагали. Он считал, что в Москве жить ему будет гораздо лучше. Его агрессивность была своеобразной защитной реакцией против страха принятия решения. Он инстинктивно понимал, что если он поссорится с каждой омской конторой, то единственный путь ему будет в Москву.

Я сказал, что прекрасно его понимаю, и что есть много разработчиков в похожей ситуации. Мы поговорили о проблемах Омска и о проблемах Москвы. Я, не покрывив душой, поблагодарил его за интересную беседу, дал свои контакты и посоветовал снова обратиться ко мне, если в Москве не срастётся.

История Васи на этом не закончилась. Он через короткое время вышел на связь и рассказал, как в Москве всё сложно и плохо, как всё дорого, и как казавшаяся высокой зарплата не покрывает расходов. Вася вернулся из Москвы обратно в Омск и первым делом обратился ко мне в поисках вакансии. Вакансий у меня не было, так что Вася устроился в соседнюю контору, но даже устраиваясь туда, просил меня сообщить, как только будет какой-то вариант в моей компании.

Вот так 30 минут, потраченные на казалось бы безнадёжного кандидата, привели к приятельским отношениям с хорошим разработчиком.

## 1.2 Моральные аспекты собеседования

Не могу не затронуть тему, которая очень слабо освещена в статьях и о которой очень мало говорят на практике. Это тема моральных обязательств, которые накладывает роль собеседующего. Без аккуратной работы с моральными аспектами интервью невозможно результативно проводить собеседования и, что

ещё более важно, очень сложно работать с людьми после приёма на работу.

Дело в том, что каждый кандидат очень сильно открывается на интервью. Каждый разработчик приходит на собеседование с намерением рассказать о себе, и рассказывает очень многое. Причём вы как собеседующий имеете (или будете иметь после некоторого опыта работы) специальные навыки для «вытягивания» информации из кандидата. Вспомните какой-нибудь фильм, где обученный сотрудник спецслужб «раскалывает» вражеского агента с помощью гипноза, пентотала натрия или методичного избияния. По возможностям получить знания о человеке вы равняетесь такому сотруднику спецслужб. Вы получаете много очень личной информации. Например, я сам сообщил на интервью, что собираюсь жениться, на целый месяц раньше, чем сделал предложение своей невесте.

Кандидат рассказывает о себе добровольно, доверяет вам. Ваша компания также доверяет вам, и именно вы будете определять лицо компании среди широкого круга профессионалов. Всё это доверие накладывает на вас ограничения, похожие на те ограничения, что имеет лечащий врач, психоаналитик или адвокат. Да, вы не давали клятвы Гиппократу, и ваши обязательства не закреплены в законе, но это не делает их менее серьёзными.

Хотите вы того или нет, но, беседуя, вы получаете очень близкий доступ к каждому кандидату. Самый минимум, который от вас требуется, это не распространять эту информацию<sup>1</sup>.

---

<sup>1</sup>Обратите внимание, что, рассказывая в этой книге истории, я изменяю имена и обстоятельства, чтобы не нанести вреда участникам этих историй. Все совпадения с действительностью — случайны и неправда. Кстати, если с вами на интервью произошло что-то интересное, пожалуйста, напишите мне на [borisovke@gmail.com](mailto:borisovke@gmail.com). Я считаю, что такие истории содержат настоящую мудрость жизни.

Но не всегда это так уж просто. Часто ваша команда прекрасно знает, когда и кого вы собеседуете. Им любопытно и они задают вопросы. Просто игнорировать эти вопросы, значит упустить возможность подготовить почву для будущей работы сотрудника в вашей команде.

Например, после собеседования вы можете удовлетворить любопытство команды: «Очень понравился кандидат. Он и ВУ-Зовские основы не забыл, и практические задачки хорошо порешал, и видно, что с командной работой у него нормально всё». И такой отзыв настроит всю команду на позитивный лад. Встретят такого кандидата после приёма на работу более тепло, чем если вы бы ответили: «Ничего я вам не скажу!»

Но категорически не стоит детально рассказывать про минусы кандидата. Да и некоторые плюсы могут быть закрытой информацией. Вы можете считать плюсом готовность кандидата к многомесячным заграничным командировкам. Но если этот плюс по неформальным каналам дойдёт до жены кандидата, то, возможно, вы этого кандидата потеряете вместе со своей репутацией.

Важно не только, что вы говорите другим. Важно то, какую картину вы строите в своей голове. Ваши мысли определяют все ваши действия. Поэтому самое главное, собеседуя людей, быть как можно более безоценочным.

Навык делать безоценочные суждения очень вам пригодится в карьере и личной жизни. Для интервьюера он необходим. С бытовой точки зрения нас часто тянет давать оценки поступкам и качествам других: «Лёша много работает. Лёша молодец», «Слава слишком много играет в компьютерные игры. Это плохо». И, как я уже говорил, к интервью многие относятся как к «оценке кандидата». Но вспомните главу 1.1 «Какова цель собеседования?» на стр.8. Оценка кандидата — это процесс, который будет запущен на очень поздних стадиях.

Оценка кандидата в первую очередь мешает тем, что сформировав в голове оценку «хорошо/плохо», вы принимаете решение до того, как получите всю информацию. Даже, если вы в середине собеседования решите, что кандидат «плохой», то что это изменит? Вы резко закончите интервью и выставите его за дверь? Подобный поступок покажет вас и всю вашу компанию в крайне дурном свете. Вы должны продолжить интервью, чтобы закончить свою часть работы и ответить на вопросы кандидата.

Мне приходилось завершать интервью досрочно, когда кандидат был настолько неподходящий, что я не мог продолжать разговор. Например, однажды ко мне пришёл парень, который умел переносить компьютеры с места на место и подключать их, а мне нужен был программист. Мне пришлось долго перед ним извиняться за его потраченное время. Потому что в той ситуации «плохим» был не он, а я и моя компания, так как не смогли раньше понять, что он не подходит.

К тому же, если у вас в голове формируется негативная оценка кандидата, то кандидат это наверняка почувствует. Неприятно вести беседу с человеком, которому вы не нравитесь. И кандидат это неприятное чувство запомнит. Если с вашего собеседования человек уйдёт с неприятным чувством, то можете считать, что интервью вы провалили.

Запомните, что нет «плохих» кандидатов. Есть кандидаты, которые вам не подходят. Очень отрезвляюще действует, когда кандидат, которого вы считали профессионально непригодным, идёт в другую компанию, получает там хорошую работу и справляется с этой работой на протяжении долгих лет. Это не значит, что вы ошиблись, просто все мы разные, и хотим разного, и показываем себя с лучшей стороны в разных условиях и разных компаниях.

К тому же ярлык «плохой» закрывает от вас конструктивный путь. Например, кандидат «плохо» умеет формулиро-

вать свои мысли. Что именно ему будет трудно делать? Кроется ли проблема в самом мышлении или в коммуникации? Как кандидат с этим справляется сейчас? На самом ли деле это будет мешать кандидату работать? Ярлык «плохо» мешает анализировать ситуацию. Как и ярлык «хорошо».

Старайтесь понять кандидата, вжиться в его шкуру и принять его таким, какой он есть. Не считайте кандидата «плохим человеком». Любые проблемы, которые вы видите, можно объяснить влиянием среды (см. историю «Про печального Олега» на стр.62).

Ваша задача не оценить кандидата, а получить информацию и понимание. Очень большое удовлетворение испытываешь, когда видишь, что несмотря на какие-то сложности, кандидату удаётся найти себя в вашей команде.

### 1.3 Исследование резюме

К любому интервью нужно готовиться. В зависимости от «сложности» кандидата я трачу от 10 до 30 минут на такую подготовку, а иногда и больше. Резюме является основным источником информации, и его нужно обязательно просматривать перед интервью.

В первую очередь надо попытаться получить целостную картину карьерного роста кандидата. Например, кандидат провёл 15 лет в одной компании, и его карьерный рост выглядел как C++ Junior (1 год) → C++ middle developer (2 года) → C++ Senior developer (7 лет) → Технический Директор (5 лет). Пока всё нормально, это очень понятная цепочка.

А сейчас этот кандидат пришёл к вам и хочет устроиться на роль Архитектора. Сразу возникают вопросы, которые надо будет прояснить на интервью: Почему он не хочет работать даль-

ше техническим директором? Какое соотношение между менеджерской и технической работой у него сейчас? Не забыл ли он, как работать руками? Ничего особо подозрительного нет, но эти вопросы должны быть прояснены. Иначе полной картины кандидата не сложится и принять обоснованное решение о нём вы не сможете.

Иногда, когда выясняешь подобные вопросы, встречаешь неожиданное сопротивление кандидата. Например, в одном резюме я увидел «пропуск» в 2 года. Спросив кандидата об этом периоде, я ответа не получил, а получил какие-то очень уклончивые формулировки с непонятными шуточками. Пришлось потратить время, чтобы получить прямой ответ. Оказалось, что кандидат занимался созданием игр с порнографическим контентом и вирусов. В некоторых организациях такие моменты могут быть очень критичными.

Важно, чтобы вы могли получить все нужные вам ответы. В реальной работе вам придётся задавать очень неудобные вопросы: «Почему у заказчика релиз не установился с очевидными ошибками, которые мы должны были найти? Почему тебя не было на вчерашнем митинге? Почему команда заявляет, что не хочет с тобой работать?» Сложные проблемы часто ведут к крайне неприятным разговорам. Способность кандидата цивилизованно обсуждать любые вопросы критически важна. Это часть того, что называется профессионализмом.

Однако это не значит, что вы обязательно должны получить полный отчёт о том, что происходило с кандидатом в каждый момент времени. Подойдёт любой ответ, который вас удовлетворит, например: «В эти годы я не мог работать из-за семейных проблем, о которых говорить сейчас я не хочу». Несколько уточняющих вопросов: «Как вы восстанавливали навыки, потерянные за эти два года?», «Не тяжело было столько жить без программирования?» — и если кандидат даёт вменяемые ответы, то тему можно считать закрытой.

Вообще, любые мелочи в резюме, которые «царапают» глаз являются очень хорошим материалом для обсуждения на интервью. Например, в одном резюме я увидел:

Национальность: русский

Ни одно из сотен просмотренных мной резюме не содержало указания национальности. Я напрягся. Работая в международной компании, очень не хочется получить в команду националиста. На прямой вопрос кандидат дал простой ответ. Он использовал какой-то генератор резюме и там просили заполнить это поле. Как резюме сгенерировалось, так и отправил. Так что мои худшие подозрения не оправдались, зато выплыл другой момент, который позже подтвердился. Кандидат был склонен решать проблемы очень просто, концентрируясь на главном и не тратя время на «полировку» деталей.

Или вот ещё одна цитата из резюме:

Мое главное увлечение - это типографика.

Во-первых, я сам люблю типографику и с удовольствием могу поговорить про неё. А во-вторых, здесь использовано «е» вместо «ё» и дефис («-») вместо тире («—»),<sup>2</sup> что для любителя типографики немыслимо. Требования к вакансии не подразумевали знаний типографики, так что минуса особого для кандидата не было. Но для меня это был достаточный повод, чтобы обсудить, как кандидат изучает новое и насколько



Рис. 1.1: Фото в резюме

<sup>2</sup>Пожалуйста, сообщайте о подобных и любых других ошибках в этой книге мне на почту borisovke@gmail.com.

ко глубокие знания он считает достаточными.

Иногда резюме содержат подсказки о характере человека. Например, мне нравится, когда кандидат проявляет юмор. Интервью с такими кандидатами можно начинать гораздо быстрее и свободней. А однажды мне попалось резюме первая страница которого выглядела примерно так, как показано на рисунке 1.1. Около половины страницы занимала фотография кандидата. Студийная, в полный рост, с тщательно выставленным светом и мейкапом.

Опять же, никакого криминала нет. Может быть, у кандидата чувство юмора развито. Но такое фото в резюме даёт пищу для размышлений. И я включил в план беседы несколько вопросов, которые проверяли, насколько этот кандидат проникнут сознанием собственной важности, и как он воспринимает критику. Оказалось, что очень плохо воспринимает. И вот уже это действительно проблема.

Или, например, я интересуюсь, какие языки знает кандидат. Сам недавно выбирал язык для изучения (выбрал испанский), поэтому мне очень любопытно, что и как учат другие люди.

Такие нейтральные темы хороши как для начала интервью, так и для его завершения. В начале интервью они помогают наладить контакт. Кандидат видит, что вам действительно интересна та же тема, что интересна ему. И это задаст правильный тон всему интервью.

А завершение интервью такой темой тоже может быть полезным. Я в самом конце интервью, даже после вопросов кандидата, иногда говорю что-то вроде: «Наше интервью уже можно считать завершённым, но меня просто очень зацепило то, что вы изучаете китайский. Правда, это настолько сложный язык, как говорят?» Подобное ваше любопытство оставит хорошее впечат-

ление. Ведь интервью уже закончено. Но интерес есть, поэтому неформальная беседа будет приятным завершением встречи.

Только не надо подходить к подобным вопросам протокольно: «Надо задать личный вопрос кандидату на прощание, поэтому я задам такой вопрос. И пошучу». Интервью — процесс взаимный, и как кандидат открывается вам, так и вы открываетесь кандидату. Вряд ли у вас актёрский талант настолько развит, что вы сможете показать заинтересованность, если тема вам скучна. Вам надо быть искренне заинтересованным в кандидате, в его опыте и его характере. Кандидат это почувствует и ответит взаимностью.

Хороший интервьюер искренне интересуется другими.

## 1.4 Начало интервью

Итак, кандидат пришёл на интервью, вы обменялись парой приветственных слов и вам надо начинать интервью. Это очень важный момент, где вам придётся попотеть.

Самая главная проблема заключается в том, что кандидат вас опасается. Кандидат готов отвечать на ваши вопросы, но первоначальное отношение к вам, скорее всего, будет настороженное. Почему? Потому что в мире очень мало хороших интервьюеров, и кандидат знает, что все шансы за то, что интервью будет плохим.

В этом нет ничего личного. Просто в мире мало мастеров своего дела. Вспомните сколько вы видели таксистов, которые действительно могли бы считаться профессионалами своего дела. Ну или хотя бы не нарушали правила дорожного движения.

Ну или хотя бы не совершали нарушения, за которые наказывают лишением прав.

Ладно-ладно, не подумайте, что я ненавижу таксистов, среди них есть отличные водители. Но можете вспомнить, например, сколько сил отнимает поиск хорошего мастера или надёжной СТО.

Проблема в том, что высокопрофессиональных людей мало. Так что кандидат совершенно справедливо подозревает, что перед ним плохой специалист. Раз вы читаете эту книгу, то вы стремитесь развиваться и к числу некомпетентных собеседователей не относитесь, но кандидат-то этого ещё не знает.

Причём кандидат находится в явно неравноправном положении. Конечно, известно, что в IT работники нужны компаниям больше, чем работникам нужна работа, но на стандартном интервью это не заметно. Обычно кандидат находится в подчинённом положении и должен отвечать на все вопросы интервьюера, который, как я уже говорил, часто просто некомпетентен.

Почти каждое интервью дарит кандидату очень неприятные сюрпризы. На моём опыте было:

- вместо собеседования мне предлагали 4 часа программировать на незнакомом мне языке (Как раз выдалась свободная неделя, на что бы её потратить?);
- мне заявляли, что я не мог придумать те решения, которые я даю, и что я где-то видел их раньше (А слабо придумать задачу с нуля, чтобы не было таких опасений?);
- над моим резюме смеялись (Теперь мне это кажется смешным.);
- мне говорили, что я слишком много общаюсь с американцами и что я говорю по-русски с акцентом (Really?!).

Это всё мелочи жизни, но для интровертов, которых в IT много, такие случаи могут быть очень травмирующими (см. историю «Про то, как шесть проектов собеседовали Машеньку» на стр.38).

В общем, первое что вы должны сделать — это показать кандидату, что вы вменяемы и что вы с кандидатом находитесь на равных в переговорном процессе. Выполнив вторую задачу, вы автоматически выполните первую, поэтому сконцентрируемся на ней.

Скоро мы будем бомбардировать кандидата вопросами, поэтому логично и честно начать с того, что мы дадим ему какую-то информацию. Как минимум, нужно представиться, сказать, сколько времени будет занимать интервью, и примерно описать круг вопросов, которые вы будете задавать. Например, сказать, что не все вопросы будут чисто техническими и что придётся решить пару практических задач.

Вы, скорее всего, будете первым человеком, который реально может рассказать о проекте, об используемых технологиях и о том, кого вы именно ищете. Так и расскажите об этом! Ответьте на какие-то начальные вопросы, которые могли появиться у кандидата, и пообещайте, что после интервью вы ответите на все вопросы, которые к тому времени накопятся.

Вы потратите 10 минут вашего времени, но зато докажете кандидату, что вы не только будете спрашивать его, что вы готовы общаться на равных. Кроме того, вы берёте на себя первую, самую тяжёлую часть разговора. Кандидату теперь гораздо комфортней отвечать на ваши вопросы. Он уже послушал вас, привык к вам, успокоился.

А вы зато задаёте тон разговора. Вы можете быть подчеркнута формальным или наоборот шутить, показывая, что серьёзный разговор всё равно может быть развлечением. Главное в этой части быть открытым.

Конечно, вы можете быть связаны различными соглашениями о конфиденциальности, но кандидат должен чувствовать вашу готовность рассказать всё, что вы можете. Потому что это именно то, что вы со своей стороны хотите от него.

Следующим пунктом хорошо задать вопрос, который а) потребовал бы развёрнутого ответа (так как в дальнейшем нам нужны развёрнутые ответы) и б) был лёгким для кандидата (чтобы не ставить его в тупик с самого начала). Я прошу: «Расскажите, пожалуйста, про ваш опыт работы». Можно аналогично спросить про текущий проект. Или про опыт работы с какой-то конкретной технологией. Или про хобби, если оно зацепило ваш взгляд в резюме.

Как результат после примерно 15 минут разговора у вас установлен контакт с кандидатом, кандидат готов не просто отвечать на вопросы, но и разговаривать, и вы можете переходить к основной части интервью. Время, потраченное в начале интервью, окупится сторицей, когда вам потребуется получать прямые ответы на неудобные вопросы.

## 1.5 Резюме и самооценка

Резюме является самым главным источником начальной информации о кандидате. Но в целом резюме разработчиков ужасны. Мало людей умеют составлять резюме, а кадровый голод позволяет кандидатам и не учиться это делать. Как квинт-эссенция у меня сейчас значится резюме разработчика, из которого я не смог понять, на каком языке программирования он пишет. То есть очень часто бывает, что указано несколько языков (Java, Python, R, ...), но непонятно, насколько каждый из языков знаком. Но в том резюме вообще ни одного языка указано не было, хотя и заявлялось 7 лет опыта разработки.

Хорошее решение я видел в компании Luxoft. Там рекрутеры высылают кандидатам стандартный шаблон резюме и просят его заполнить. А в этом шаблоне, кроме прочего, есть табличка со списком основных языков программирования и технологий. Кандидат для каждой строчки должен заполнить уровень владения соответствующей технологией от 0 (ничего не знаю) до 5 (эксперт). Такая табличка стирает индивидуальные особенности кандидатов, что для собеседования, конечно, плохо. Зато появляется уверенность, что минимальная необходимая информация будет доступна до интервью.

Так вот, в процессе многолетней работы с такими резюме стала видна одна закономерность. Если в резюме много оценок 4 и 5 (от продвинутых знаний до уровня эксперта), то скорее всего кандидат имеет уровень junior'a.

Действительно опытные эксперты обычно гораздо более скромны в оценке своих знаний. Пятёрки они ставят всего в нескольких строчках, для технологий, с которыми они работали десяток лет, или по которым они защитили кандидатскую.

Для ряда технологий стоят четвёрки. Это там, где у них просто многие годы опыта, но недостаточно, чтобы они стали экспертами мирового уровня. И ещё кучка технологий, которые оценены в 3 и меньше. Опытный человек всегда имеет широкие познания в смежных областях.

Причём, многие простые технологии опытные разработчики просто не указывают, считая их само собой разумеющимися. Вот и получается, что если смотреть только оценки, то действительно опытные эксперты выглядят какими-то троечниками.

В психологии это называется эффектом Даннинга—Крюгера и качественно изображается следующим графиком:



Рис. 1.2: Эффект Даннинга—Крюгера

Профи всегда знает, насколько велика область за пределами его знаний и рассуждает примерно так: «За знание языка C++ я только 4 могу себе поставить. Конечно спецификации C++ v.11 и C++ v.14 я знаю очень хорошо, но с теорией компиляторов я не очень дружу и не могу иногда объяснить, почему создатели языка приняли то или иное решение. Вот Вася — тот точно эксперт. Он на международных конференциях по этой теме выступает. Я не так крут, как Вася. Блин. Может даже 3 надо поставить».

Новичок рассуждает примерно так: «Знание языка C++ у меня точно на пятёрку. Я же знаю все операторы циклов: и for, и while, и даже этот странный do..while. И даже зачёт по C++ сдал, хотя и с трудом. Препод, зверь, валил пострашному».

В процессе работы неоднократно кто-то из команды прокачивался в какой-то технологии и шёл исправлять в своём резюме оценку 5 на 4. Так что если вы видите очень скромное описание основных навыков, то не стоит отбрасывать такого кандидата.

Наш менталитет поощряет занижение собственных достижений. Возможно кандидат просто очень скромный.

Собеседуя кандидатов, нужно этот момент учитывать. Если разработчик говорит, что какую-то технологию он знает не очень, то это может означать, что он просто никак докторскую по этой теме закончить не может.

Новички же сильно переоценивают свои знания. Вместо общих вопросов об уровне владения технологиями, лучше задавать конкретные: Был ли опыт использования этой конкретной технологии в реальном проекте? Какие сложности встречались и как они были преодолены? Чем нравится и чем не нравится эта технология?

## 1.6 Структура собеседования

Интервью — это не просто куча вопросов, которые вы накидываете кандидату. В каждый момент собеседования вы должны понимать что и для чего вы делаете. Поэтому давайте рассмотрим условные «типы» интервью, то есть типы вопросов, которые вы задаёте. Я называю их условными, потому что реальное интервью обычно содержит кусочки разных типов.

**HR интервью.** Обычно проводится сотрудником HR отдела. Это короткое интервью, на котором оценивается общий уровень адекватности кандидата, проверяется заинтересованность кандидата в работе, определяется формат дальнейшего общения. Здесь, как и во всех других интервью, не только у кандидата спрашивают что-то, но кандидату даётся информация. Кандидату рассказывают о компании, о процессе собеседования, о том, что его ждёт дальше и чем всё это закончится.

Не стоит считать, что для кандидата всё очевидно. Собеседования в разные компании очень различаются и кандидат (особенно неопытный) может быть сильно удивлён, что ему сразу не сказали уровень зарплаты или что собеседования нужно будет проходить на английском и в вечернее время. Многие компании делают памятку (в напечатанном виде или на сайте), которая отвечает на основные вопросы. Если такой памятки нет, то, возможно, вам придётся самому ответить на вопросы, относящиеся к HR.

**Техническое интервью.** Обычно, именно эта часть считается «настоящим» собеседованием. Не только технари, но, что удивительно, и управленцы считают оценку технических знаний единственно важной частью собеседования. Причина этого заблуждения проста: абсолютное большинство компаний не имеют людей с нужными навыками для оценки чего-то кроме технических навыков.

Техническое интервью часто совсем «не похоже» на интервью. Это может быть тест, тестовый проект, решение задач на кодирование или предоставление кода, написанного в прошлом. Есть много инструментов, которые могут помочь в оценке технических знаний кандидата. Но беседа живого эксперта с кандидатом не может быть заменена всеми этими вспомогательными средствами.

Кстати, английский язык — это тоже технический навык, и его оценка мало чем отличается от оценки умения писать на Java. Равно как, если вы беседуете менеджера, то умение применить Scrum тоже относится к навыкам техническим.

Техническому интервью разработчиков посвящён следующий раздел этой книги.

**Менеджерское интервью.** Я называю его менеджерским потому, что обычно его проводит нанимающий менеджер. Но

его вполне может проводить и технический специалист, если его квалификация позволяет это.

Этот тип интервью предназначен для исследования личности кандидата. Собеседующий должен понять, насколько кандидат впишется в текущую команду, какие проблемы будут с ним, как можно использовать его сильные стороны. Высокий технический уровень кандидата совсем не означает, что он будет отличным работником в вашей компании. И на практике я часто видел, как прекрасный, но неправильно нанятый технический специалист, приносил гораздо больше вреда, чем пользы.

Менеджерскому интервью также посвящён целый раздел этой книги.

**Офер.** Собственно предложение работы кандидату, которое также часто выливается в отдельное интервью. Более детально описано в главе 4.1 «Офер» на стр.95.

Не нужно путать тип интервью с целями интервью. Например, цель «Оставить хорошее впечатление о компании» должна быть частью любого интервью. Неправильно думать, что если вы оцениваете технический уровень кандидата, то достаточно сконцентрироваться на этом и не обращать внимание ни на что другое. Часто кандидат принимает решение о принятии офера, основываясь на уровне технического собеседования.

Также технический специалист не может игнорировать «менеджерскую» часть интервью. В итоге этот кандидат придёт в вашу команду, и вам придётся с ним работать. Эта работа может быть счастьем или мучением, и в ваших интересах узнать кандидата получше.

На практике примерно 20% времени технического интервью я трачу на «менеджерские» вопросы. Как кандидат относится к используемым в вашем проекте технологиям? Сможет ли кандидат изучить новые технологии? Будет ли он прислуши-

ваться к вашим замечаниям по его коду? Все эти вопросы нужно выяснить.

Даже часть, относящаяся к оферу, не может быть полностью проигнорирована на техническом интервью. Например, если вы видите явный пробел в знаниях кандидата, который будет ему мешать, нужно это сказать ему (аккуратно) и получить подтверждение от кандидата, что он ликвидирует этот пробел в будущем.

Это может выглядеть вот так: «Я вижу, что вы имеете мало опыта в работе с многопоточностью. Наш проект требует хороших знаний в этой области. Если вы будете работать с нами, то вам придётся работать с многопоточностью постоянно. Готовы ли вы к этому? Как вы думаете, сможете ли вы в течение пары месяцев прокачаться в этой области? Как вы это сделаете?» Если кандидат примет офер, он должен хорошо понимать, на что он подписывается.

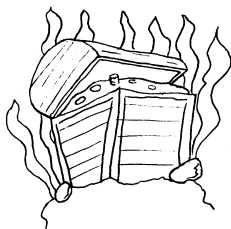
Если говорить конкретно о распределении времени, то для часового технического интервью я рекомендую начать с короткого рассказа о себе, проекте и компании (HR часть). Следующие полчаса лучше посвятить основной части интервью, техническим вопросам. После технических вопросов пятнадцать минут надо уделить вопросам менеджерским. И это оставит десять минут на вопросы кандидата, что будет продолжением HR части.

Если вы руководитель, то логичнее, наоборот, менеджерскую часть делать полчаса, а техническую — 15 минут, но порядок лучше оставить прежним: сперва техническая часть, потом менеджерская. Технические вопросы привычней для кандидата, вызывают меньше напряжения и являются подготовкой для менеджерской части.

Таким образом, типовая структура интервью следующая: Знакомство → Технические вопросы → Менеджерские вопросы → Ответ на вопросы кандидата

Очень важно отслеживать время, чтобы успеть всё. Затянувшееся знакомство с интересным кандидатом может не оставить времени в конце интервью на ответы на вопросы, а это неприемлемо. Обычно я выкладываю перед собой свои наручные часы, чтобы точно знать, успеваю ли я обсудить все вопросы.

Если какой-то вопрос отнимает больше времени, чем было задумано, то имеет смысл просто перейти к следующему. Иногда у вас и у кандидата есть время, чтобы провести интервью дольше запланированного, но часто вы оба подчиняетесь жёсткому графику, и такой возможности нет.



## Глава 2

# Техническое собеседование

Область IT состоит из технарей, и задачи перед ними стоят очень технические. Именно поэтому у нас настолько высоко ценятся эксперты, и именно поэтому компании прикладывают столь значительные усилия, чтобы оценить технический уровень кандидатов.

Конкретные вопросы на техническом очень сильно зависят от используемых в проекте технологий, языка программирования и философии разработки, принятой в компании. Конечно, собеседование разработчика C++ для встраиваемых криптографических решений очень сильно отличается от собеседования разработчика приложений для анализа данных на языке R. Но принципы остаются одинаковыми.

В этой главе мы поговорим об этих принципах и рассмотрим примеры конкретных задач, которые я использую на собеседованиях.

## 2.1 Открытые вопросы

Обычно, для оценки технического уровня кандидату задают по списку очень узкие вопросы: «Когда вызываются статические конструкторы в C#?», или «Реализацией какого шаблона проектирования являются события?», или «Тип DateTime передаётся по ссылке или по значению?» Но этот подход не становится решением, а ведёт к появлению целого набора других проблем.

Если вопросы достаточно сложные, то кандидат может легко забыть конкретный ответ на конкретный вопрос. Ведь в условиях стресса на собеседовании мы легко можем забыть детали спецификации языка, правда? Поэтому приходится добавлять много вопросов на одну и ту же тему. Такое тестирование начинает отнимать всё больше времени, а результаты становятся всё менее и менее относящимися к реальной работе.

Самая же главная проблема такого подхода — кандидат замыкнётся, так как бомбардировка человека вопросами совсем не помогает наладить контакт. И кандидат быстро устанет от бессмысленных вопросов и сочтёт ваш подход непрофессиональным. И будет прав.

Но ведь любой разработчик определенно может сказать, насколько крут его собеседник после короткого разговора. Вы, возможно, сами не раз это видели в кулуарных разговорах на конференциях. Когда встречается эксперт, то уже через несколько минут становится очевидно, что он эксперт. Как же повторить этот трюк в условиях собеседования? Как сделать его управляемым?

Решением является отказ от списка узких вопросов в пользу беседы на технические темы. Собеседование должно проводиться в формате разговора. При этом не нужно спрашивать отдельные факты, а нужно оценивать зрелость кандидата и уро-

вень в целом, исходя из того, насколько свободно он себя чувствует в разных областях. Для этого необходимо задавать довольно общие вопросы и получать от кандидата развёрнутые ответы.

Возвращаясь к примеру с кулуарными разговорами на конференции, можно заметить, что вряд ли кто-то когда-нибудь произвёл впечатление, вспомнив полный список параметров стандартной функции или перечислив все известные способы запуска отдельного потока вычислений. Но часто эксперт показывает свой уровень, выразив мнение о ближайшем будущем какой-то технологии или рассказав, как он сумел решить какую-то проблему в проекте. Поэтому в собеседовании стоит придерживаться открытых вопросов, не подразумевающих конкретные ответы, а подталкивающих кандидата к обсуждению какой-то темы.

Вот несколько примеров таких открытых вопросов:

- Какая самая серьёзная проблема вам встречалась в ваших проектах?
- Что бы вы хотели изучить в ближайшее время?
- Если бы вам достался на поддержку большой старый продукт с большим количеством кривого кода, то как бы вы стали исправлять в нём баги?
- Какие проекты вам нравятся больше всего?
- Какие технологии из тех, с которыми вы работали, вы считаете самыми бесполезными?

С этим подходом есть одна проблема. Несмотря на то, что кандидаты с презрением относятся к обычным «протокольным» вопросам типа «назовите точный тип первого параметра функции `scanf()`», открытые вопросы многих из них ставят в тупик. Такие вопросы кажутся слишком общими, не имеющими точно-

го ответа, спорными. Многие разработчики с подозрением к ним относятся и молчат.

Разговорить кандидата и объяснить, что вы хотите услышать — это ваша задача. Например, вы решили поговорить о шаблонах проектирования. И задали несложный вопрос: «Какие проблемы с использованием шаблонов проектирования вы знаете или встречали на практике?» В ответ вполне реально получить не длинный ответ, а напряжённую тишину. И если вам кажется, что кандидат смотрит на вас непонимающим взглядом не потому, что он действительно не знает проблем шаблонов проектирования, а потому, что он не понимает, чего вы от него ждёте, то попробуйте задавать вопросы более развёрнуто, переформулируйте вопрос, приведите пример ответа или сделайте всё это разом.

Например, так: «Да, я понимаю, что мой вопрос слишком общий. Просто не бывает плюсов без минусов. Любая технология имеет какую-то цену и применима не везде. Многие книги рассказывают нам, что шаблоны проектирования — это классная штука, и это правда. Но мне хочется поговорить о проблемах, которые они рождают. Ведь всегда есть проблемы. Например, шаблоны надо учить и тратить на это время. В канонических книгах перечислены десятки шаблонов, а в одном проекте применяется ничтожный процент из них. То есть время, потраченное на изучение многих шаблонов потрачено зря. Может быть вы знаете ещё какие-то минусы шаблонов проектирования, которые мы могли бы обсудить?»

Такой развёрнутый вопрос стимулирует кандидата дать развёрнутый ответ. Кандидат может поспорить с вами и сказать, что даже если шаблон не применяется, то время, ушедшее на его изучение, всё равно нельзя назвать потраченным впустую, что изучение шаблонов развивает разработчика (назвав таким образом ещё один плюс). Или поговорить о том, что шаблоны проектирования учат структуре мышления, и у разработчика лю-

бая программа состоит из шаблонов проектирования, реализация которых задаёт некоторый доменно-ориентированный язык, на котором написана бизнес-логика приложения.

## 2.2 Уровни знания

Беседуя с разработчиками на разные темы я заметил, что уровень зрелости разработчика можно определить по набору тем, на которые он может вести беседу. Есть базовые знания, которые доступны разработчикам даже с небольшим опытом, а есть действительно сложные области, в которых разбирается только эксперт. Иерархия тем показана на рисунке 2.1.



Рис. 2.1: Уровни знания по технологии

Давайте рассмотрим какую-нибудь область, например, юнит-тесты и разложим знания разработчика в соответствии с этой иерархией:

**Определения.** Сюда можно включить базовые понятия, которые получает junior-разработчик в самом начале изучения: «Что такое юнит-тест?», «Что такое тест-метод?», «Какие

фреймворки для юнит-тестирования бывают?» Обычно не стоит тратить время на задавание этих вопросов. Если кандидат не знает ответы на них, то это скоро станет и так очевидно, а правильные ответы не скажут о разработчике ровным счётом ничего.

**Плюсы.** Что дают нам юнит-тесты? Зачем мы их вообще используем? Ответы на эти вопросы даются в любой книге по юнит-тестам. Разработчик должен знать ответы на них, чтобы применять юнит-тесты осмысленно:

- Они позволяют проводить автоматическое регрессионное тестирование;
- Без них трудно делать функциональное тестирование для многих программ (например, реализация какой-то библиотеки на C++);
- Они являются своеобразной документацией (особенно, если они содержат комментарии).

**Минусы.** О чём не пишут ни в каких книгах, так это о минусах технологий и подходов<sup>1</sup>. Но ведь каждый разработчик много раз сталкивался с проблемами новых прекрасных технологий. Managed code. Многопоточность. Микросервисная архитектура. NoSQL. Шкура опытных разработчиков покрыта шрамами, каждый из которых подписан этими названиями. Знание минусов — это то, что отличает матёрого разработчика от оптимистично настроенного новичка.

Например, для юнит-тестов:

- Написание юнит-тестов требует много времени. А их поддержание требует времени ещё больше;

---

<sup>1</sup>Если не верите, попробуйте найти в этой книге упоминания о минусах предлагаемых мной методов. И если найдёте, то сообщите, пожалуйста, мне на почту [borisovke@gmail.com](mailto:borisovke@gmail.com), чтобы я убрал это в следующем издании.

- Стопроцентное покрытие юнит-тестами часто невозможно. Для бизнес-логики юнит-тесты хороши, но если много UI и взаимодействия со сторонними системами, то юнит-тесты могут быть почти бесполезны;
- Запуск юнит-тестов требует времени, которое добавляется к каждому билду. Иногда это время становится недопустимо большим.

**Философия.** Когда разработчик начинает понимать взаимосвязь между разными технологиями, их место в процессе разработки и своё место в этом процессе, он выходит на уровень философии. На этом уровне кандидат может рассказать, почему юнит-тесты стали неотъемлемой частью процесса разработки, как они связаны с Continuous Integration и Continuous Deployment, как можно снизить затраты на их создание и как можно доказать заказчику, что они нужны.

В своих собеседованиях я обычно в каждой теме пробегаюсь по уровням, игнорируя самый базовый уровень определений. Я открываю обсуждение вопросом: «Зачем нужна эта технология?» Если кандидат даёт чёткий и развёрнутый ответ, то следующим пунктом спрашиваю: «А какие проблемы есть с использованием этой технологии?» Очень опытные кандидаты часто, отвечая на эти открытые вопросы, сами выходят на уровень философских рассуждений, но им можно помочь, задав вопрос: «А как вы сами относитесь к этой технологии?» Похожий подход нужно по порядку применить ко всем областям, которые вы хотите обсудить. Я это называю «сканированием знаний» кандидата.

Не стоит ограничиваться только философскими вопросами. Разработчик может иметь высокий общий уровень, что даст ему возможно хорошо отвечать на высокоуровневые вопросы, но «плавать» в относительно простых темах. Однако же если разработчик быстро назвал основные плюсы технологии, основ-

ные минусы и описал, как эта технология вписывается в современный подход к разработке, то уровень его в этой технологии высок. Он мог пропустить что-то из-за невнимательности, или стресса, или своих убеждений, но даже если вы промучаете его ещё полчаса дополнительными хитрыми вопросами, то оценка уровня не изменится. Вы можете уточнить у кандидата, почему он не назвал что-то, но только если вы считаете пропущенное действительно очень важным.

Если вам нужно очень быстро (например, за 20 минут) прособеседовать мощного архитектора, то вполне достаточно задать один развёрнутый философский вопрос. Например: «Как современные практики и подходы к разработке позволяют разрабатывать код быстро и качественно?» Ответ, скорее всего, позволит вам определить уровень кандидата в целом. Хотя, конечно, будет лишён детальности.

## **Про то, как шесть проектов собеседовали Машеньку**

Рекрутёры нашли новую кандидатку, Машеньку. Опыта у неё было немного, но резюме давало надежду, что она дотягивает до уровня Middle Developer, ну или хотя бы является твёрдым Junior'ом, которого можно быстро подтянуть.

Рынок кандидатов, как обычно, был почти пуст, а потребность в кандидатах была, как обычно, высокая. Поэтому новый разработчик нужен был сразу в шесть разных проектов.

Чтобы не гонять кандидатку несколько раз и не давать преимущество ни одному из проектов, было назначено часовое интервью одновременно на все шесть проектов. Интервьюеры об этом не знали, поэтому были удивлены, когда в назначенный час в переговорке собралось восемь человек (некоторые проекты выставили двоих собеседующих).

Ещё сильнее была удивлена Машенька. Она была стеснительной девушкой и явно испугалась шумной толпы тимлидов. Она села на стульчик в центре комнаты, чувствуя тот же ужас, который испытывала недавно на защите диплома перед высокой комиссией.

Самый активный из интервьюеров начал:

— Ну что ж, расскажите нам, пожалуйста, для начала про принципы SOLID.

Кандидатка напряглась, собираясь с мыслями, но ответить ничего не успела, так как подключился второй тимлид:

— Да ладно SOLID. Здесь и старшие разработчики плавают. Лучше скажите, есть ли у вас опыт работы с базами данных?

— Нет, позвольте, — первый не сдавался, — для моего проекта знание SOLID является ключевым!

— А для нашего ключевым является знание БД! — второй либо любил спорить, либо не любил принципы SOLID.

— Да дайте же ей ответить что-нибудь! — вступился за девушку третий интервьюер.

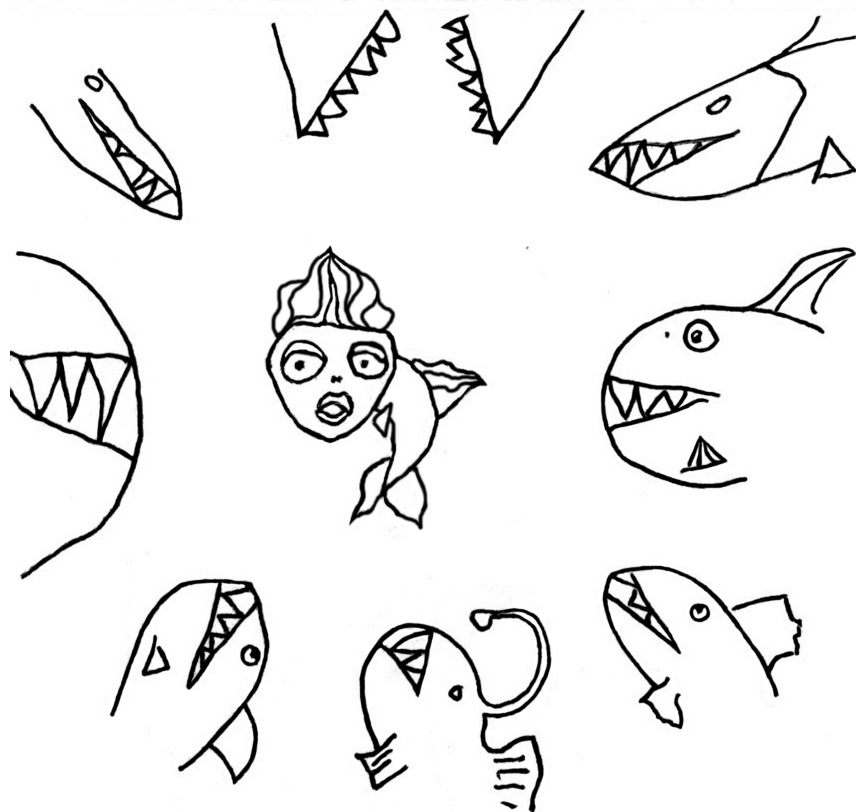
Машенька уточнила:

— А про что отвечать, про SOLID или базы?

— SOLID! Базы! — указания прозвучали одновременно. Представитель одного из проектов мрачно встал и молча вышел из переговоров. В глазах Машеньки читался ужас.

Через пятнадцать минут дружеского переругивания был установлен порядок собеседования. Каждый успел задать один вопрос, некоторые даже два. Машеньке было очень трудно, так как темы менялись со скоростью света, наводящие вопросы противоречили основным и интервьюеры часто перебивали её и друг друга. Когда Машенька ушла, все тимлиды признались, что у них не получилось понять, что она знала.

Неизвестно, что подумала Машенька, но все остальные подумали одно и то же: «Больше — никогда!»



## 2.3 Задачи на собеседовании: FizzBuzz

При собеседовании специалиста вам нужно посмотреть, как он реально работает. Для тестировщика нужно дать ему что-то протестировать. Для менеджера — решить какую-то управленческую задачу. Для разработчика — дать ему написать код.

Зачем смотреть, как разработчик пишет код? Потому что написание кода очень сильно отличается от ответов на вопросы. При этом задействуются совсем другие знания, работают другие рефлексy. Можете перебрать в уме каких-то знакомых вам разработчиков, чтобы понять, что разработчики работают очень по-разному.

Вот, например, Вениамин накидывает код с невероятной скоростью и дикой кучей ошибок. Он получает большое количество замечаний на ревью и от тестировщиков, но исправляет их тоже мгновенно. За всё время работы так и не утихают споры на тему, не быстрее ли писать код сразу нормально. Но команда дружно просит не давать ему на реализацию новую сложную функциональность. Вениамин — бог быстрого багофикса.

А вот Сергей программирует так, что его код можно спокойно сразу отдавать на поддержку маньяку-убийце. Сергей продумывает всё и вся, и тестировщики считают тикеты от него личным вызовом. Правда, лучше не просить его быстренько закрыть критичный баг. Сергей делает работу долго. Но качественно. Но долго.

Вы знаете, какие именно разработчики вам нужны. Поэтому вам нужно попросить кандидата написать что-то прямо на интервью. Это должна быть совсем небольшая задача. Реализовывать её лучше на бумажке, так как на бумажке это будет быстрее, и кандидат не будет отвлекаться.

Классическая задача, которую я использую — это FizzBuzz. Она формулируется следующим образом:

Напишите программу, которая выводит на экран числа от 1 до 100. При этом вместо чисел, кратных трем, программа должна выводить слово «Fizz», а вместо чисел, кратных пяти — слово «Buzz». Если число кратно и 3, и 5, то программа должна выводить слово «FizzBuzz». Это программная реализация детской игры, в которой игроки должны по очереди называть числа или слова «Fizz», «Buzz» и «FizzBuzz» по тем же правилам.

Решение этой задачи простое и кандидат может написать его за минуту или две. В интернете есть куча обсуждений на эту тему и много разнообразных решений на разных языках. Вот пример такого решения на C#:

```
for (int i=1; i<=100; i++)
{
    string output = "";
    if (i % 3 == 0)        output += "Fizz";
    if (i % 5 == 0)        output += "Buzz";
    if (output.Length == 0) output = i.ToString();
    Console.WriteLine(output);
}
```

Я смотрю, как кандидат подходит к решению этой задачи. Вот он написал в сторонке начало выходной последовательности, потом каким-то псевдокодом набросал решение, прокрутил вручную несколько итераций, отмечая ручкой текущую операцию, внёс пару исправлений, перепроверил и переписал всё на нормальном языке программирования. Потом пара вопросов от меня («А не сложно ли будет в вашем коде заменить 5 на 7?»), и мы можем идти дальше.

В интернете многие пишут, что эта задача просто раскрыла им глаза, что 90% соискателей в принципе не могут написать рабочий код или тратят на него больше 15 минут. На моей практике результаты не настолько печальные. Действительно, многие разработчики впадают в лёгкий ступор и тратят неразумное время на решение этой задачи даже с моими подсказками. Но таких мне встречалось не более 20%. Возможно, отбор рекрутёров в моих компаниях был лучше, чем в среднем по отрасли.

Я не принимаю на работу программистов, которые не справляются с такой задачей. В моих проектах от программистов требуется написание бизнес-логики, иногда довольно сложной, поэтому бессмысленно нанимать человека, который явно не способен реализовывать алгоритмы.

Можно, после провала с FizzBuzz, дать второе задание, аналогичное, чтобы подтвердить, что проблема действительно существует и не связана с этой конкретной задачей. Но в моей практике, когда программист не мог написать FizzBuzz, он не мог написать и вторую практическую задачу.

Есть очень много возмущённых отзывов на тему, что человек не может называться программистом, если он не может написать FizzBuzz. Я более спокойно отношусь к этому. Да, любой выпускник профильного ВУЗа должен уметь решить эту задачку. Да, решение этой задачи включает только очень базовые навыки программирования, и печально, что не все с ней справляются. Но надо признать, что есть много компаний, где от разработчика не требуется реализации хоть сколько-нибудь сложных алгоритмов.

Люди работают и получают хорошие деньги, не умея писать программы. Встречаются очевидные и очень печальные пробелы в знаниях в казалось бы базовых областях (теория сложности, структуры данных, SQL, логика), но это не ставит крест на кандидатах навсегда. Когда у них будет достаточная мотивация, они подтянут знания в этих областях.

Вот только я принимаю решения о найме здесь и сейчас. Слишком высок риск принимать таких людей в мои проекты, поэтому я не беру программистов, которые не умеют писать код. Возможно, у ваших проектов другие требования.

Кроме полной неспособности решить задачу FizzBuzz встречается много других ситуаций. Например, человек может забыть, как проверять делимость чисел. Я ему тогда просто подсказываю. Здесь важно, чтоб он сразу признался, не тратя время понапрасну. В условиях стресса на собеседовании можно забыть что угодно.

А иногда кандидат пишет код, но не может найти в нём ошибку, если я говорю, что она есть. То есть кандидат не может отлаживать код в голове, он не может его полноценно читать. И это уже проблема, которая будет проявляться в его работе.

А некоторые кандидаты пишут и переписывают код этой крошечной программы по несколько раз и результат выглядит крайне неаккуратно. Не стоит надеяться, что в реальных программах будет по-другому. Вам, скорее всего, придётся потратить время, чтобы научить кандидата писать хорошо.

Нужно иметь в запасе как минимум два варианта подобной простой программистской задачи. Вторая элементарная задача, которую я использую на собеседованиях, это вывести в консоли «пирамидку» заданной высоты:

```

      x
     xxx
    xxxxx
   xxxxxxx
  xxxxxxxxx
 xxxxxxxxxxx
xxxxxxxxxxxxx
```

## 2.4 Задачи на собеседовании: Цикл в списке

Эта задача гораздо сложнее, чем FizzBuzz, которая может считаться разминочной. Я сам когда-то получил эту задачу на собеседовании и срочно включил её в свой арсенал. Задача формулируется так:

Написать метод, который определяет, есть ли цикл в односвязном списке.

Задача нарочито формулируется без всяких ненужных уточнений. Такая формулировка достаточна для понимания. А если кандидат, например, не знает, что такое «цикл» или «односвязный список», то он просто может задать вопрос.

Ситуация, с которым должен работать требуемый метод может быть иллюстрирована вот таким хороводом символов Марса:

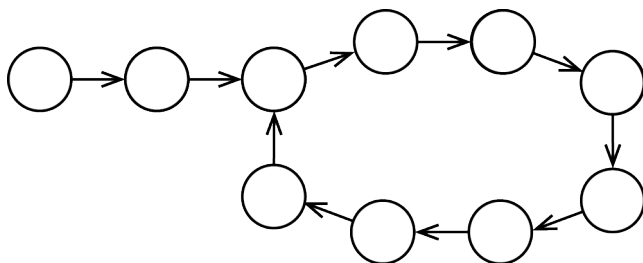


Рис. 2.2: Односвязный список с циклом

Существует два типа решений этой задачи:

**Через два указателя.** Помещаем два указателя в начало списка. На каждой итерации двигаем первый указатель на 1 шаг вперёд, а второй — на 2. Если второй указатель «догоняет» первый, значит, цикл есть.

**Через запоминание пройденных узлов.** На каждой итерации запоминаем текущий элемент, проверяем, не запомнили ли мы уже следующий элемент, и сдвигаем текущий элемент на следующую позицию.

Первое решение вы вряд ли услышите, разве что кандидат будет с очень хорошей алогоритмической подготовкой и опытом решения олимпиадных задач по программированию. Второе решение имеет интересный вариант, когда дополнительная память не используется, а проверка пройденных элементов делается просто проходом по списку с начала. Если кандидат озвучил такое решение как один из вариантов — это хорошо.

Оба решения требуют очень аккуратного кода, чтобы избежать `Null Reference Exception`.

К обоим решениям стандартный дополнительный вопрос: посчитать вычислительную сложность получившегося алгоритма. Здесь традиционно возникает вопрос алгоритмической сложности поиска в хэш-таблице (любимой структуре хранения данных кандидатов) и традиционный ответ, что это  $O(1)$ . В Википедии, например, указано более правильно, что  $O(1)$  достижимо только «в среднем», а так-то может быть до  $O(n)$ .

В общем, несмотря на простоту формулировки и решения эта задача предоставляет много возможностей для обсуждения деталей с кандидатом.

К сожалению, абсолютное большинство кандидатов имеют слишком слабую алгоритмическую подготовку, чтобы осмысленно подходить к выбору структур данных для решения этой задачи с точки зрения затрат времени и памяти. Тем не менее, эта задача позволяет более глубоко оценить знания кандидата, так как кандидату, кроме реализации логики нужно:

- Реализовать какую-то структуру классов;
- Очень аккуратно работать с указателями;

- Принимать во внимание граничные случаи (пустой список, список без цикла);
- Объяснить решение (без чертежа это сложновато).

## 2.5 Проверка тестовых заданий

Проверка тестовых заданий не сводится к простановке плюсики за выполнение и минусика за ошибку. Во время выполнения задания вам также не стоит просто скучать. Наблюдение за кандидатом даст важную информацию о работе кандидата над реальными проектами.

Например, задаёт ли кандидат вопросы? Хорошо, когда кандидат не имеет никаких вопросов и сразу выдаёт приемлемый результат. Ещё лучше, когда он быстро обдумывает задание и предлагает пару уточнений. И совсем другое, когда кандидат молча замирает на пару минут, а когда вы спрашиваете его, как дела, оказывается, что у него есть какой-то вопрос, который он стесняется задать. Можете не сомневаться, что в реальной работе вас ждут те же проблемы, помноженные на сложность проектных задач.

Другой проблемой может быть ситуация, когда кандидат не решается начать работу, задавая вопросы, которые не имеют прямого отношения к заданию. «Надо ли создавать структуру классов? А имена можно любые давать? А можно сперва черновик сделать, а потом начисто переписать? А вопросы задавать можно?»

Стоит обратить внимание, использует ли кандидат пометки и чертежи для себя, чтобы написать алгоритм. Если кандидат не делает никаких записей и при этом путается в реализации, то это тревожный звоночек. Видимо, он не привык реализовывать сложные алгоритмы.

Также важно, использует ли кандидат рисунки, чтобы объяснить ход своих мыслей. Вам стоит побыть непонятливым, чтобы увидеть, насколько кандидат может быть понятным. Вспомните, сколько раз вы видели ситуацию, когда заказчик не мог понять техническую проблему, объясняемую разработчиком. Разработчики в таких ситуациях обвиняют заказчиков, считая их упёртыми и глупыми, но обычно проблема в объяснениях.

Лучше прямо попросить кандидата объяснить что-нибудь детально. Например, он говорит, что вычислительная сложность алгоритма  $O(n)$ . А может ли он это показать? Можно так и спросить: «А вы могли бы это как-то объяснить так, чтобы объяснение можно было показать заказчику, и тот бы понял?» Навык просто объяснять сложные вещи очень полезен.

И, конечно, любые проблемы, которые вы видите у кандидата, не должны восприниматься как нечто окончательное и непреодолимое. Например, кандидат «завис» при решении задачи. Понятно, что это плохо. Но что именно плохо? Он боится задавать вопросы? Или он пасует при возникновении любых проблем?

Даже полная неспособность решить задачу является хорошим поводом спросить кандидата: «А что помешало в решении?» Но это мы уже углубляемся в область нетехнической части собеседования, о которой мы поговорим чуть позже.

## Про красавицу Марину

Как-то надо было мне нанять в свой проект тестировщика. Искал я его уже довольно давно и безуспешно. Рынок пустой, свободных кандидатов нет. И менеджер соседнего проекта, Таня, тоже была в поисках тестировщика. Мы уже отчаялись найти готового специалиста и готовы были брать junior'a, если кандидат хороший.

Собеседовали кандидатов одновременно в два проекта. Причём я был в худшем положении, так как Таня собеседовала кандидатов сама, а от моего проекта техническое собеседование проводил лид тестирования, Владимир, очень хороший специалист, но не очень опытный интервьюер. Мне надо было проводить ещё дополнительное собеседование, а Таня могла принять решение сразу на первом интервью.

Я смирился, что первый хороший кандидат уйдёт Тане. Но зато, успокаивал я себя, Владимир получит с ней хороший опыт собеседований. А собеседовать junior'ов особенно тяжело. Я специально говорил Владимиру: «Не уделяй сильно много внимания опыту. Гляди, чтобы глаза у кандидата горели. Ищи такого кандидата, на обучение которого тебе не жалко будет тратить своё время».

И вот на горизонте появилась кандидат, Марина. Она не имела опыта тестирования, но по словам рекрутеров, была очень мотивирована и рвалась самообучаться и работать. Так что интервью было назначено, и Владимир с Таней ушли собеседовать Марину. После собеседования Владимир зашёл ко мне в очень радостном настроении и сказал, что Марину нужно брать. Она прекрасно впишется в команду. Он готов её учить всему. Она тоже готова учиться. В общем, лучший junior на планете.

Я обрадовался и отписался рекрутерам, чтобы они назначали интервью со мной на ближайшее время. Хотя радость моя сдерживалась уверенностью, что Таня сейчас отпишется, что она нанимает Марину сама. Но время шло, а Таня не отписывалась. Это было странно. После слов Владимира я ожидал, что Таня такую кандидатку утянет в свой проект.

Так что я пошёл к Тане и спросил, что там с Мариной, и почему кандидатка, которая так понравилась Володе, до сих пор не нанята.

— Ты бы лида своего женил. — сказала мне Таня, — Марина с ним флиртовала, а Владимир совсем ум потерял от её

красоты. Нанимать я её не буду, так как кроме внешности она никаких плюсов не имеет. Так что если вам нужна такая Марина — берите.

Я Таню поблагодарил и пошёл беседовать Марину сам. Она оказалась правда очень красивой девушкой, уверенной в себе и действительно склонной к флирту. А вот особой склонности к тестированию она не проявила. Пробившись через её улыбки и уклончивые ответы, я выяснил, что ей нужно просто иметь хорошую запись в трудовой книжке, а работать в этой компании сколько-нибудь долгое время она не планирует. Так что финальное моё мнение было ближе к таниному, чем ко мнению Владимира, и нанимать Марину я не стал.

Так я ещё раз удостоверился, что интервью могут быть действительно сложными для собеседующего. И что коварные кандидаты не чураются наносить удары ниже пояса.

А перед беседами лучше вступить в брак.

## 2.6 Технический проект

Во многих компаниях обязательной частью приёма на работу программиста является выполнение большого технического проекта. Такой проект похож на реальные проекты компании и включает бизнес-логику, UI, backend, и часто состоит из нескольких отдельных модулей.

Объём требуемых от кандидата усилий различается от компании к компании и иногда достигает полноценной рабочей недели. Многие кандидаты, возмущенные объемом работы, отказываются продолжать собеседование и пишут негативные отзывы о компании.

Когда проект требует выделения всего лишь одного дня (4-6 часов) реакция кандидатов, в основном, положительная. Люди

готовы потратить столько, если они действительно заинтересованы вакансией.

Кроме потери кандидатов, включение такого этапа в процесс затягивает найм<sup>2</sup>. Даже если проект требует всего лишь 4 часов работы, то, скорее всего, кандидат сможет выделить это время только на выходных. А значит, вам придётся просто ждать.

Проверка сделанного проекта тоже занимает время. Промотреть большой объём кода ничуть не проще, чем провести интервью. Да, к тому же, выполненный проект всё равно нужно обсудить с кандидатом, иначе трудно понять, почему он реализовал его так, а не иначе. То есть нужно будет потратить время ещё и на дополнительную встречу с кандидатом.

Причём многие справедливо указывают на то, что этот дополнительный этап практически бессмысленен. Хороший технический интервьюер может довольно точно определить уровень кандидата с помощью интервью, без таких изматывающих упражнений на кодирование. Так почему же этот этап вообще используется на практике?

В первую очередь, это нужно компаниям, где нет технических экспертов, умеющих проводить интервью. Вы можете сэкономить вашей компании время и деньги, которое она тратит на неэффективные методы оценки кандидатов, просто качественно проводя собеседования вместо ненадёжного и долгого этапа реализации технического проекта.

Другой причиной является желание проверить, насколько сильно кандидат хочет работать на вас. Если он готов потратить несколько дней на проект для вашей компании, значит, он точно серьёзно рассматривает вашу вакансию. И это правда. Но тоже

---

<sup>2</sup>В главе 4.4 «Основное экономическое соотношение найма» на стр.105 мы будем более подробно говорить о том, насколько дорого обходятся компании задержки с наймом сотрудников.

самое может проверить интервьюер на менеджерской части интервью.

К сожалению, такая популярность технических проектов на собеседованиях объясняется тем, что большинство компаний не имеют специалистов, способных качественно провести технические и менеджерские интервью. Оценить готовое приложение и проревьюить код гораздо проще, чем грамотно провести собеседование. Это неэффективно, это отпугивает многих хороших кандидатов, но компания, которая не умеет проводить интервью, не имеет никакой альтернативы.

Есть ли ситуации, когда действительно имеет смысл давать кандидату технический проект? Да, конечно. Один из случаев описан в главе 4.2 «Разработчик, который не хочет менять работу» на стр.98. Если кандидат не уверен, что он хочет работать в вашей компании, он может быть заинтересован в том, чтобы попробовать «реальную работу» без всяких обязательств.

Другая ситуация, где технический проект полезен — это найм junior-разработчика. Если разработчик никогда не писал коммерческий код, то имеет смысл дать ему попробовать свои силы. Для него это будет не нудная часть приёма на работу, а полезное упражнение. Но в любом случае проект не должен требовать больше 6 часов работы. Такой объём уже достаточен для «серьёзного» кода, но не вымотает кандидата.

Кроме этих двух перечисленных случаев я не вижу достаточной ценности в техническом проекте. Недостатки этого этапа найма слишком значительны. Я рекомендую использовать технический проект при найме как можно реже и с очень чётким пониманием, зачем вам нужен этот этап.

## 2.7 Проверка знания английского

Во многих международных компаниях знание английского необходимо и это знание проверяется при приёме на работу в обязательном порядке. И также в обязательном порядке делаются одни и те же две ошибки.

**Первая ошибка.** Многие компании заявляют, что английский критически важен, но берут кандидатов, которые английский не знают и, что хуже, не хотят знать. Это кажется безумием, но кадровый голод в сфере IT очень высок, поэтому очень обидно бывает, когда попадается превосходный специалист, единственным минусом которого является недостаточное знание английского.

Часто такому кандидату после некоторых сомнений молча высылаются офер. Ведь английский реально легко подучить. А кандидаты обычно говорят, что они усердно учат английский и, конечно, хотят подтянуть его уровень<sup>3</sup>.

Потом кандидат принимает офер и выясняет, что с его уровнем владения языком работать ему плохо. Он не может выполнять свои прямые обязанности. У него проблемы с заказчиком. Он не может двигаться по карьерной лестнице. Требуемый уровень языка часто кажется такому кандидату нереально высоким.

Кандидат либо покидает компанию сам, либо его приходится увольнять. Кандидат возмущается, так как с его точки зрения от него требуют невозможного. Вы недовольны, так как с вашей точки зрения кандидат не приложил достаточно усилий. Всем плохо, а компания несёт убытки.

---

<sup>3</sup>Про это написано в главе 3.8 «Социальное одобрение» на стр.89. Не стоит рассчитывать, что если кандидат отвечает, что он любит изучать английский, то он реально любит его изучать. Это не лукавство и не обман со стороны кандидата. Это установка, которая навязывается обществом. Вам придётся приложить дополнительные усилия, чтобы выяснить, готов ли кандидат улучшать свой английский.

Что же делать? Надо очень чётко проговорить с кандидатом, какой уровень знания языка и за какое время нужно получить, а также какие проблемы будут у кандидата, если он этого уровня не достигнет. И до офера нужно получить согласие кандидата с такими условиями (а вы должны поверить в искренность этого согласия). Более детально этот процесс расписан в главе 4.1 «Офер» на стр.95. А необходимость проговорить с кандидатом уровень языка, который он должен достигнуть, ведёт нас к обсуждению второй ошибки.

**Вторая ошибка.** Часто вся тема английского языка отдаётся на откуп преподавателям, к которым, совершенно заслуженно, в IT большое уважение. И они определяют, например, что рядовые сотрудники должны иметь уровень Basic, старшие — Intermediate, а остальные — Advanced<sup>4</sup>. Для этих уровней составляются тесты (или берутся готовые), и все сотрудники, новые и имеющиеся, эти тесты проходят.

Звучит логично? Нет. По очень простой причине: преподаватели не могут знать, какие реально знания нужны в работе, и, соответственно, не могут эти знания проверить.

Например, я видел разработчиков, которые легко читают документацию на английском, грамотно называют методы и даже вменяемо комментируют код, но совершенно теряются, если текст нетехнический. Они могут описать проблему в письме, но не могут грамотно спросить заказчика, что с этой проблемой делать. Они не хотят учить английский, так как в их работе им ничего не мешает. Любое тестирование покажет, что их уровень не дотягивает до Basic, но это не совсем так.

Или вот другая тоже распространённая ситуация. Специалист может общаться с заказчиком на упрощённом и ло-

---

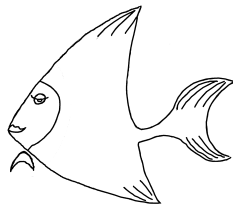
<sup>4</sup>Иногда для полноты картины отдельно вписывают, что генеральный директор вообще может быть неграмотным.

маном языке. Он не употребляет Third Conditional и Past Perfect. Он игнорирует дифтонги и межзубные  $\theta$  и  $\delta$ . Но заказчик может прекрасно его понимать.

Я видел даже менеджеров высокого уровня, которые имеют весьма плохое произношение, но их опыт и знание большого количества стандартных оборотов позволяет им успешно вести переговоры и заключать многомиллионные контракты.

Если предполагается, что кандидат будет устно общаться с заказчиком, то интервьюеру достаточно просто попросить его рассказать на английском о текущем проекте. Это будет достаточно технический разговор, и по нему будет легко понять будут ли у кандидата проблемы в общении. Не пытайтесь сгрузить эту проверку на преподавателей английского. Вы же не приглашаете преподавателя программирования, чтобы определять уровень разработчиков? Умение пользоваться английским такой же навык.

Если текущая команда английский знает плохо, и вы хотите это изменить, то категорически нельзя брать кандидатов с плохим английским. Они не будут улучшать свой язык, они будут ориентироваться на коллег. Вы просто будете увеличивать массу людей с плохим английским. Исправление такой ситуации — это отдельная тема, далеко выходящая за рамки этой книги.





## Глава 3

# Менеджерское собеседование

Технические навыки не говорят ничего о том, будет ли реально нанятый кандидат приносить пользу, и сможет ли он встроиться в команду. Поэтому даже на техническом собеседовании часто больше внимания уделяется тому, как кандидат думает и как он относится к работе, чем тому, какие конкретные умения у него уже есть.

В этой главе мы поговорим о том, как можно определить, что кандидат из себя представляет не как специалист, а как человек.

### 3.1 Дедуктивный метод

В детстве я был просто зачарован Шерлоком Холмсом в те моменты, когда он, базируясь на исцарапанных часах или пыли на пальто, выдавал про человека кучу очень детальной информацией да ещё с присказкой: «Это элементарно, Ватсон!»

Причём все эти выводы объяснялись, и любой мог проверить их обоснованность.

Примерно такое же ощущение я испытал как-то, когда присутствовал на собеседовании одного разработчика из моей команды очень опытным менеджером. После собеседования менеджер сказал: «Нет, этот парень совсем не командный игрок. Это самая главная проблема у него».

Когда я выразил своё удивление, объяснил:

- Посмотри, сколько раз он разделял код на «мой» и «не мой». Про проблемы с «чужим» кодом он даже говорить не стал.
- Когда ты пытался объяснить ему, что не мог найти его на рабочем месте, то он просто не понял, в чём проблема. Он же потом появился и помог — значит, вообще всё хорошо. А то, что ты нервничал и не мог без него сделать свою работу — это не его беда.
- Когда ты сказал, что были проблемы с его кодом, он сразу стал агрессивным и сказал, что будет с тобой по этому поводу разбираться. Нет бы воспринять спокойно и предложить что-то конструктивное.
- Он сказал, что хочет работать из дома по графику, который никто не будет знать, а общаться с другими он вообще не хочет.

После такого списка фактов я не мог не признать, что разработчик действительно не командный. Вспомнив свой опыт работы с ним, я и сам мог прийти к тому же выводу. Да и все названные моменты я слышал своими ушами, но не сумел понять их значение. Хотя когда для меня эти знаки собрали в кучу и показали, истина стала очевидной.

Так как же надо подходить к собеседованию, чтобы самому получать такую же чёткую картину характера кандидата?

Давайте разберём ситуацию из моего опыта. Однажды я спросил кандидатку: «А бывает так, что с какими-то людьми у вас почему-то возникают конфликты?»

Ответ был такой: «Ой, я совсем не конфликтный человек! Я с коллегами очень хорошо общаюсь! И за чаем всегда болтаем про всякие нерабочие вопросы, наши семьи, наш отдых. Это только конфликтные люди со всеми ругаются. К сожалению, есть и такие».

А что это за люди такие? Кандидатка уточнила: «Как-то так получается, что это выскочки, которых повысили за что-то. Лидом человека сделали или менеджером. Нормально общались, а тут, как подменили человека. Почему-то люди очень часто начинают задаваться, хотя ничего из себя не представляют. Сами не знают, чего хотят, а кричат. Я считаю, это звёздная болезнь, прямо эпидемия звёздной болезни».

А можно что-то с этим сделать, как-то избавиться от этих конфликтных ситуаций? Кандидатка и тут дала прямой ответ: «Да вряд ли. Людей не исправишь. Если уж они такие зазвездившиеся, то остаётся только терпеть. Возможно, более высокое руководство может их поставить на место. Но руководству всё равно, они такие же».

На этом этапе я уже получил достаточное понимание конфликтности кандидатки и не стал дальше говорить на эту неприятную тему, чтоб не выводить её из комфортного состояния. В этом не было надобности. Я уже понимал, что кандидатка весьма конфликтна.

Почему? Потому что она сама признала, что есть много людей, с которыми у неё конфликты (лиды, менеджеры, причём, их так много, что «прямо эпидемия»). Это просто определение конфликтности. Если у человека много конфликтов, значит он конфликтный.

Отчего же возникают конфликты у кандидатки? Она про это тоже рассказала открыто. Конфликты начинаются, как

только от неё начинают что-то требовать. Этим как раз отличаются лиды и менеджеры. Они начинают не просто работать рядом, болтая о не относящихся к работе вещах, они начинают давать задания и требовать их выполнения. И сразу на этом этапе с кандидаткой начинаются проблемы, даже если до того общались хорошо (это в точности то, что она рассказала).

Кандидатка не готова сама разрешать конфликты. Единственный выход она видит в том, что кто-то более сильный, более высокий менеджер (которого она тоже не уважает), вступит в конфликт с её противниками и «поставит их на место».

Ругань и конфликты в её понимании являются вполне приемлемым инструментом. И конфликты это именно ругань (она так и поняла вопрос). Причём выбранный ею тон был очень пренебрежительным и эмоциональным. Она про эти конфликты не забывает даже на собеседовании.

Так что можно уверенно утверждать, что с кандидаткой будут проблемы, так как она склонна крайне эмоционально воспринимать рабочие моменты и будет генерировать конфликты.

Но как же так? Ведь можно найти другие объяснения этим ответам! Давайте, попробуем: «В компании специально отбирают весь руководящий состав, исходя из их способности кричать на других людей. При любом повышении в той компании требуют, чтобы повышенный начинал кричать на тех, с кем ранее общался хорошо. Кандидатка понимает, что это неприемлемо, но её там удерживали силой, и вот она вырвалась и устраивается на работу».

Абсолютно неправдоподобное объяснение, правда? Примерно соответствует ситуации, когда к Шерлоку Холмсу пришёл бы человек, нанёсший царапины на свои часы и осторожно присыпавший печной сажой рукав своего пальто.

Здесь нам помогает бритва Оккама (XIV век, проверенный инструмент), в соответствии с которой не стоит выдумывать

запутанные и неправдоподобные объяснения при наличии простого и логичного. Если кандидатка несколько раз показывает признаки своей конфликтности, значит она конфликтная.

Что же нужно, чтобы применять этот метод?

**Заранее определите, что вы хотите узнать.** Не стоит втягивать кандидата в долгие и беспредметные разговоры, чтобы потом ночами сидеть и анализировать сказанное в надежде покрыть все ваши вопросы. Кандидат говорит про то, что вы спрашиваете, а вы должны составить список тем, которые вам нужно прояснить. Мне была важна неконфликтность кандидата, поэтому я исследовал этот вопрос. В другой ситуации важно может быть другое. Можете вспомнить все проблемы, которые вам отравляли жизнь, чтобы составить список тем для интервью.

**Уточняйте.** Вопросы и ответы часто допускают множество толкований. Вам же нужно получить уверенность. Поэтому задавайте дополнительные вопросы до тех пор, пока вы не будете точно понимать, что за человек перед вами. Кандидат может просто неправильно выразиться, а вы делаете далеко идущие выводы. Всегда стоит переспросить, чтобы быть уверенным.

**Обдумывайте ответы кандидата быстро.** Есть две опасности: преждевременно посчитать, что с кандидатом всё ясно, и, наоборот, мучать кандидата вопросами на одну и ту же тему, когда у вас уже есть достаточно информации. Не делайте преждевременных выводов, но и не ищите сверххитрых трактовок ответов кандидатов. Вам нужно очень быстро обдумывать то, что говорит кандидат и принимать решение, задавать ли ещё какие-то вопросы или двигаться дальше.

Планомерное применение этого метода в собеседованиях (и не только) позволит вам гораздо лучше понимать, что люди на самом деле говорят и думают.

Для примера давайте я приведу ответ неконфликтного кандидата на тот же самый вопрос: «А бывает так, что с какими-то людьми у вас почему-то возникают конфликты?»

Развёрнутый эталонный ответ, который я как-то получил, звучал примерно так: «В большой команде всегда есть достаточно много конфликтов. Это конфликты интересов. Менеджер хочет, чтобы работа была сделана как можно более быстро, разработчик хочет, чтобы код был как можно более элегантен, тестировщик хочет, чтобы код работал так, как описано в документации, аналитик хочет, чтоб его не мучили вопросами и не просили менять требования. К счастью, все члены команды имеют одну цель, создание продукта, и это даёт базис, чтобы все эти конфликты разрешить. Нужно быть конструктивным и не давать перерастать конфликтам в словесную перепалку. Методология разработки говорит каждому, что нужно делать, а все конфликты можно разрешить просто поговорив или эскалировав проблему менеджеру».

Нет нужды что-то дальше спрашивать у такого кандидата. Можно переходить с ним к следующей теме.

## Про печального Олега

Олег, судя по резюме, был очень опытным разработчиком. Его резюме впечатляло. Да и на технические вопросы он отвечал очень быстро и исчерпывающе. Вот только психическое состояние Олега вызывало опасения. Он периодически прятал лицо в своих руках. Он отключался от беседы и пропускал вопросы. Когда вопросы требовали большого интеллектуального напряжения, он просто прерывал свой ответ на середине и просил задать следующий вопрос. Любые сложные вопросы в такой ситуации задавать было просто бессмысленно.

В тот раз я проводил техническое интервью и после меня Олега ещё должен был собеседовать мой менеджер. Я ему

выслал свой отчёт, честно сказав, что хотя программист Олег, наверное, хороший, но в его текущем состоянии работать с ним невозможно. Мой менеджер почесал затылок и назначил своё интервью. На втором интервью Олег, по словам менеджера, вёл себя вполне нормально, но, поотвечав на вопросы, сказал, что он решил сменить место проживания и что он уезжает в другой город.

Я был недоволен результатом интервью, так как мне не удалось понять, почему Олег вёл себя так странно. Открытого разговора с Олегом не получилось, и никакие разумные объяснения мне в голову не приходили. К счастью, даже в миллионном городе круги общения разработчиков очень тесно пересекаются. Через полгода меня пригласил директор и сказал, что ему удалось получить объяснение «загадке Олега».

Оказывается, у Олега была размолвка с его девушкой, Машей. Маша обиделась на что-то, не отвечала на его звонки. Олег, конечно, беспокоился и пытался достучаться до Маши без всякого успеха. К слову, Маша работала в моей компании. Так что желание Олега устроиться к нам на работу, возможно, имело целью быть поближе к Маше.

Вот только явившись на собеседование, Олег лицом к лицу столкнулся с Машей, идущей под ручку с другим молодым человеком. За то время, пока я шёл в переговорку, Олег с Машей успели устроить бурную сцену объяснения. Олег выяснил, что между ними всё кончено, и что у Маши уже новый роман. Параллельно они наговорили друг другу неприятного, и Маша убежала.

Так что Олег пережил очень большое потрясение прямо перед нашим с ним интервью. Не удивительно, что собеседование потеряло для него всякую привлекательность. Удар по нервам был такой, что никакого дополнительного напряжения на интервью он не мог выдержать. К тому же желание быть поближе к Маше и работать в нашей компании у него плавно сменялось

желанием быть от Маши подальше. Поэтому он даже решил переехать в другой город.

Так что загадка Олега разрешилась. А я ещё раз убедился, что даже для очень странного поведения кандидата может быть вполне логичное объяснение.

## 3.2 Анализ коннотаций

В живом разговоре всегда есть два слоя: фактический и эмоциональный. Эти слои тесно связаны. Мы говорим о том, что происходит и параллельно выражаем своё отношение к этому. Для людей эмоции представляют гораздо большую ценность, чем сами факты, поэтому часто эмоциональный слой содержит гораздо больше информации, чем фактический. Даже в сугубо профессиональной коммуникации эмоциональный слой очень важен.

Можете вспомнить обсуждение какого-то неприятного бага в сработанной команде. Эмоции там настолько выражены, что чтобы избежать обид их направляют на неодушевлённые предметы (сервер, код, приложение, сам баг, документацию) и людей, не присутствующих в разговоре (заказчик, пользователь).

Эмоциональный слой настолько важен, что часто эмоциональная окраска даёт полностью противоположное значение используемым словам. Например, фраза, сказанная с сарказмом и без — это две фразы с противоположным смыслом, хотя слова в них одни и те же: «Да, конечно, я исправлю этот баг за пару часов».

Это кажется очевидным, но на практике многие собеседующие совершенно игнорируют эмоции в разговоре, упуская 90% того, что кандидат действительно им сообщает. Помните пример с конфликтной кандидаткой на странице 59? Очевидно ли вам, что та кандидатка говорила о коллегах довольно злобно

и с явным негативом? Если нет, то давайте выпишем все слова с негативной эмоциональной окраской, которые она использовала (я выделил их жирным шрифтом). Обратите внимание, что чем дальше, тем больше негатива вскрывается:

Ой, я совсем не конфликтный человек! Я с коллегами очень хорошо общаюсь! И за чаем всегда болтаем про всякие нерабочие вопросы, наши семьи, наш отдых. Это только конфликтные люди со всеми **ругаются**. К сожалению, есть и такие.

Как-то так получается, что это **высочки**, которых повысили за что-то. Лидом человека сделали или менеджером. Нормально общались, а тут, как подменили человека. Почему-то люди очень часто начинают **задаваться**, хотя **ничего из себя не представляют**. Сами не знают, чего хотят, а **кричат**. Я считаю, это звёздная **болезнь**, прямо **эпидемия звёздной болезни**.

Людей **не исправишь**. Если уж они такие **звездившиеся**, то остаётся только **терпеть**. Возможно, более высокое руководство может их **поставить на место**. Но руководству **всё равно**, они **такие же**.

Если игнорировать эмоции, то очень легко не увидеть действительного сообщения кандидатки. Так интервьюеры довольствуются фактическими утверждениями («я совсем не конфликтный человек», «болтаю с коллегами обо всём») и пропускают мимо ушей то, что человека реально тревожит («есть люди, которые на меня ругаются»).

Эти дополнительные, «эмоциональные» значения используемых человеком слов называются коннотациями, необходимо уметь их слышать, видеть и учитывать.

Если для вас всё, описанное выше, очевидно, то я вас могу только поздравить. Значит, у вас развит эмоциональный интеллект, вы хорошо понимаете чувства других людей и можете это

использовать. Анализ коннотаций вы проводите автоматически, на лету, не прикладывая больших усилий.

Если же вы не уверены в своих возможностях, то для тренировки делайте такой анализ письменно. Можете для тренировки взять какое-нибудь обсуждение на форуме, вроде «iOS против Android» и подчёркивать одной чертой все слова с положительными коннотациями и двумя с отрицательными. Уверен, что текст будет очень густо расчерчен.

Помните эти нудные школьные сочинения на тему «Что чувствовал Обломов к своему дивану?» Так вот навыки, полученные в то время, нужно развивать дальше. Это реально поможет вам в вашей карьере.

Кстати, обратите внимание, что слова с негативными коннотациями сами по себе не являются чем-то плохим. Важно, на что обращены эти негативные эмоции и как кандидат с этими тревожащими его ситуациями привык справляться.

Очень хорошо, когда человек использует и слова с позитивными коннотациями, и слова с негативными коннотациями. Это показывает открытость кандидата, его готовность говорить о том, что он чувствует. Такой человек прямо говорит, что ему не нравится и что ему нравится, чтобы вы могли понять, подходит ли он вам и подходит ли ваша компания кандидату.

Причём даже если вы хорошо чувствуете эмоции, вам придётся постараться, чтобы вывести кандидата на открытый разговор. Если вы просто спросите кандидата: «Расскажите про проекты, на которых вы работали,» то вы получите протокольное перечисление проектов и использованных технологий. Со всем другой результат будет, если вы спросите: «Расскажите про самый любимый ваш проект. Не важно, недавний или совсем старый. Просто проект, на котором вы бы с радостью поработали ещё раз».

Используйте эмоционально окрашенные слова сами, чтобы кандидату было легче проявлять свои эмоции.

В идеале любая обсуждаемая с кандидатом тема должна содержать эмоциональную составляющую, чтобы вы лучше понимали кандидата. Но вот, для затравки, примеры вопросов, которые могут помочь выяснить отношение кандидата к работе:

- Почему вы хотите сменить работу?
- Что для вас интересный проект? Что для вас неинтересный проект? Какой ваш проект был самым интересным?
- С какими людьми вам тяжело общаться?
- Почему вам нравится программировать?
- Вспомните самого лучшего вашего начальника. Почему он лучший?

## Про агрессивного Алексея

Пришёл ко мне как-то на собеседование разработчик Алексей. Он работал в одной из соседних компаний на должности лида разработки. То есть у него хватало и опыта, и технических знаний, и понимания процессов. Алексей был довольно активным, отвечал чётко и быстро.

Но через некоторое время я заметил странности в его ответах. На некоторые вопросы он очень бодро докладывал мне абсолютный бред. То есть было очевидно, что знаний у него по заданному вопросу нет, а текст, который он выдавал, больше походил на случайный набор терминов. Бывает иногда, что человек путает индекс в базе данных и индекс в массиве и отвечает не то. В таких ситуациях я обычно просто иду дальше. Но здесь уж очень много таких ответов было.

И когда в следующий раз я получил от Алексея непонятный ответ, я начал копать глубже. Я объяснил свой вопрос заново более чётко и сказал, что я не понимаю, как ответ Алек-

сея относится к этому вопросу. И результат был неожиданным. Алексей покраснел, его кулаки сжались, он ажно привстал со своего места и очень агрессивно и громко стал повторять свой ответ. Я не понимал, что происходит.

— Что такое транзакции в базах данных? — спрашивал я Алексея.

— Транзакциями называются специальные объекты в базах данных, для работы с которыми используются особые классы, — отвечал он.

— Алексей, но ведь это не ответ на вопрос. Это же не объясняет, что такое транзакции. Вы не знаете, что это такое? Может, просто дальше пойдём? Или, может, вы другими словами объясните?

— Я уже ответил на вопрос!

Любая моя попытка уточнить ответ или объяснить Алексею, почему его ответ меня не устраивает, вызывали приступ гнева с его стороны. Я был максимально корректен, но это не помогало. Некоторые вопросы вызывали неадекватные ответы, а попытки разобраться в этих ответах вызывали гнев.

Довольно скоро я понял, что Алексей не может ответить «Не знаю». В принципе. Ни на какой вопрос. К моменту осознания этого факта я уже довольно хорошо понимал, в каких областях Алексей «плавает», поэтому для проверки своей гипотезы я задал несколько реально сложных вопросов из этих областей. Получил в ответ порцию откровенного бреда. Со стороны это смотрелось, наверное, сюрреалистически. Кандидат генерировал поток слов, щедро пересыпанных техническими терминами, но абсолютно не имеющих никакого смысла.

Такое нежелание признаться в собственном незнании мешает в работе. Решение любой проблемы начинается с определения, что мы про неё знаем, что мы про неё не знаем, и что мы должны сделать, чтобы знать больше. Нормально, если разработчик скажет, что не знает, почему лежит продуктовый сервер,

и что ему надо часа два, чтобы разобраться. Но если вы не можете получить ответ «не знаю» от разработчика, то вы не сможете понять, что делать с проблемой.

Я решил разобраться, как Алексей работает на его текущей позиции, если он не может признаться в собственном незнании. И оказалось, что руководство компании Алексея не интересуется причинами любых проблем. Оно объявляет разработчиков виноватыми и кричит на них изо всех сил. Решение любых вопросов сводится к тому, чтобы о них не узнало руководство. Защититься можно, если так же сильно кричать в ответ. Что именно кричать — неважно, так как всё равно никто не хочет разобраться, а все хотят найти виноватых. Использование технических терминов помогает, так как руководство их не понимает. А признание в незнании чего угодно, вызывает обычные крики со стороны руководства, так как незнание — это какая-то проблема, и в ней, очевидно, виноваты разработчики.

Алексей в этой компании проработал много лет, поэтому он привык к такому образу действий и выработал защитные механизмы, позволяющие ему существовать с максимально возможным в таких условиях комфортом. Меняться Алексей не хотел, так как проблемы не видел. С его точки зрения он действовал эффективно. В то, что бывают другие руководители и другие способы решения проблем он не верил. Хмыкнул, сказал, что руководитель, который хочет решить проблему, а не найти виноватого — это пропаганда какая-то. Такое в книжках пишут, а в жизни так не бывает.

Это собеседование заставило меня очень сильно задуматься о том, как сильно меняется человек в зависимости от условий работы. И о том, как важно найти для себя компанию, где не только будет хорошая зарплата и интересные проекты, но и где твоя психика будет развиваться в нужном тебе направлении.

### 3.3 Технические вопросы от менеджера

Очень часто кандидата отдельно собеседуют технический эксперт, который проверяет его технические знания, и менеджер, который исследует общий профиль кандидата, его soft skills, его мотивацию и так далее. И неоднократно я видел в менеджерах нежелание задавать технические вопросы. Зачем? Ведь эти знания уже проверил эксперт! Менеджеры не хотят тратить своё время. А иногда они боятся показаться незнающими или просто не уверены в своей возможности оценить технические ответы кандидата.

Я настойчиво рекомендую менеджерам задавать технические вопросы (равно, как и техническим экспертам задавать вопросы «менеджерские»). Дело в том, что в реальных проектах менеджерам часто приходится глубоко влезать в технические детали. Им нужно понимать, что происходит в разработке и понимать это на уровне достаточном, чтобы принимать решения, объясняться с заказчиком и отслеживать, насколько действительность соответствует планам.

Менеджеры на практике задают такие вопросы, которые, хотя и являются техническими, но, обычно, не покрываются техническим интервью. Эти вопросы являются «промежуточными» между техническими и менеджерскими.

Например, если эксперт хочет узнать, умеет ли кандидат анализировать и улучшать производительность приложений он спросит про инструментализацию и профайлеры, он спросит про опыт в улучшении производительности, он спросит про подходы в построении высокопроизводительных приложений с нуля.

Но это совсем не то, что нужно знать менеджеру. На менеджерских собеседованиях я задаю вопрос: «Заказчик пишет

вам гневное письмо, утверждая, что ваше приложение работает медленно. Что вы будете делать?»

Этот вопрос покрывает совсем другую сторону работы разработчика. Даже если кандидат прекрасно умеет оптимизировать приложения, в реальной жизни задачи не приходят готовыми, «на блюдечке с голубой каёмочкой». Перевести задачу из практической, неформальной формы в форму, пригодную для решения — это часто тоже задача разработчика.

Например, с одним из кандидатов получился следующий диалог в ответ на вышеприведённый вопрос о медленном с точки зрения заказчика приложении. Заказчик недоволен, что будете делать?

— Я попрошу организовать митинг с заказчиком, — ответил кандидат.

— Отлично. И что вы на этом митинге будете делать? — начало ответа мне понравилось. Когда информации нет, то это хорошая идея поговорить напрямую.

— Я попрошу заказчика показать проблему. Может быть её и нет.

— И дальше?

— Всё, на митинге делать нечего. Пойду анализировать код.

Я прямо представил себя на таком митинге, где заказчика просят показать проблему, а потом разработчик убегает, и я остаюсь один на один с заказчиком, чтоб объяснить ему, что происходит.

Гораздо спокойней я бы чувствовал себя с разработчиком, который:

- Вместе с приглашением на митинг спросил бы у заказчика, в какой части приложения возникает проблема.
- Просмотрел бы ещё разок, какие требования по производительности у нас были.

- Посмотрел бы , какие требования к оборудованию должны быть для нормальной работы приложения.
- На митинге не позабыл бы спросить у заказчика, что для него «быстро» и «медленно».
- Приготовил бы к митингу пункты, которые нужно проверить (конфигурация оборудования заказчика, загрузка сети, загрузка сервера БД и т.д.).
- Был бы готов сообщить заказчику, как мы будем исследовать и решать эту проблему.

Многие из этих пунктов может сделать и менеджер, но они должны быть основаны на технических знаниях, на знании архитектуры приложения. Для того, чтобы разработчик стал высокооплачиваемым, ему нужно быть готовым такие проблемы решать.

При обсуждении технических вопросов играют роль не собственно технические знания, а способность кандидата использовать эти знания в условиях, приближенных к реальным. Я беру случаи из своей практики, выкидываю из них ненужные подробности и получаю ситуации для обсуждения: «Заказчик просит ускорить приложение», «Вам нужно исправлять баги в большом legacy проекте с кучей технического долга», «Вы делаете ревью кода и заметили, что вся команда делает одни и те же ошибки». Когда кандидат будет описывать, что он будет делать в этих ситуациях, он должен будет использовать свои знания из разных технических областей и свои soft skills.

Если вы не сможете эффективно обсуждать подобные ситуации в тепличных условиях собеседования, то будьте уверены, что в реальной работе вас будут ждать большие проблемы.

## Побег из курятника

Из компании, которая описана в истории «Про агрессивного Алексея» на стр.67 потянулась целая череда разработчиков. Зная, как там организуется работа, я этому не удивлялся. Собеседования я проводил совместно в лидом соседнего проекта, Евгением, с которым мы уже долго вместе работали и с которым процесс совместных собеседований был отлажен.

Первым пришёл совсем слабый junior-разработчик. Он выслушивал наши вопросы, улыбался, пытался что-то ответить, но получалось у него так себе. Практические задания он совсем не сделал. Хотя парень был активным. Он задавал много вопросов по тем темам, которые были ему наиболее непонятны, и расстались мы с надеждой, что через пару лет сможем поработать вместе.

Второй разработчик был уже сильнее. На некоторые вопросы отвечал бодро. Техническое задание, с которым не справился junior, выполнил хорошо. На более сложном задании, правда, уже совсем ничего сделать не смог. Но тоже был активным, и мы хорошо поговорили.

Третий разработчик практические задания выполнил отлично. С теорией было похуже. Он хорошо отвечал на основные вопросы, но когда я начинал копать глубже, то выяснял, что понимания ему не хватает. Как-то царяпнуло меня такое несоответствие знания теории и практики. Но Евгений первым догадался, что происходит. «Ты знаешь, Константин, — сказал он мне, — А ведь эти кандидаты сливают своим товарищам все наши вопросы и задачи».

Я немного подумал, и правота Евгения стала для меня очевидной. С теоретическими вопросами это не было проблемой, так как мы проверяли понимание, а не знание каких-то формулировок. Так что единственное, что нам нужно было поменять в процессе, чтоб не выслушивать заученные ответы — это сразу

начинать со сложных вопросов, ответы на которые посмотреть нигде нельзя.

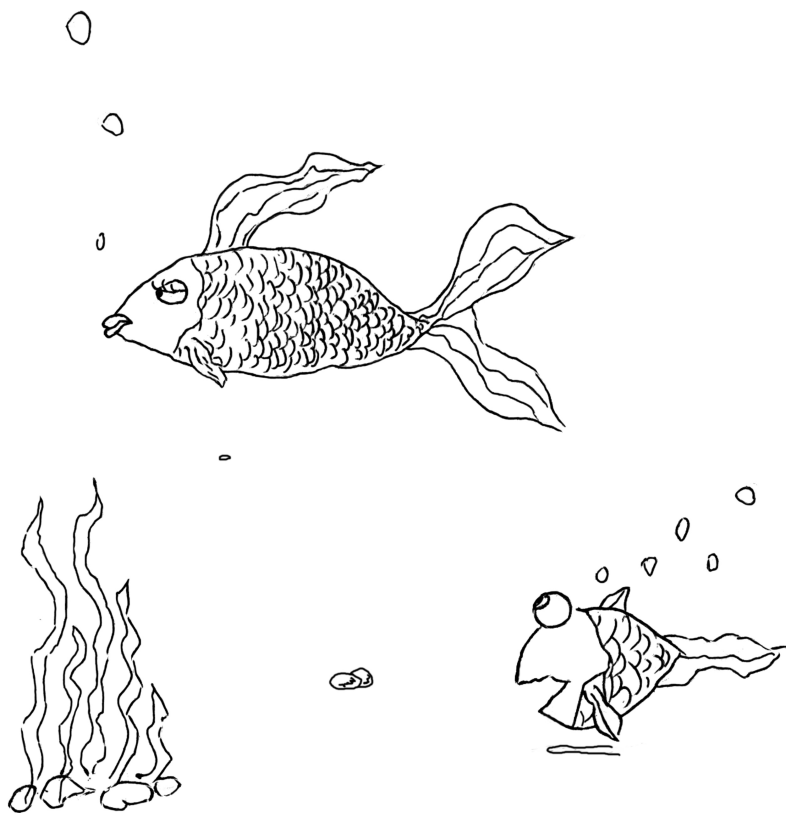
А вот с практическими задачами было гораздо сложнее. В моём арсенале было буквально две любимые задачки, и кандидаты уже наверняка их совместно прорешали дома. Когда кандидат, не задумываясь, пишет заранее подготовленное решение, это не даёт возможности понять ход его мыслей. Поэтому нужно было придумать что-то новенькое. Причём придумать надо было быстро, так как четвёртый кандидат был назначен буквально через полчаса после третьего. В результате пока Евгений гонял четвёртого кандидата по теории, я в быстром темпе сочинял практические задания.

С тех пор я имел запас практических задач на случай, если кандидат придёт с «домашней заготовкой». Ну и обращал внимание на то, были ли недавно кандидаты из той же компании.

### **3.4 Вопросы на собеседовании: Изучение нового**

Очень часто в работе нужно изучать абсолютно новую область. Тестировщики изучают автоматизацию и тест дизайн. Разработчики изучают новые технологии. Аналитики изучают новые инструменты и языки описания систем. Все изучают новые методологии разработки. Очень важно знать, может ли кандидат взять новую для себя область и стать в ней экспертом в короткие сроки.

Как всегда, лучшим способом определить, может ли кандидат быстро учиться, будет просто задать прямой кандидату: «Предположим вам нужно выучить совершенно новую область для вас (библиотеку распознавания образов, например). Как вы подойдёте к этому?»



Хороший ответ, который я получаю на такой вопрос, звучит примерно так:

«Я попробую найти коллегу, который знает эту область, и спрошу его, как лучше всего её изучать и на что стоит обратить внимание. Если такого коллеги нет, то я могу найти специалиста в интернете. Например, задать вопрос какому-нибудь эксперту на форуме.

Обычно стоит начать изучение с каких-нибудь tutorials, чтобы реализовать что-то реально работающее. В процессе обязательно возникнут какие-то вопросы и проблемы. Их решение можно найти в Google или с помощью знакомых экспертов. Когда уже будут какие-то навыки, лучше начать писать реальный проект. Реализация будет неоптимальной и, наверняка, придётся много переписывать, но зато выучишь именно то, что тебе надо, и будет виден прогресс.

Чтобы глубоко изучить эту нужную область, можно найти какую-нибудь обстоятельную книгу. А ещё сейчас много обучающих сайтов, вроде Coursera, где можно найти нужный курс».

В таком ответе мне нравится сразу несколько моментов:

- Кандидат не надеется только на себя, а сразу хочет найти кого-то, с кем можно посоветоваться. Видно, что человек готов работать в команде. В новой области он не сможет сразу стать экспертом, но, пользуясь «коллективным разумом», он сможет принимать разумные решения.
- Перечислено много разных подходов. Какой-то один может не подойти, но что-нибудь да сработает.
- Ориентация на практику. При таком обучении, есть надежда, что проект начнёт получать какую-то пользу в мини-

мальные сроки. Причём итеративный подход к обучению мне тоже нравится. Кандидат готов ошибаться, учиться на своих ошибках и двигаться вперед постепенно.

Для примера могу привести ответ, который я слышал от одного из кандидатов и который я считаю проблемным:

Лучший способ что-то выучить — это книга. Я возьму хорошую книгу и прочитаю её от корки до корки. Только в этом случае можно быть уверенным, что я действительно имею хорошую базу. Я так выучил всё, что знаю. Например, чтобы выучить C++ я взял классический труд Страуструпа и долго читал его, чтобы узнать все детали языка.

В каких-то областях так можно учиться, но в моей практике обычно приходится изучать что-то, для чего просто нет хорошей литературы<sup>1</sup>. Да и читать серьёзную книгу часами тяжело и неэффективно. Можно потратить много недель, а потом выяснить, что на практике применить полученные теоретические знания не так-то и просто. Поэтому с таким кандидатом я не могу рассчитывать на быстрое изучение нового. Я могу только надеяться на те знания, которые он уже имеет.

Очень хорошо кандидата спрашивать о изучении тех областей, в которых у него проблемы. Например, у кандидата проблемы с английским языком. Но он говорит, что готов быстро закрыть все пробелы. Пусть он достаточно хорошо знает грамматику, но испытывает трудности с говорением, а ему нужно задавать вопросы и получать ответы на английском. Этот навык действительно легко развить, но справится ли конкретный кандидат с этим? В такой ситуации я спрашиваю напрямую, как он планирует это сделать.

Однажды я получил такой ответ:

---

<sup>1</sup>Это одна из причин, по которым я решил написать эту книгу.

Я готов вкладываться в своё обучение. Я пойду на платные курсы английского языка. Сейчас много таких. Оплачу курс и стану учиться вечерами.

Такой ответ для меня неубедителен. Желание платить свои деньги и тратить своё личное время, чтобы чему-то научиться, очень похвально. Но позиция кандидата очень пассивная. Он ожидает, что кто-то его научит, но сам он учиться, похоже, не умеет. Мы с ним обсудили очень конкретную проблему, и решение для этой проблемы должно быть очень конкретное. Следующий ответ был бы гораздо лучше:

Я пойду в несколько местных обучающих центров и посоветуюсь, как лучше мне лучше подтянуть английский. Но чтобы улучшить говорение, надо его практиковать. Так что я постараюсь найти кого-нибудь с кем можно говорить по-английски. Есть специальные сайты, где можно найти собеседника из других стран. У меня есть знакомые с хорошим английским, попробую попрактиковаться с ними. Кроме того, мне надо улучшить восприятие английской речи на слух. Например, слушать сериалы на английском языке. Если буду заниматься по часу-два каждый день, то через пару недель эффект уже будет значительный.

В таком ответе есть и активная позиция, и несколько вариантов решения проблемы. Если даже ни один из этих вариантов не подойдёт, то можно быть уверенным, что такой кандидат придумает ещё что-нибудь.

## 3.5 Вопросы на собеседовании: Сложная задача

Один из обязательных вопросов на моём собеседовании звучит так: «Представьте, что вы работаете над какой-то задачей и она оказалась очень трудной. Вы не знаете, как двигаться дальше и не уверены, что вообще её решите. Что вы будете делать?»

Этот вопрос не настолько прост, как кажется, так как покрывает сразу несколько областей: технический кругозор, стрессоустойчивость, умение работать в команде. Задача поставлена очень общо, конкретные действия, конечно, зависят от деталей проблемы. Поэтому самое главное, что ожидается от кандидата — это большой список возможных действий, чем больше, тем лучше:

- Погуглить возможные решения;
- Попросить помощи у коллег;
- Задать вопрос на специализированном форуме;
- Если проблема связана с какой-то коммерческой библиотекой, то обратиться в службу поддержки этой библиотеки;
- Сообщить своему менеджеру и попросить у него помощи;
- Если возможно, то вообще не делать эту задачу (изменить требования);
- Переключиться на другую задачу. Возможно, решение придёт, когда над ним не работаешь;
- Закопаться в код поглубже, нерешаемых задач нет.

Длина такого списка возможных действий определяет способность кандидата работать над такими «нерешаемыми» задачами. Эти задачи не решаются в лоб, поэтому чем больше вариантов может придумать кандидат, тем больше вероятность, что один из них сработает.

Особенно хорошо, когда в этом списке есть неочевидные решения (вроде варианта не решать эту задачу). Просто прекрасно, когда кандидат учитывает нетривиальные проблемы, связанные с этими вариантами. Например, просить помощи на форуме можно, только если проблемный код не содержит значимой интеллектуальной собственности (чтобы можно было его разместить в общем доступе).

Но кроме того интересно, есть ли у кандидата в ответе варианты попросить у кого-то помощи (у менеджера, у коллег, у экспертов в интернете). Если этого нет, то у кандидата проблемы с коммуникацией.

Отдельно интересно, упомянет ли кандидат, что о проблеме нужно сообщить менеджеру или лиду. Если кандидат работал в команде с нормальными процессами, то он проблему обязательно озвучит, чтобы вся команда знала о ней. Если нет, то вы можете получить в команду человека, который периодически «закапывается» в какую-то задачу и никто не знает, что вообще есть какая-то проблема в проекте. Такое поведение очень распространено и очень разрушительно, поэтому обратите внимание, готов ли кандидат рассказать другим, что у него есть трудности, с которыми он не может справиться.

Один раз у меня было очень показательное обсуждение с кандидатом. Вместо списка возможных решений, он возмущённо сказал, что очень странно спрашивать про сложные задачи, так как единственное, что тут можно сделать, это нырнуть в код поглубже и работать над задачей, пока она не будет решена.

Я не видел смысла продолжать обсуждение, так как вопрос вызвал у кандидата негатив, поэтому просто пошёл дальше. Но это кандидата тоже не устроило и у нас состоялся следующий диалог:

— Пойдите. А какой ответ вы ожидали на этот вопрос?

— Я не ожидаю какого-то определённого ответа. Здесь именно важно, узнать, что вы думаете. Например, многие говорят, что они первым делом попробуют наугад решение.

— Но ведь это очевидное решение! Конечно, я тоже попробую поугадить! Неужели вы думаете, что я не умею пользоваться Гуглом?

— Уверен, что умеете. Но это вопрос не на ваши навыки, а на образ ваших мыслей. Например, после вашего ответа я знаю, что вы не считаете нужным проговаривать очевидные вещи.

— Конечно! Они же очевидные!

Я думаю, что кандидат так и считал меня и мои вопросы глуповатыми. Но зато этот вопрос вскрыл очевидную проблему кандидата с общением. Он не указал, что можно посоветоваться с кем-то насчёт возникшей проблемы. Кроме того, он считает вещи, очевидные для него, очевидными для других. И он возмущённо отреагировал на абсолютно нейтральный мой вопрос и столь же нейтральное обсуждение. Не имеет даже смысла оценивать способность такого кандидата решать сложные задачи. С такими проблемами в коммуникациях в большинстве проектов до сложных задач он не дойдёт.

Также обращу внимание на важный момент: вы всегда должны понимать сами и быть способны объяснить кандидату, зачем вы задаёте тот или иной вопрос. Кандидату это объяснение может не понравиться, но вы не должны задавать вопросы вслепую, надеясь, что потом сможете сделать какой-то вывод из всех полученных ответов. Осмысленные действия интервьюера — это необходимый минимум для проведения собеседования.

### 3.6 Вопросы на собеседовании: Болевые точки в коммуникациях

У каждого из нас есть какие-то проблемы, которые нас раздражают, с которыми мы плохо справляемся, которые отравляют нам жизнь. Для кого-то это включённый кондиционер, для кого-то это разговор на повышенных тонах, для кого-то это нечёткая постановка задач. Особенно большое влияние имеют такие проблемы при общении людей в команде. Поэтому такие болевые точки кандидата должны быть выделены на интервью в обязательном порядке.

Здесь, как и во всех других ситуациях, лучше всего работает прямой и открытый подход. Я просто спрашиваю кандидатов: «С какими людьми вам тяжело общаться?» Примерно в трети случаев в ответ вы получаете рассказ о прошлом менеджере или заказчике, с которым у кандидата были проблемы. И дальше уточняющими вопросами вы проясняете, что мешало кандидату нормально общаться в такой ситуации.

Примером подобного обсуждения может служить следующий диалог из моей практики:

— С какими людьми вам тяжело общаться?

— Обычно проблем с общением у меня нет, но вот сейчас у меня заказчик, с которым мне трудно разговаривать. Из-за любой мало-мальской проблемы он начинает ругаться матом на меня. Я считаю это неприемлемым. После каждой такой стычки я хожу расстроенным.

— Вы пробовали это как-то решить?

— Да, я несколько раз говорил об этой проблеме заказчику. Объяснял, что это неконструктивно и что это меня оскорбляет. Но он не обращает внимания. Похоже, это единственный способ общения, который он понимает.

— Как-то ещё пробовали оказать воздействие на заказчика?

### 3.6. ВОПРОСЫ НА СОБЕСЕДОВАНИИ: БОЛЕВЫЕ ТОЧКИ В КОММУ

— Да, я эскалировал эту проблему своему руководителю, но он сказал, что заказчик очень важен и надо терпеть. Тогда я написал письмо заказчику, поставив в копию своего руководителя, где прямо написал, что если он продолжит общаться в подобном тоне, то я буду просто вешать трубку. Всё равно через маты никакой вопрос решить нельзя. Теперь так и делаю. В принципе, это помогает. По крайней мере, я теперь не хожу, как оплёванный.

— То есть проблема решена?

— Не совсем. Мне не нравится, что приходится применять такие жёсткие методы. Я предпочитаю общаться в атмосфере взаимного уважения. Собственно, это одна из причин, по которой я меняю работу. Смена работы должна стать решением проблемы.

Достаточно прямой, подробный и понятный ответ. Само решение, придуманное кандидатом, меня не особо волновало. В моих проектах таких агрессивных заказчиков нет, а если и появятся, то привести их в чувство можно будет легко. Я обращал внимание только на то, что кандидат активно пытался решить проблему, пытался договориться, обращался за помощью. Кандидат сделал со своей стороны достаточно, чтобы справиться с проблеммой. И пришёл к некоторому решению. Так что такой кандидат мне подходит. Критичных для меня болевых точек он не имеет.

Но гораздо чаще вместо такого разговора случается совсем другой:

— С какими людьми вам тяжело общаться?

— С теми, кто тебя не слушает. Вообще с такими людьми непонятно, как разговаривать. Вот только с утра была такая ситуация. Пытаюсь объяснить человеку, что ему надо бумажку подписать, а он не слушает! В ответ бурчит что-то непонятное. Я раз ему объяснил, второй, третий. А он не слушает! Всё своё что-то талдычит и талдычит!

— А что талдычит-то?

— Да кто его знает? Я не слушал даже. Ерунду какую-то. Да и не важно это! Важно, что не слушает совсем меня!

Видите парадокс? Кандидат не слушал своего собеседника, но осуждал, что его самого не слушали.

Человек часто считает очень плохими именно те проблемы, которые имеет он сам. Это распространённый психологический паттерн поведения. Вы слушали когда-нибудь участников «кондиционерных войн»? Они ведут себя одинаково и высказывают абсолютно одинаковые доводы. Те, которым жарко, говорят: «У меня есть право включить кондиционер, если мне жарко! Кому-то холодно, но почему я должен страдать? От жары я болею. А им всё равно! Они даже прячут пульт от кондиционера. Это низко! А когда я прячу пульт, они приносят откуда-то другой и выключают мой кондиционер!»

Вы можете заменить пару слов и получить позицию противоположной стороны. Потому что реально эти позиции одинаковы. И такая же ситуация с большинством других конфликтов. Возможно, у вас бывало так, что вы заходили в комнату и видели разгар какой-то перепалки. И часто в такой ситуации, если вы не видели начало конфликта, то вам невозможно определить зачинщика, определить кто прав и кто виноват.

Потому что так работают реальные проблемы в коммуникациях. Если человек не умеет работать с чужой агрессией, то он часто отвечает такой же агрессией. Человеку кажется, что его реакция обоснованна, ведь его притесняют. Но на самом деле его реакция так же разрушительна, как его притеснение.

А часто чужая «агрессия» надумана. Как в вышеприведённом диалоге, где кандидат думал, что его не слушают. Неизвестно, прислушивались к нему или нет. Из-за своих проблем он настолько был настроен против других, что не мог понять, что происходит в реальности. Его собеседник мог очень внимательно его слушать, но это не имело значения.

### 3.6. ВОПРОСЫ НА СОБЕСЕДОВАНИИ: БОЛЕВЫЕ ТОЧКИ В КОММУ

Аналогичное поведение встречается очень часто. Люди обвиняют других в отсутствии чувства юмора, потому что те не смеются над их шутками (не смешными). Или осуждают коллег за то, что они очень громко разговаривают (даже если орать на них по этому поводу десять раз на день).

Когда кандидат отвечает на любой вопрос отслеживайте такие «выпады» в сторону других и пытайтесь немного исследовать каждый такой случай. Во многих ситуациях вы выясните какие-то особенности характера кандидата, с которыми вам придётся как-то жить в случае найма.

Но и не стоит сразу любую жалобу обращать на самого человека, так как иногда действительно может попасться кандидат, пытающийся сбежать от ужасного коллектива и неадекватного руководства.

## Про ненадёжного разработчика

Как-то в мой проект предложили перевести одного очень ненадёжного разработчика, Петю. Вроде, несколько лет назад он работал хорошо. И даже карьеру некоторую начал, получив пару повышений. А потом потерял мотивацию, начал деградировать, обиделся на весь мир и никак не мог найти себе место, где он справлялся бы с работой. Так как когда-то он работал нормально, то просто увольнять Петю не хотелось, вот и пытались HR найти ему место, где он мог бы снова найти себя.

Мой тимлид Петю давно знал и видеть его в проекте не хотел.

— Ну давай его хотя бы прособеседуем. Узнаем, что он из себя представляет, — предлагал я.

— Да фигня эти твои собеседования! — говорил тимлид, — Он будет говорить одно, а в работе будет вести себя совсем по-другому.

— Да не бывает такого. Люди не считают своё поведение чем-то плохим, поэтому и говорят, что думают, — настаивал я.

— Не, нифига. Все себя с хорошей стороны пытаются показать, а не правду сказать.

— Ну вот чего самого плохого ты боишься?

— Да я боюсь, что он при появлении сложной проблемы просто встанет и уйдёт. А мне за ним разгрести!

— Ну вот давай это и выясним.

На том и порешили. Назначили интервью с Петей и стали беседовать. Мне, конечно, хотелось понять, что из себя представляет Петя, но ещё больше мне хотелось передать своему лиду свой подход к собеседованиям и показать, что можно понять, как себя будет вести человек в реальной ситуации.

У меня была подсказка, мой лид знал Петю и знал, что нам надо выяснить. Поэтому я задал вопрос, который потом стал частью моего стандартного опросника: «Петя, представь, что ты работаешь над какой-то задачей и она оказалась очень трудной. Ты работаешь-работаешь, но никакого результата не видно. Как её решить, ты не знаешь. Что ты будешь делать?»

Пете вопрос не понравился. Он скривился и ответил: «Ну что тут сделаешь? Ещё поработаю». Меня такой ответ не устроил, так как это не ответ на поставленный вопрос, а некоторая отговорка. Поэтому я продолжал настаивать: «Так ты же уже поработал. И результата нет. Может быть, если продолжать так же, как раньше, то результата и не будет. Как решить задачу, если она не решается? Возможно раньше были такие ситуации и ты как-то с ними справлялся?»

Петя погрузнел ещё больше. Явно идея работы над задачей, которую он не может решить, его не вдохновляла. Но он ответил: «Я для себя понял, что не имеет никакого смысла продолжать работу, если она не приносит результата. Я трачу силы

— и всё впустую. В таком случае лучше пойти домой и отдохнуть. Результат такой же, но хотя бы не измотаешься».

Таким образом я получил для себя доказательство того, что опасения моего лида оправданы. А ему показал, что Петя не особо скрывает, что он готов бросить работу, если встречается сложность. Для Пети всё логично. Если ты работаешь и не получаешь результата, то зачем работать? Можно бросить работу.

С моей точки зрения и с точки зрения моего лида такое поведение очень плохое. Но не для Пети. Поэтому он не видит смысла скрывать своё отношение. Да, он знал, что «надо упорно работать», «стремиться находить нестандартные решения» и т.д. Но это для него было пустыми словами, поэтому он не мог давать развёрнутые ответы, исходя из этой «идеальной» установки, в которую он не верил. А свои реальные убеждения для него естественны и обоснованны, поэтому он их и высказывает.

## 3.7 Никто не считает себя плохим

Есть один момент, который очень важно понимать. Ни один кандидат не считает свой образ мыслей и действий неправильным. Это очень помогает при собеседовании, так как кандидат готов давать честные ответы.

Например, я как-то собеседовал менеджера, который считал, что он может уволить любого члена команды просто за то, что тот не нравится. Он довольно детально объяснил свою точку зрения: «Да, можно тратить время, пытаюсь понять, что именно не нравится, и потом выискивая поводы для увольнения. Но зачем это всё? Мне с ним не нравится работать, я начальник — можно увольнять».

Для меня такие рассуждения были дикими. То есть я даже не могу сказать, что эти рассуждения неправильные. Вряд ли кто-то из команды, непонравившийся тому менеджеру, смог бы дальше с ним работать. Но сам подход для меня очень странный. Но не для этого кандидата!

Он так работал долгое время. На его месте работы этот подход всех устраивал. Его, возможно, даже хвалили за решительность. Почему он должен был мне отвечать не так, как он думает? Как он может предугадать, что я отношусь плохо к такому подходу? Никак. Поэтому кандидат просто будет отвечать то, что думает. И это прекрасно.

В истории «Про ненадёжного разработчика» на стр.85 я уже показывал кандидата, который прямо говорит, что если у него что-то не будет получаться, то он просто пойдёт домой. Поэтому что он считает это нормальным. У меня как-то был диалог на подобную тему с кандидатом:

— Предположим, вы задержались на работе, исправляя мелкий но очень важный баг, — описывал я ситуацию — Все разошлись, вы доделали свою работу, но заметили другую очень серьёзную проблему, с которой нельзя делать релиз. С утра нужно отправлять релиз заказчику, а приложение сломано, и ошибка не в вашем коде. Что вы будете делать?

— Подойду к менеджеру и сообщу о проблеме, — отвечает кандидат.

— Но менеджер ушёл домой. Вы одни в офисе.

— Тогда я тоже уйду домой. Зачем мне что-то делать?

Для меня это ответ неприемлемый, а для кандидата естественный. И тут худшее, что интервьюер может сделать, это показать своё негативное отношение к словам кандидата. Потому что люди в IT работают умные, опытные и профессиональные. Они очень быстро поймут, какие ответы вы ждёте, и будут вам

их давать. С тем же успехом вы можете вести собеседование собственного отражения.

Причём скрыть своё неодобрение вам будет очень сложно. У людей богатые невербальные средства выражения эмоций, и кандидат поймёт, что сказал что-то не то по вашему молчанию, приподнятой брови или нервному постукиванию пальцами по столу. Поэтому вам не просто нужно не показывать вашего неодобрения, вам надо всей душой поддерживать кандидата, проникнуться его отношением и относиться к его ответам нейтрально, а лучше позитивно.

В конце концов он делает вашу работу гораздо проще, открыто выражая своё мнение. Не стоит платить за это презрением.

## 3.8 Социальное одобрение

Необходимо учитывать, что есть так называемые социально одобряемые ответы. Они охватывают удивительно большой круг вопросов. И все кандидаты знают «правильные» ответы. Например, если вы спросите кандидата, нравится ли ему заниматься саморазвитием, то он ответит «Да» без размышлений.

А чего тут думать? Конечно! Хотя если вы подумаете, то вспомните многих коллег, которые не тратят свои ресурсы, деньги и время на дополнительные курсы, книги и тренинги. То есть в саморазвитие они не вкладываются. Ведь это такое напряжение! Когда же тогда отдыхать? Но признать, что развитие им в текущий момент времени неинтересно, вряд ли кто-то захочет.

Здесь нет никакого лукавства. Я сам помню, как попался на удочку социально одобряемых ответов. Меня несколько раз спрашивали, хочу ли я ездить в зарубежные командировки, и

я отвечал «Да!» Но когда возникали реальные возможности ездить в такие командировки, я в них не особо рвался. Потому что момент мне казался неподходящим, потому что командировки казались слишком длительными, потому что работы было много и удалённо было бы тяжело всё успеть.

Только через некоторое время моя жена мне подсказала, что командировки я не люблю. Я задумался и понял, что да, мне нравится занимать позицию, где нужны иностранные командировки, мне нравится работать в международных компаниях, а там такие командировки часто необходимы, мне просто нравятся разные страны и культуры. Но командировки для меня являются скорее минусом, чем плюсом. Это важная, интересная, но тяжёлая часть моей работы.

Поэтому, задавая вопросы, очень тщательно продумывайте, не испортит ли ваш вопрос готовая установка, внедрённая в мозг кандидата обществом. Например, если вы хотите узнать, насколько кандидата мотивируют деньги, то не стоит спрашивать: «Хотите ли вы зарабатывать больше?» Ответ очевиден. Намного результативней спросить: «Согласны ли вы работать на 15% больше, чтоб получать зарплату на 15% выше?»

Спрашивать, нравится ли кандидату изучать английский язык, тоже смысла нет. Лучше спросить, какие методы изучения он считает оптимальным для себя, а потом спросить, когда он ими пользовался в последний раз и чего добился.

## Про тренировку безоценочности

Автомобилисты очень любят оценивать других участников движения. Причём дают оценки не только с позиции «хорошо»/«плохо», а используя гораздо более развёрнутые эпитеты, учитывающие уровень умственных способностей, способ получения водительских прав, происхождение или сексуальные наклонности других водителей и пешеходов.

Потому что кажется ведь очевидным, как надо вести себя на дороге правильно (так, как я). Да и какие вопросы о неправоте вообще могут быть, если в городе вы едете 60км/ч, а вас обгоняет дорогой спорткар, идущий явно больше сотни? Или если вы на жаре стоите в пробке, а рядом по обочине нагло пронесится большой внедорожник, поднимая вокруг пыль? Или если поперёк вам на перекрёстке выезжает другой автомобиль и создаёт затор, мешая вам проехать? В такие моменты становится очень обидно и испытываешь уверенность, что эти нехорошие водители не знают правил дорожного движения, наплевательски относятся к другим и должны быть помещены в специализированное лечебное учреждение.

Я, как и все, нервничал и вешал обидные ярлыки на таких водителей. А потом на одном автомобильном форуме прочитал короткую историю. Форумчанин писал: «Вы знаете, мне как-то на работу позвонили и сказали, что мой отец с сердечным приступом попал в реанимацию, что он в очень тяжёлом состоянии и может умереть в любой момент. После этого я мало что помню, но в больнице я был через 15 минут. Хотя обычно дорога занимает там около получаса. Наверное, превышал. Наверное, подрезал и проезжал на красный. Но очень благодарен всем, кто меня в тот день пропустил, хоть я их и не помню».

И эта история произвела на меня впечатление. Оказывается у человека может быть весьма весомое объяснение его неправильных действий. Можно поругать его за то, что он сел за руль в таком состоянии. Но абсолютно точно он нарушал правила не потому что «дебил» или «права купил». Если кто-то думал так о нём, видя на дороге, то он был не прав.

И мне показалось очень несправедливым вешать обидные ярлыки на людей, которые их не заслужили. Да, возможно, 99% таких нарушителей делают это из хулиганских побуждений. Но может быть, кто-то один из тех, кого я ругал, имел какой-то понятный мотив? Даже если в 99% случаев я был прав, то остав-

шийся 1% — это слишком много. Я решил, что я не должен осуждать других несправедливо.

И я стал по-другому смотреть на «дурных водителей», окружающих меня. В некоторых ситуациях я видел у них встревоженный взгляд и нервные жесты. Может быть они так же едут в реанимацию к кому-то и не понимают, что они делают? И когда такая мысль приходила в мою голову, мне не хотелось ругать этих водителей.

Потом я стал придумывать объяснения тем действиям людей, которые я видел. Кто-то прёт по обочине? Надо же, насколько человек хочет помочь другим. Двигается по обочине, чтобы не увеличивать пробку. Наверное, на перекрёстке тоже по обочине повернёт направо. А нет, встроился обратно в пробку. Наверное, ему позвонили и пришлось изменить планы.

Не включил поворотник? Видимо, повреждение электрических схем. Порше едет 120км/ч по встречке? Видимо это новая машина ГИБДД выполняет преследование преступника, а звуковая сирена и проблесковый маячок повреждены меткой стрельбой вражеских автоматчиков. Проехал на красный? Тормоза отказали.

Иногда объяснения были очень уж неправдоподобными, иногда весьма вероятными. Через некоторое время я стал понимать, что поступки, которые я считал просто чудовищными, скорее всего являются следствием небольших и объяснимых ошибок.

Я увидел, что многие развязки действительно трудно проехать, если ты не местный. Часто навигаторы дают неправильные указания. Часто бывают какие-то третьи участники движения, которые закрывают знаки, не пускают в полосу или ещё как-то вынуждают других нарушать правила.

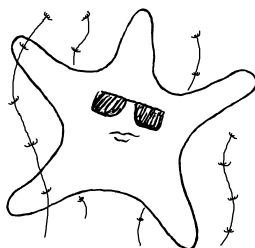
Но правдивость придуманных мной объяснений неважна. Важно то, что я стал с юмором относиться к тем же ситуациям, которые меня раньше раздражали. Например, машина стоит в

левой полосе, из которой движение разрешено только налево. При этом у неё включён правый поворотник. Куда она поедет? Мы с моими детьми делаем ставки и потом все радуемся вместе с победителем.

Я стал понимать, что даже если кто-то неправ, то меня напрямую это не касается. Не стоит расплывать своё внимание и тратить время на выражение недовольства. А если опасность угрожает мне, то опять же не надо ругаться, а надо маневрировать или тормозить, чтобы этой опасности избежать.

Я стал понимать, что мои действия со стороны могут выглядеть так же неадекватно, и меня называют очень неприятными словами, просто потому, что другие не видят, что мне под ноги с тротуара чуть не кинулся ребёнок, или что с моей стороны не работает светофор, или что велосипедист чуть не упал мне на капот из-за неровности дороги.

В результате я стал меньше судить других, и это сделало мою жизнь спокойней. И это касается не только ситуаций на дорогах.





## Глава 4

# Выведение

Есть ещё несколько интересных моментов, которые мы не обсудили ранее. Я их вынес отдельно в эту главу.

### 4.1 Офер

Итак, вы решили кандидата брать. Часто на этом этапе все расслабляются, отправляют формальное письмо с указанием зарплаты кандидату и считают свою миссию законченной. Иногда это действительно работает. Если у кандидата ожидания просты, и если ваши нужды тоже не отличаются оригинальностью, и если вы обговорили все детали работы, то кроме этого письма ничего и не надо. Кандидат выйдет на работу, будет работать с 9:00 до 18:00 с перерывом на обед, а вы ему будете платить зарплату.

Но часто бывает, что найм кандидата не такой прямолинейный. Например, вы нашли себе лида или архитектора, который должен взять новое направление, или должен спасти очень критичный проект, или занимать не совсем стандартную роль. Возможно, все собеседования отняли много сил и времени с обеих сторон, и вы хотите повисить свои шансы на то, что канди-

дат примет ваше предложение. В таких ситуациях стоит offer делать лично или хотя бы по телефону, дополнив моментами, о которых мы поговорим чуть ниже.

Конечно, формальный offer, высланный на почту кандидату или врученный лично всё равно нужен. Это документ, фиксирующий ваши договорённости.

Но кроме фиксации договорённостей обязательно нужно поздравить кандидата с успешным прохождением всех этапов собеседования. Любой offer — это победа кандидата, признание его высокой квалификации и радостное событие для него. Вам также выгодно, чтобы кандидат считал offer от вашей компании радостным событием. Так и сделайте это событие радостным! Можете к offer приложить фотографию праздничного салюта, лишним не будет.

Важно сформулировать, почему вы приняли решение нанять именно этого кандидата. Совсем не нужно давать детальный отзыв о всех плюсах и минусах кандидата, это сильно повредит делу (см. главу 4.3 «Как давать обратную связь кандидатам» на стр.102). Но вы должны показать кандидату, что его найм — не случайное решение, что вы знаете, что делаете.

Важно, чтобы желания кандидата, высказанные на собеседованиях, соответствовали вашим планам на него. Если кандидат хочет сменить работу из-за очень скучного проекта, где он делал одни и те же операции, то не надо объявлять ему, что вы его нанимаете из-за способности к рутинной работе. Вы не должны нанимать человека в такой же скучный проект, так он разочаруется и уйдёт от вас через очень короткое время. Такой кандидат может захотеть сменить работу, только если вы сможете ему предоставить динамичный и интересный проект. Очень важно показать кандидату, что вы это поняли, что вы учитываете его желания и что вы можете предложить ему то, что нужно.

Особенно важно делать отдельную встречу для офера, если у вас есть сомнения по поводу кандидата. Не сомнения в его пригодности (такого кандидата нанимать нельзя), а небольшие детали, некоторые несоответствия профиля кандидата вашим нуждам. Вы уверены, что кандидат справится со всеми сложностями, поэтому и делаете offer. Но мало верить, надо договориться с кандидатом, что он будет прикладывать дополнительные усилия в нужном вам направлении.

Например, кандидат недостаточно знает английский язык, а ему нужно интенсивно общаться с американским заказчиком. Вы уверены, что кандидат сможет за месяц подтянуть свои знания. Но хочет ли кандидат напрягаться? Возможно, кандидат выражал готовность подтянуть английский, но вряд ли на собеседовании вы договорились с ним о конкретных сроках и о конкретном уровне, которого он должен достигнуть.

Или вам кажется, что кандидат не имеет склонности к интенсивной работе, что он хочет найти «тихое место», где он будет выполнять свои обязанности в постоянном темпе. И если ваш проект совсем не тихий, то правильным будет сказать об этом кандидату. Только делать это нужно экстремально корректно, избегая оценок «хорошо»/«плохо». Необходимо очень конкретно описать кандидату свои ожидания.

Например: «Проект, в котором вы будете работать, очень динамично развивается. В течение ближайшего полугодия работы будет много, приоритеты будут постоянно меняться и надо будет переключаться между задачами. Возможны переработки в количестве 10 часов в месяц, которые будут оплачены по двойной ставке. Через полгода переработки прекратятся».

Кандидат должен очень чётко понимать, на что именно он соглашается, если принимает ваш offer. И здесь вы можете прояснить важные детали. Например, кандидат может согласиться на переработки, только если ему будет позволено овертаймы

вырабатывать удалённо, из дома. Для вас такое условие может быть приемлемым, а может не быть.

Если вы договариваетесь о таких деталях, то делается офер с уверенностью, что он будет принят, так как все условия уже согласованы с кандидатом. При таком обсуждении офер фактически является ещё одним этапом собеседования.

Стоит заметить, что на этом этапе есть все условия, чтоб договориться. Вы уже проверили кандидата, цените его навыки и готовы принять его на работу. Кандидат видит эту вашу готовность, видит, что вы серьёзно подошли к его найму, и эмоционально готов сам вложиться в ваше сотрудничество.

Обязательно спросите у кандидата, когда он будет готов дать ответ, и включите в офер дату, до которой он действует. Офер является очень важным документом и для разработчика (так как он основываясь на нём уходит с текущего места работы), и для компании (так как это обязательства, которые надо выполнять). Бывает, что получив офер, будущий работник берёт время на подумать и исчезает на неделю или две без определённого ответа. Для вас это очень уязвимая ситуация, так как вы не можете принять другого кандидата и не можете быть уверены в текущем.

Нельзя требовать от человека немедленного ответа. Многим нужно время подумать или обсудить ситуацию с семьёй, но бизнес не стоит на месте и за пару дней кандидату надо дать ответ. Обязательно договоритесь об этом с кандидатом.

## **4.2 Разработчик, который не хочет менять работу**

Бывают случаи, когда кандидат на простой вопрос «Почему вы хотите сменить работу?» Даёт неожиданный ответ: «Я не хочу менять работу. Меня всё устраивает. Я просто интересуюсь

происходящим на рынке и хочу понять, что представляет собой ваша компания».

Не нужно обижаться на кандидата за то, что он тратит ваше время. Наоборот, стоит поблагодарить его за столь прямой ответ и доверие, которое он вам оказывает своей открытостью.

А потом нужно оценить свою заинтересованность в таком кандидате. На моей практике подобные кандидаты оказывались очень интересными. Если кандидат не прикрывается какой-то отговоркой, вроде «хочу роста по зарплате» или «хочу новых интересных проектов», и если он делает это корректно и вежливо, то это свидетельствует о зрелости.

Причём подобный ход очень сильный с точки зрения переговоров (а собеседование — это всегда переговоры). Если кандидат использовал бы отговорку, то вы могли бы уточнить у него, какую зарплату он хочет или что не устраивает его в текущих проектах, и использовать эту отговорку как рычаг в переговорах. А что же делать, если кандидат никакой заинтересованности не выказывает?

Обычная реакция на сильного кандидата, которого хочется нанять, но который наниматься не хочет, это сделать ему выгодный оффер. Кандидат хороший, значит вы можете предложить ему хорошие деньги, которыми, возможно, он заинтересуется. Это ошибка. Чтобы замотивировать опытного лояльного специалиста на смену компании только деньгами, вам придётся сделать неразумное предложение. На практике я видел много случаев, когда удвоения текущей зарплаты было недостаточно для смены работы. А даже если кандидат согласится, то его мотивация работать на вас будет очень слабой. Проблемы в будущем будут такие, что вы станете удивляться, зачем вообще вам нужен был этот кандидат. Вам надо сперва создать мотивацию.

Заметим, что какой-то интерес у кандидата есть, иначе он не стал бы тратить на вас своё время. Но этот интерес недо-

статочен для офера. Пытаться сразу нанять такого кандидата неразумно. Нужно играть в открытую и сказать об этом: «Мы не делаем оферов людям, которые не хотят у нас работать, это не имеет смысла. Но мы очень уважаем вашу искренность и интерес, проявленный к нашей компании. Я буду рад дать вам всю информацию, которую могу. Давайте продолжим интервью, чтобы я понял ваш уровень, и посмотрим, как мы можем двигаться дальше». Это выровняет ваши позиции в переговорах, так как офер — это, скорее всего, то, на что рассчитывал кандидат. Офер — это признание его профессионализма, а также информация о зарплатах в вашей компании. Но раз он не хочет у вас работать, то вы не хотите делать ему офер. Всё честно, понятно и нет никаких уловок ни с чьей стороны.

А вот дальше надо действовать по схеме трудных переговоров, когда обе стороны не очень настроены работать друг с другом. Нужно двигаться мелкими шажками, обмениваясь друг с другом тем, что имеет ценность: информацией и своим временем. Кандидат пришёл к вам и тратит своё время, и вы тоже тратите своё время на него. Он открыт и вы открыты. Вы на равных. Дальше вы задаёте свои вопросы, получая информацию о кандидате. Кандидат получает возможность задать свои вопросы.

Кроме ответов на вопросы кандидата обязательно приложите дополнительные усилия со своей стороны. Расскажите про компанию, про себя, про проект, про ваши цели и ваши надежды. Но не делайте слишком много и не требуйте слишком многого от кандидата. Ваша задача создать небольшую положительную разницу по усилиям, чтобы кандидат чувствовал, что ему тоже надо что-то сделать для вас. На первых порах достаточен обмен информацией. Когда вопросы с обеих сторон кончатся, поблагодарите кандидата и расстаньтесь с надеждой на продолжение, но без взаимных обязательств.

Дальше вам надо серьёзно подумать, что ещё вам нужно от кандидата, чтобы удостовериться, что он ценен для вас. Это реальная работа, это усилия с вашей стороны, о которых вам надо будет сказать кандидату: «Я много думал, как мы можем работать дальше и понял, что мы знаем всё-таки недостаточно друг о друге. Вот моя команда приготовила сложное техническое задание, которое в точности соответствует тем заданиям, которые будут на проекте. Буду рад, если вы найдёте время и сможете в течение пары недель его выполнить». При этом от вас требуются усилия действительно подготовить это задание, возможно, вовлекая всю команду и стараясь, чтобы это задание было нескучным, реально соответствующим проекту, но не похожим на кусок работы, который кандидат сделает для вас бесплатно.

Если кандидат выполнит это задание, то нужно созвониться или встретиться с ним, обсудить решение, рассказать, как это было решено в реальном проекте, снова поблагодарить его. И напомнить, что это задание даёт ему много информации о реальном проекте, что вы его сделали специально для кандидата, и что это больше, чем делается для обычного кандидата. Но здесь всё опять-таки честно: вы прикладываете усилия, и кандидат прикладывает усилия тоже.

Потом вы можете пригласить кандидата для знакомства с командой, или договориться с заказчиком, что он посмотрит резюме кандидата (а для этого надо резюме обновить и перевести на английский), или пройти дополнительное тестирование. Суть в том, что это должны быть не «пустышки», а какие-то реальные действия, которые для вас добавляют информации о кандидате. Ну и конечно, каждое очередное усилие со стороны кандидата будет свидетельствовать о том, что он таки заинтересован.

И вот такими шажками вы довольно скоро вы придёте к ситуации, когда вы точно знаете, что кандидат сможет хорошо работать в вашей компании, что он обладает всеми нужными на-

выками, и что его заинтересованность достаточна. Естественно, где-то в этом процессе вам (и кандидату) надо будет убедиться, что ваша компания может предложить достойную зарплату. Эта зарплата не обязательно даже больше, чем на предыдущем месте работы. Я сам не раз менял работу с потерей в зарплате ради интересных задач и будущих перспектив.

И вот тогда вы можете сесть с кандидатом, обсудить, что вы с ним проделали кучу работы, что вы стали практически родными друг другу, и что вот предложение, которое вы ему можете сделать. Причём предложение включает не только какую-то сумму денег, а очень конкретное содержание (с которым кандидат уже знаком, так как он делал ваше задание) и работу с конкретными людьми (заказчиком, командой, с которыми кандидат познакомился). Согласие кандидата на этом этапе — это больше формальность, так как в процессе вашего взаимодействия вы проработаете все детали, важные для него.

### 4.3 Как давать обратную связь кандидатам

Очень важный момент, это то, как именно нужно сообщать кандидатам о результатах интервью. Если всё хорошо и кандидат получил оффер или просто прошёл на следующий этап, то вопросов обычно нет. А вот если человека отсеяли по какой-то причине, то мало кто корректно может ему об этом сказать.

Некоторые компании считают необходимым выдать кандидату детальный отчёт с указанием, какие именно минусы его кандидатуры были выявлены. Но на самом деле, единственная ситуация, когда вы должны рассказать кандидату, что вы о нём думаете — это когда он прямо об этом вас попросил. В других ситуациях это делать категорически не нужно.

Конечно, нельзя держать людей в неведении. Если кандидат не прошёл на позицию в вашей компании, то ему об этом нужно сообщить. Но расписывать, что именно в нём вас не устраивает, абсолютно лишне.

Отказ сам по себе не является приятной новостью, а приправленный каким-то негативом он может сильно расстроить и даже озлобить кандидата, что помешает дальнейшей работе с ним. Причём надо понимать, что то, что является минусом для вашей компании, может являться большим плюсом для другого места работы. Например, вам не понравилась излишняя жёсткость кандидата и отсутствие в нём стремления быть частью команды. Но в текущей своей работе, возможно, кандидату как раз и нужно принимать жёсткие решения, а командной работы нет. Возможно, кандидат приложил много сил, чтобы выработать в себе эти качества и не собирается от них отказываться в настоящее время. Ваш незванный комментарий на этот счёт может убедить кандидата, что у вас просто не хватает профессионализма. И это убеждение он будет поддерживать и сообщать всем своим знакомым.

Периодически ко мне обращаются кандидаты, которые обиделись на письмо, присланное им по результатам интервью. Очень трудно убедить их глядеть на вещи с позитивной точки зрения и не воспринимать критику как что-то плохое. Позитивное отношение к критике — это очень полезный навык, но владеют им далеко не все.

Совсем другое дело, когда кандидат просит вас дать ему обратную связь. Это очень положительно характеризует самого кандидата. На моей практике меньше 2% всех кандидатов просят рассказать им о выявленных плюсах и минусах. Максимум, кандидатов интересует, прошли они или нет. Желание получить знания для своего развития — это очень хороший знак.

Выдача такой обратной связи — это очень ответственное действие. Если у вас не готов хорошо сформулированный ответ

для кандидата, лучше взять небольшой таймаут и пообещать выслать ему отзыв чуть позже.

Отзыв лучше всего делать по правилу бутерброда. Любые плохие новости нужно поместить внутри хороших. Наверняка вы видите какие-то плюсы, начните отзыв с них. Например, само желание кандидата получить обратную связь является таким плюсом. Скажите, как это здорово и насколько это редко. Кандидату будет приятно. Закончить отзыв надо тоже на позитивной волне. Если человек хочет развиваться, значит в будущем у него будет всё хорошо. Вот и скажите ему об этом.

Любые сильные стороны стоит отметить, даже если вам они кажутся очевидными. У кандидата много лет интереснейшего опыта? Озвучьте это. Вам нравится, как кандидат себя ведёт на интервью? Скажите ему, что это важно.

Негатив надо давать очень осторожно. Лучше говорить не о том, что плохо, а о том, что нужно изменить. Разница кажется незначительной, но она есть и она важна. Нужно не тыкать кандидата носом в его недостатки, а давать ему направление для дальнейшего развития. Например, недостаточно сказать: «Ваша алгоритмическая подготовка крайне низкая». Кандидат наверняка не знает, что делать с этой информацией. Гораздо лучше сказать: «Вам нужно подтянуть знания основных алгоритмов сортировки и поиска на графах, и порешать логические задачки. Это вам даст основы для понимания более сложных тем, таких как сложность алгоритмов».

Но важно не пересластить пилюлю и если какие-то качества кандидата поставили крест на его кандидатуре, то надо дать ему это понять. Иначе кандидат будет сбит с толку. Ему не назвали никаких критичных минусов, даже похвалили, а вакансия в результате ушла кому-то другому. Если вам что-то не понравилось в кандидате, то скажите ему об этом.

Не надо вываливать все минусы, которые вы видите. Этот список может быть большим. Выделите один-два самых кри-

тичных пункта, которые, на ваш взгляд, мешают кандидату. Остальное просто проигнорируйте. Кандидату достаточно иметь какой-то один пункт для работы над собой. Не стоит давать ему понять, что ему надо себя переделать полностью, чтобы работать у вас. Если минусов у кандидата слишком много, это значит, что он более пригоден для какой-то другой работы в какой-то другой компании.

## 4.4 Основное экономическое соотношение найма

Бизнес управляется экономическими соображениями. Топ-менеджеры и владельцы компании пропускают мимо ушей любую информацию, которая не выражается в деньгах. Даже если вы не имеете дела с финансовыми расчётами, то полностью игнорировать эту реальность у вас не получится.

Не будем вдаваться в детали, а ограничимся самой основной формулой, которая влияет на весь процесс найма. Итак: цена найма намного меньше цены ненайма, а цена ненайма в свою очередь много меньше цены неверного найма,  $C_{\text{найма}} \ll C_{\text{ненайма}} \ll C_{\text{неверного найма}}$ . Термин «намного меньше» определить сложно, но когда я считал это для своих проектов, то разница была примерно на порядок. То есть каждая цена из названных была в 10 раз больше предыдущей.

Давайте пройдем по составляющим этого соотношения.

**Цена найма** — это то, сколько тратит компания на приём на работу одного сотрудника. В эту графу включается зарплата рекрутеров, стоимость времени интервьюеров, стоимость рекламы и мероприятий по привлечению кандидатов и даже стоимость аренды конференц-залов для собеседований. Эти деньги очень легко увидеть и они кажутся очень значительными, но это

ровно до тех пор, пока вы не смотрите на следующую составляющую, цену найма.

**Цена найма** — это деньги, которые компания теряет пока сотрудник не нанят. Например, заказчик согласился на найм ещё одного разработчика на свой проект. С момента этого согласия и до момента выхода разработчика компания недополучает деньги. Это и есть цена найма. Или вы видите риск срыва сроков по проекту и решаете нанять ещё одного сотрудника, чтобы этого избежать. Тогда цена найма — это стоимость этого риска, то есть то, сколько компания потеряет, если сроки будут сорваны, умноженное на вероятность срыва сроков.

Да, иногда трудно понять, где именно компания теряет деньги. Например, вы видите, что команде трудно и вы набираете человека просто, чтобы избежать переработок в будущем. Но ведь это опять снижение рисков! Подумайте, что будет, если начнутся авралы и пара ключевых людей решит покинуть команду. Никому мало не покажется. Компания рискует потерять большие деньги и цена этих рисков и будет ценой найма.

Мы всегда нанимаем сотрудника, чтобы заработать больше денег. В условиях рыночной экономики (а это именно те условия, в которых мы работаем), никаких других причин быть не может, и все другие причины выражаются через деньги. Если вы не наймете сотрудника, компания потеряет деньги. Причём эти деньги намного больше, чем цена найма. В IT работают высококлассные, очень дорогие специалисты, которые приносят очень большие деньги.

Иногда бывает, что вы нанимаете кого-то, но последний кандидат от рекрутеров приходил только пару дней назад. Вы наводите справки, но не получаете ответа, когда будет следующий кандидат. Значит, следует эту ситуацию эскалировать (вашему руководителю, например). Если вы расскажите руководству, что бизнес теряет деньги, то реакция будет быстрой и эффективной. Компания выделит бюджет на рекламу, наймёт

больше рекрутеров или сделает что-то ещё. На моей практике компания решила однажды нанять 500 высококлассных специалистов «про запас». Так получалось дешевле, чем терять деньги из-за ненайма в будущем.

Когда руководство компании считает цену ненайма, иногда следует паническая реакция, и выдаётся установка нанимать любого, кто хоть сколько-нибудь подходит, невзирая на все явные или скрытые проблемы, которые есть с кандидатами. И тогда единственная ваша защита — это следующая составляющая соотношения.

**Цена неверного найма** – это то, сколько компания потеряет, если вы наймёте неправильного человека. Здесь есть много вещей, которые трудно оценить. Но кое-что известно. Например, если вам придётся уволить сотрудника, вы снова заплатите цену найма (так как вам нужно будет нанять другого). И вам придётся заплатить стоимость ненайма за всё время, которое он отработал. Работу, которую сделал неправильно нанятый сотрудник, скорее всего, придётся выкинуть. И если вы потратили 2 месяца, чтобы понять, что вы наняли не того, то всё это время вы потеряли. И это время вы не искали замену. Так что вы потеряли много времени и потеряете ещё больше, чтобы найти другого.

Но это даже не самое страшное. Стоимость рисков, которые вы привносите неправильным сотрудником гораздо выше, чем эти потери. Если вы уволите сотрудника сразу после его приёма на работу, то что будет с репутацией вашей фирмы? А сколько потеряет ваша компания, если заказчик поймёт, что вы не умеете нанимать?

Я был свидетелем, как свеженанятый разработчик в грубых выражениях высказал заказчику, что он думает про него, про проект и про компанию, на которую он работал. Он сразу уволился, но ущерб репутации был серьёзный. Можно было потерять не только текущий проект, но и все будущие проекты от этого заказчика, и очень большая команда работала, чтобы

этого не произошло. Один промах в найме и проект сразу стал очень рискованным.

Цена неверного найма очень важна, потому что эта сумма определяет вашу ценность как интервьюера. Если вы можете избежать найма неправильных людей, вы спасаете вашу компанию от гигантских трат. И если вас пытаются уговорить взять кого-то, не совсем подходящего, обрисуйте, какие проблемы ждут компанию на этом пути.

## Про неожиданные встречи со старыми знакомыми

Однажды я перевёлся в новое подразделение своей компании. Я упаковал свои вещи и переехал в другое здание, в небольшой кабинет, где и увидел впервые своих новых коллег. Начал здороваться и знакомиться со всеми.

Первым меня встретил молодой менеджер, Михаил.

— Привет! — сказал ему я, — Очень приятно познакомиться!

— Привет! — ответил он, — Тоже приятно. Только мы знакомы. Ты меня собеседовал года четыре назад. Тогда я в .NET хотел пойти. Ты меня не взял и в результате я пошёл в Java. А сейчас вот в менеджерах. Вместе будем работать.

Я смутился. Очень часто те, кто не прошёл собеседование, считают, что с ними поступили несправедливо. А тут мне нужно в новом коллективе отношения строить. Кто знает, как мне такие прошлые обидки откликнутся. Со слегка натянутой улыбкой я сказал что-то вроде: «Надо же какое совпадение!» и повернулся к другому моему новому коллеге, Максиму.

— А мы тоже знакомы, — Максим улыбался широко и доброжелательно, — И примерно при таких же обстоятельствах. Ты меня собеседовал 2 года назад и не взял.

— Кстати говоря, — подал голос из угла комнаты третий мой новый коллега, Никита, — Меня ты тоже собеседовал, полгода назад. Я ещё студентом тогда был. Ты меня тоже «прокачал».

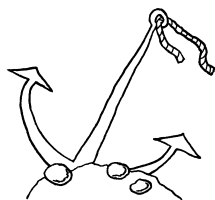
Видимо, моя улыбка стала натянутой до комичности, потому что все трое рассмеялись.

— Да ты не парься, — сказал Максим, — Мне тогда это на пользу пошло. Я составил список тем, в которых «плавал», подтянул их и следующее собеседование вот в это подразделение прошёл.

Никита согласно покивал головой. Он тоже не держал никакого зла. У него собеседование со мной вообще было первым собеседованием в жизни. И оно ему понравилось. Так что когда он набрался чуть-чуть опыта, то попробовал снова прийти в мою компанию, но попал уже в другой проект.

Так вышло, что в небольшой комнатке, где я оказался, три человека (а это была треть людей, что работала тогда со мной рядом), были у меня на собеседовании и не были приняты. Но все мои страхи оказались напрасными. Наоборот, меня встретили, как старого знакомого и до сих пор ко всем тогдашним коллегам у меня очень тёплые чувства.

А мне это послужило очень наглядным уроком, что даже если ты не берёшь кандидата на работу, то вполне может быть, что он прокачает свой уровень и скоро будет работать с тобой (или даже будет твоим боссом). Каждое собеседование — это просто маленький шагок на карьерном пути каждого, и интервьюер должен приложить усилия, чтобы этот шагок оставил по возможности тёплые чувства.



*Константин Борисов*

БРАТЬ ИЛИ НЕ БРАТЬ?

ИЛИ

Как собеседовать разработчика



9 785001 228011

---

Подписано в печать 29.11.2018. Формат бумаги 60x84 1/16.  
Печать офсетная. Усл. печ. л. 6,56. Тираж 100 экз. Заказ 804.

---

Издательство «Перо»  
109052, Москва, Нижегородская ул., д. 29-33, стр. 15, ком. 536  
Тел.: (495) 973-72-28, 665-34-36