# Вопросы работодателю на проект

pdf

- Вопросы сгруппированы в короткую и длинную версию, с повторами.
- Длинная версия для этапа адаптации(onboarding).
- Короткая рассчитана на 20-30 минут обсуждения на предварительной встрече.

## Предварительная встреча

### Rus

1. какие темы спрашивают на интервью? Angular, HTML/CSS, SSR, HTTP, Websocket, NGRX, шаблоны проектирования, тестирование, авторизация/безопасность, анимация/вёрстка, алгоритмы/задачки?
2. Сколько у вас обязательных регулярных совещаний в неделю/месяц? Можете описать график и продолжительность обязательных совещаний?
3. рабочее время по MSK TZ?
4. Какие технологии используются для доступа к средам разработки: DOCKER, VPN, RDP, WEB, AZURE, AMAZON, VDI?
5. Можно ли работать под Mac/Linux?
6. как часто случаются переносы сроков выполнения задач, как к ним относится Заказчик?
7. сколько облачных серверов для разработки/тестирования? Насколько легко их добавить?
8. как давно развивается проект в котором я буду участвовать?
9. размер бэклога?
10. версии фреймворка, библиотек?
11. Время сборки проекта? CI/локально
12. размер техдолга? Динамика за год? Как его оцениваете, как к нему относится Заказчик?
13. с кем я буду взаимодействовать в команде разработки: роль, как давно в проекте?
14. есть ли в проекте внешние подрядчики или другие команды, с которыми я буду взаимодействовать и по каким вопросам?
15. кто в команде работает удалённо(не в офисе)?
16. Вы замеряете текущую скорость/поток команд за неделю/спринт? Какие средние ожидаемые показатели?
17. какой порядок выхода из проекта, есть ли ограничения на минимальные сроки - день, неделя, месяц?
18. мои дальнейшие действия: интервью, тестовое задание, проверка СБ, подписание соглашений?

### Eng

1. What topics/experiences interviewers will ask? Angular, HTML/CSS, SSR, HTTP, Websocket, NGRX, design patterns, testing, auth/sec, animation/layout, algorithms/tasks solving?
2. How many regular mandatory meetings in your teams per week/month? Can you provide the typical schedule and duration?
3. Working time in MSK TZ?

4. What technologies are used to access the development environments: VPN, RDP, WEB, AZURE, AMAZON, VDI?
5. Is it possible to work on Mac/Linux desktops?
6. How often overtime work occur, how does the Customer react on them?
7. How many cloud servers you have for development/testing?
8. How long has the project been developing in which I will participate?
9. The backlog size?
10. Versions of the framework, libraries?
11. Project build time? CI/local.
12. The techdebt size? Last year trends? How you measure/manage the techdebt?
13. Is there any external contractors or other teams in the project with whom I will interact and on what issues?
14. Who will I interact with in the development team? role, when joined the project?
15. Who works remotely in the team (not in the office)?
16. Did you measured current team speed/throughput in tasks per week/sprint per developer?
17. What is the procedure for exiting the project, are there any restrictions on the minimum terms - a day, a week, a month?
18. My next steps: interview, test task, security check, signing agreements?

## интервью

1. сколько времени на интервью у каждого участника
2. какие темы спрашивают на интервью? Angular, HTML, CSS, HTTP, Websocket, NGRX, шаблоны проектирования, тестирование, авторизация/безопасность, анимация/вёрстка, алгоритмы/задачки?
3. готовы ли сообщить мне обратную связь в конце технического интервью(что порадовало, что следует подтянуть)
4. мои дальнейшие действия: интервью, тестовое, проверка СБ, подписание соглашений?

## организация

1. рабочее время по MSK TZ?
2. как ведётся график отпусков, можно ли согласовывать отпуска за полгода
3. какие правила/ограничения согласования внеплановых выходных(dayoff), больничных, отпусков
4. есть ли в проекте внешние подрядчики или другие команды, с которыми я буду взаимодействовать и по каким вопросам
5. график обязательных совещаний
6. как часто случаются переработки, как к ним относится Заказчик
7. что категорически нельзя делать разработчику
8. каковы ваши ожидания, опишите идеального кандидата
9. как ведётся учёт времени: JIRA или что-то ещё заполнять

## проект

1. Сколько у вас обязательных регулярных совещаний в неделю/месяц? Можете описать график и продолжительность обязательных совещаний?

2. Какие технологии используются для доступа к средам разработки: DOCKER, VPN, RDP, WEB, AZURE, AMAZON, VDI?
3. Можно ли работать под Mac/Linux?
4. сколько облачных серверов для разработки/тестирования? Насколько легко их добавить?
5. технологии проекта: 3D графика, ИИ, большие данные, высокие нагрузки, pixel perfect, мобильный клиент, ГИС, state management, интернет/локальный, blockchain
6. как давно развивается проект в котором я буду участвовать?
7. размер бэклога?
8. версии фреймворка, библиотек?
9. Время сборки проекта? CI/локально
10. как работаете над качеством кода
    1. как оцениваете технический долг, как часто его оцениваете, как к нему относится Заказчик
    2. есть ли правила улучшения кода - соглашения, линтеры
    3. принято ли исправлять ошибки(рефакторить, подготавливать код) перед тем как реализовывать новый функционал
11. на сколько спринтов вперёд готова аналитика/дизайн
12. когда наступает/заканчивается новая/текущая веха
13. есть ли у вас документация, и как часто её обновляете: тест сценарии, дизайн руководства, бизнес логика, описание API, соглашения по коду
14. есть ли в проекте согласованная архитектура, подходы к именованию и расположению файлов
15. что буду делать из перечисленного: рефакторинг, проектирование, документирование, программирование, тестирование, вёрстка, дизайн, devops(webpack/gulp/docker), администрирование(linux/nginx)
16. какая доля кода написана людьми уже ушедшими из проекта, как давно они ушли

## инструменты

1. Время сборки проекта
2. Какие технологии используются для доступа к стендам: VPN, RDP, WEB, AZURE, AMAZON
3. Какие ОС используют разработчики: Mac, Win, Linux
4. какие инструменты используют разработчики: VSCode, JIRA, SWAGGER, CONFLUENCE, FIGMA
5. какие инструменты используете для общения: skype, hangouts, slack, teams
6. Какие среды развёрнуты для разработчиков: test/QA/stage/dev
7. какие средства автоматизации дизайна/сборки/тестирования(UI, API): postman, selenium, jasmine, vagrant, chef, jenkins
8. какие средства совместной работы над аналитикой/дизайном: figma, draw.io, google docs, confluence

## методики

1. как часто случаются переносы сроков выполнения задач, как к ним относится Заказчик?
2. как договариваетесь про дизайн UI и контракты API: кто делает, согласует, в каком виде и где фиксируются, примеры и описание
3. как оцениваете вклад разработчиков в проект, их скорость/качество работы
4. как и кто будет меня оценивать, как он поймёт, что я хорошо или плохо прошёл испытательный срок

5. что используете из гибких методологий работы: спринты, ретроспективы, вехи, стендапы, доски, покер
6. что входит в описание задачи: макет, API, типы данных, сценарии, схемы
7. как принимаете решения по добавлению задач в спринт, как часто меняете/сдвигаете задачи после начала спринта
8. есть ли правила ранжирования важности задач и ошибок
9. есть ли правила проверки кода(code review) и работы с репозиторием(commit, merge), какие используются подходы git-flow, github-flow.

## коллеги

1. есть ли у вас отдельные аналитики(BA)/тестировщики(QA)/админы(devops)
2. кто может регулярно и охотно отвечать на мои вопросы по коду и предметной области
3. с кем я буду взаимодействовать в команде разработки: роль, как давно в проекте?
4. кто покинул команду за прошлый год? Почему?
5. Довольны ли разработчики кодом, который пишут?
6. кто в команде работает удалённо(не в офисе)
7. Вы замеряете текущую скорость/поток команд за неделю/спринт? Какие средние ожидаемые показатели?

# Признаки здорового проекта

1. можно работать удалённо под linux/mac(есть впн, docker, облако)
2. коллектив с хорошим чувством юмора
3. нет текучки кадров
4. руководство проекта понимает необходимость сдвижки сроков выполнения задач, есть понятные и чёткие правила на этот счёт
5. для оценки качества работы используется не только попадание в сроки, но и пропускная способность(кол-во задач в неделю), и качество кода(% без ошибок)
6. по метрикам в JIRA/ESLint видно, что возможно поддерживать разумные сроки выполнения задач
7. самые опытные разработчики довольны кодом, который они пишут и поддерживают
8. можно работать не более 40 часов в неделю. Нет регулярной необходимости в сверхурочной работе, а если она возникает, то нет проблем с оплатой по сверхурочному тарифу или компенсацией выходными
9. руководство проекта умеет прояснять и обсуждать/корректировать свои ожидания, проводит совещания вежливо и конструктивно
10. вход и выход задач контролируется руководством проекта не поштучно, а за период. Задачи поштучно контролируются, при необходимости, техлидом
11. есть управляемый процесс удаления устаревшего/неиспользуемого функционала
12. есть процесс регулярной оценки качества кода проекта в целом
13. существующих инструментов(сервисов, виртуальных серверов) для командной работы достаточно, работают бесперебойно
14. руководство проекта прислушивается к просьбам/проблемам разработчиков, реагирует конструктивно и без волокиты
15. используется гибкая методология организации труда: scrum/спринты, фиксация охвата работ