

кто я?	начальный	средний	продвинутый
кого оцениваю?			
Технический кругозор			
администрирование			
Сети			
теория	знает базовые теоретические основы (модель OSI) отличает tcp от ip знает для чего нужен DNS	понимает почему и как случаются потери и задержки, слова вроде VPN, BGP не вызывают вопросов. закрывает доступ к tcp трафику на порту 80 используя фаервол не составит проблем	понимает для чего и как строится маршрутизация способен поднять сеть для небольшого количества узлов, позаботившись о ее безопасности
навыки	ping, telnet, traceroute	iptables	dig, tcpdump, pf
ОС			
	знает ОС бывают разные, знает базовые отличия	понимает чем отличаются потоки от процессов, примерно представляет как выделяются ресурсы	примерно понимает как реализована система безопасности, как работает планировщик ресурсов и тд
виртуализация			
	подозревает, что бывает разная виртуализация и примерно представляет о чем это. способен запустить windows в virtualbox	понимает отличие между гипервизором и паравиртуализацией. Может поднять окружение на docker или запустить гипервизор и настроить виртуальных машин	может выбрать какой тип виртуализации применим в конкретной ситуации, хорошо понимает ограничения/возможности и как посчитать ресурсы. Способен развернуть тестовый стенд на выбранном окружении и поддерживать в количестве n
инструменты (*nix)	top, ps, cat, grep	sed/awk, shell-scripting	iotop, netstat
devops			
базы данных	все делает вручную	использует простые sh скрипты	умеет и применяет специализированные инструменты (puppet/salt/fabric/ansible/chef и тд)
теория	может рассказать в чем основная идея реляционных БД и почему они называются так	понимает в чем разница реляционных и нереляционных БД. какие зачем и куда	понимает как устроена типичная СУБД, может рассказать об этом без подготовки
практика	select + simple joins, update/insert/delete	outer/inner joins, функции (sum, min, max и тд), группировка	indexes, views, хранимые процедуры, триггеры
программирование			
теоретические основы (Programming 101)	знает, что есть разные ЯП, может догадаться почему их так много. Понимает что в ЯП есть примитивы, способы комбинации и способы абстракции. Может дать хоть какие-то примеры примитивов, способов комбинации и способов абстракции	основные парадигмы и концепции. Сходу расскажет чем отличается функциональное программирование от объектно-ориентированного; многопоточность, чем отличается процесс от потоков, основные принципы синхронизации	что такое компилятор, как работает интерпретатор/компилятор в общих чертах; рекурсия, метапрограммирование
python	типы данных, для чего нужны функции и классы	наследование, декораторы и знание стандартных библиотек	метапрограммирование

golang	типы данных, для чего нужны функции и классы, как собрать	работа с процессами/потоками и знание стандартных библиотек	участие в написании тестов совместно с командой разработки, управление зависимостями
алгоритмы архитектуры	базовые алгоритмы (очередь/стек/сортировка)	деревья, хеш таблицы	синхронизация, map reduce
паттерны	знает, что в принципе это все неспроста и должны быть какие-то общие методы и решения. Может перечислить пару простых шаблонов	знает широко известные шаблоны, может применить при необходимости	понимает какие к месту, а какие нет, способен создать собственные
архитектура ПО	понимает, почему часто выделяют фронтенд и бекэнд, как обычно взаимодействуют разные части друг с другом	уровни изоляции; SOA, EDA (event-driven), ACID и прочие аббревиатуры способен расшифровать и объяснить разницу	распределенные вычисления - когда, о чем, зачем; нейронные сети и прочее
менеджмент			
процесс разработки			
ЖЦ ПО/методологии	может рассказать про водопад и что-нибудь про скрам	понимает как адаптируются разные методологии	может применить теоретические знания и адаптировать их
CI/CD	понимает зачем CI	понимает как сделать CI	может поддерживать и организовать с 0
тестирование			
bug hunting	останавливается как только увидел ошибку, отчеты об ошибке читаемые, но не содержат ничего дальше скриншотов	способен провести исследование и выявить общий источник найденных проблем, способен аргументировать логам/скриншотами/прочей дебажной информацией	способен провести исследование и выявить общий источник найденных проблем, иногда вплоть до строки кода
тест дизайн	знает несколько основных техник тестдизайна (как минимум про классы эквивалентности)	способен составить тестовую модель, знает когда нужны решающие таблицы, а когда pairwise	способен оптимизировать тесты не уменьшая покрытия; использует знания из соседних областей для улучшения покрытия понимает какие типы тестирования можно применить еще и как
тест менеджмент	умеет работать по задачам, способен дать оценку на тестирование конкретной задачи	понимает, что есть разные приоритеты и возможности. умеет находить компромиссы. способен организовать тестирование в заданные сроки с привлечением других членов команды.	может сделать выводы и предложить улучшения. умеет использовать стандартные инструменты (диаграмма Ганта и тд)
работа с требованиями	читает требования, внимательно следит, что все сделано ровно так как написано	анализирует требования, выявляя ошибки, противоречия и неполноту	умеет выявить недостатки еще на бумаге и предсказать возможные проблемы в будущем, способен предложить решения по устранению
общий кругозор	знает что существуют разные виды тестирования, способен рассказать в чем отличия хотя бы 3 из них	знает основные виды тестирования, может рассказать каких затрат требует тот или иной вид	знает вспомогательные/редко используемые виды тестирования, может рассказать зачем и с какими затратами можно их провести

инструменты	таблицы/mindmaps, бумага, карандаш, curl, browser console, any log viewer	linkchecker, shell scripts, grep/awk	jmeter/selenium etc, debug tools
стандарты	-	ISO9000, ISO 9126, isqtb	IEEE 829-1998, ГОСТ Р ИСО/МЭК 12119-2000
документирование			
интервьюирование	способен сформулировать вопросы, найти правильного человека и получить ответы	понимает как фиксировать полученные результаты, способен устранить противоречия и недомолвки (убедиться, что оба понимают вещи одинаково)	умеет проводить групповые интервью, способен добиться консенсуса
написание/поддержка	способен описать полученные результаты в читаемом виде	умеет правильно структурировать и оформлять информацию. умеет оформлять документацию с переиспользованием контента, принятых стилей и тд	может обеспечить корректность и актуальность документации (кросс-ревью и тд)
soft skills (Emotional Intelligence)			
письменное общение	понимает как написать письмо, знает общие правила деловой переписки	может организовать внятную переписку с правильным выбором участвующих, понимает нюансы оформления дискуссии	может направить дискуссию в нужное русло и добиться однозначного результата, используя только письменное общение
организация и ведение встреч	может принять участие в уже организованной, готовится к встречам	понимает когда стоит организовывать, когда нет, умеет определить оптимальный состав участников	способен держать участников в русле обсуждаемой темы, может распределить роли и следить за выполнением
self management/time management	может придти вовремя на встречу	соблюдает дедлайны, выстраивает свой рабочий процесс преимущественно самостоятельно	может обучить коллег основным принципам и личным примером
adaptability	морально готов к изменениям	легко адаптируется	быстро адаптируется и использует преимущества изменений
conflict resolution	решает на интуитивном уровне	знает какие типы конфликтов могут быть и как можно действовать в каждом случае	Предотвращает возникновение конфликтных ситуаций. В случае сложившихся конфликтных ситуаций разрешает их наиболее оптимальным путем.
знание предметной области			
	понимает как работает система в рамках своего компонента, знает базовую интеграцию с другими	способен связать и протестировать несколько компонентов	понимает как работает система в целом, какие есть особенности и нюансы
english			
	pre-intermediate	intermediate	upper intermediate