

Перечень тем для изучения для уверенного прохождения собеседований

1. SDLC + роли команды + артефакты

Что знать:

- SDLC (Agile, Waterfall, Iterative, V-model)
- этапы разработки и что происходит на каждом
- роли: Product Owner, Business Analyst, System Analyst, Architect, QA, DevOps
- ключевые артефакты: SRS, BRD, Use Case, BPMN, Sequence, ERD, API specs

Практика:

– Уметь рассказать 2–3 проекта: что ты делал, какие артефакты создавал.

2. Требования

Темы:

- бизнес / пользовательские / функциональные / нефункциональные требования
- бизнес-правила и ограничения
- acceptance criteria
- как работать с конфликтующими требованиями
- приоритизация требований (MoSCoW, Kano, RICE)

Практика:

– Написать самостоятельно: user story → acceptance criteria → use case.

3. Архитектура (ключевой блок)

Что выучить:

- Монолит vs микросервисы
- SOA vs microservices
- ESB, API Gateway
- что такое оркестрация и хореография
- транзакции, ACID consistency
- кэширование (Redis), очереди, retry
- почему shared database - антипаттерн

Практика:

– Уметь устно построить C4: Context → Container → Component.

4. Интеграции: синхронные и асинхронные

Синхронные:

- REST
- SOAP
- gRPC
- GraphQL
- RPC

Асинхронные:

- Kafka
- RabbitMQ

- Webhook-и
- файловый обмен
- pub/sub

Что уметь объяснять:

- когда выбирать sync vs async
 - retry, idempotency, DLQ
 - offset, partition, consumer group
-

5. API / HTTP / REST

Что обязательно знать:

- ресурсы, методы: GET/POST/PUT/PATCH/DELETE
- коды ответов: 200/201/400/401/403/404/409/500
- идемпотентность
- уровни зрелости REST
- структура URL
- query params vs path params
- версионирование API
- webhooks и retry

Практика:

– Спроектировать 5–10 REST-эндпоинтов: поиск, создание, отмена, callback.

6. Очереди и событийная архитектура

Обязательные темы:

- Kafka: topic, partition, offset, consumer group
- RabbitMQ: exchange types, routing key, queue, ack/nack

- разница queue vs log
- event-driven design
- как делать retry, idempotency, DLQ

Практика:

– Объяснить устно: как проходит событие «Платёж → callback».

7. Базы данных + SQL + ERD

Что выучить:

- разница реляционные / нереляционные
- нормализация: 1NF / 2NF / 3NF
- денормализация - зачем
- JOIN: inner, left, right, full
- индексы: когда помогают, когда нет
- транзакции
- шардирование и репликация
- ClickHouse vs PostgreSQL

Практика:

– Ежедневно решать 10 SQL-задач (group by, having, join, window).
– Спроектировать 5 ERD из головы: корзина, заказы, билеты, блог, CRM.

8. BPMN (очень важный блок!)

Что знать:

- start / intermediate / end events
- message flow vs sequence flow
- exclusive / parallel / inclusive gateway

- boundary timer event
- subprocess
- правила моделирования между системами

Практика:

– Нарисовать BPMN для: покупка билета, оформление заказа, callback банка.

9. UML: Use Case, Activity, Sequence

Темы:

- структура use case (preconditions, main flow, alternative flow)
- activity diagram vs BPMN
- sequence diagram: lifelines, messages, alt/opt

Практика:

– Нарисовать sequence для: «Пользователь → Банк → Callback».

10. Безопасность (часто спрашивают)

Что выучить:

- отличие authentication vs authorization
 - OAuth2, JWT
 - refresh token
 - почему HTTPS = шифрование + целостность + аутентификация
 - OWASP basics
-

11. Нефункциональные требования

Что знать:

- производительность

- отказоустойчивость
 - масштабируемость
 - мониторинг / алёртинг
 - latency vs throughput
 - SLI / SLO / SLA
-

12. Логирование, мониторинг

Темы:

- уровни логов
 - correlation id / trace id
 - метрики (Prometheus/Grafana как концепция)
 - health checks
-

Приоритет изучения (чтобы быстро вырасти)

Если ранжировать важность **для собеседований**, то порядок такой:

1. **REST / HTTP / API**
2. **Базы данных + SQL + ERD**
3. **Интеграции (sync/async)**
4. **BPMN / Sequence**
5. **Архитектура (MSA, транзакции, ACID, кэширование)**
6. **Kafka / RabbitMQ**
7. **НФТ и мониторинг**
8. **UML и требования**

Эти темы закрывают 90% собесов.