



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Лекция №6
Ролевой состав коллектива разработчиков информационных систем

Методы и средства проектирования информационно-аналитических систем

	<i>(наименование дисциплины (модуля) в соответствии с учебным планом)</i>
Уровень	специалитет
Форма обучения	<i>(бакалавриат, магистратура, специалитет)</i> очная
Направление(-я) подготовки	<i>(очная, очно-заочная, заочная)</i> 10.05.04 «Информационно-аналитические системы безопасности» <i>(код(-ы) и наименование(-я))</i>
Институт	Институт кибербезопасности и цифровых технологий (ИКБ) <i>(полное и краткое наименование)</i>
Кафедра	Информационно-аналитические системы кибербезопасности (КБ-2) <i>(полное и краткое наименование кафедры, реализующей дисциплину (модуль))</i>
Используются в данной редакции с учебного года	2023/24 <i>(учебный год цифрами)</i>
Проверено и согласовано «___» _____ 20__ г.	 <i>(подпись директора Института/Филиала с расшифровкой)</i>

Москва 2024 г.

Учебные вопросы:

1. Функциональные роли коллектива.
2. Стандартный список ролей коллектива разработчиков.
3. Аспекты влияющие на формирование команды.

1. Функциональные роли коллектива

Проще всего рассматривать структуру команды разработки в коллективах от 10 человек. В них, как правило, также есть и второстепенные функции (бухгалтерия, юристы, клининг и т. д.). Но их чаще всего отдают на аутсорс. Кроме того, должны быть продавцы, маркетологи и HR.

На начальном этапе директор (Chief) может справляться с частью функций сам (продавать, продвигать услуги, искать и нанимать сотрудников). Такая ситуация характерна, прежде всего, для стартапов, где команда может брать на себя все функции сразу. Так как это недавно запущенный проект, и его цель – окупить инвестиции и получить прибыль в максимально короткие сроки, директор может быть и продавцом, и разработчиком, и курьером. Однако чаще всего там уже есть деление на сферы ответственности. Кто-то взял на себя роль CEO, кто-то — CTO, а кто-то занялся продажами и общением с клиентами.

Попытки перекинуть непрофильные задачи на других членов команды приводят к конфликтам. По мере роста компании необходимо выделять под каждую функцию отдельного сотрудника. В классической компании, занимающейся разработкой программного обеспечения под заказ, так и происходит. Нагляднее всего это видно на этапах создания ПО.

Этапы создания ПО: создание концепции/ТЗ; проработка архитектуры программного обеспечения; создание технической документации; реализация проекта; тестирование и приемка; внедрение; техническая поддержка.

На каждом этапе должен быть свой отдел со своими задачами: проектный офис; отдел проектирования программного обеспечения; отдел тестирования и документирования; отдел разработки; отдел внедрения и сопровождения.

Основной состав группы — это специалисты, полностью занятые в создании нового программного продукта: менеджеры проекта; программисты; тестировщики; разработчики документации; инженерные психологи; технологи по разработке ПО.

Вспомогательная группа — это специалисты, не занимающиеся созданием программ, но, тем не менее, играющие важную роль в реализации проекта: группа менеджмента и маркетинга продукта; специалисты по технической поддержке ПО; администраторы бета-тестирования.

Основные фазы тестирования

Pre-Alpha: прототип, в котором всё ещё присутствует много ошибок и наверняка неполный функционал. Необходим для ознакомления с будущими возможностями программ.

Alpha: является ранней версией программного продукта, тестирование которой проводится внутри фирмы-разработчика.

Beta: практически готовый продукт, который разработан в первую очередь для тестирования конечными пользователями.

Release Candidate (RC): возможные ошибки в каждой из фичей уже устранены и разработчики выпускают версию на которой проводится регрессионное тестирование.

Release: финальная версия программы, которая готова к использованию.

По сути в любой команде, занимающейся разработкой продукта, можно выделить следующие роли в команде:

Developer

Занимается производством программных продуктов.

Это роль исполнителя: руководитель ставит задачу на автоматизацию того или иного процесса, разработчик ее выполняет. Эта роль часто сегментируется:

по разделению ответственности:

Backend developer — разработчик программно-аппаратной части комплексного ПО;

Frontend developer — разработчик клиентской стороны пользовательского интерфейса к программно-аппаратной части.

по платформам: Web, Mobile, Server-Side и так далее.

User Experience Designer (UX)

Занимается производством карт пользовательского опыта. Этот человек изучает и оценивает, как пользователи относятся к разрабатываемому программному обеспечению. На нем лежит ответственность за то, чтобы продукт был прост в использовании, восприятии ценности, полезности и эффективности. Он продумывает и оценивает процессы и сценарии использования ПО. Эту роль ошибочно путают, а порою и совмещают с ролью **UI Designer**. UX и UI Designer отличаются не только предметной областью, но и спецификой мышления. UX Designer больше про аналитику и систематизацию, чем про эргономику и эстетику.

User Interface Designer (UI)

Занимается производством графической составляющей интерфейсов. Этот человек разрабатывает визуальную часть пользовательского интерфейса. Основными целями работы UI дизайнера являются: интуитивность восприятия, простота, юзабилити и эстетика интерфейса ПО.

Quality Assurance (QA)

Занимается проверкой результата. QA занимается тестированием всего, как бы странно это ни звучало. Системный подход специалиста QA позволяет тестировать как программный код, так и продуманность карт пользовательского опыта.

Human Resource (HR)

Занимается первичным подбором кандидатов. Он обеспечивает прозрачное прохождение всех этапов собеседований при трудоустройстве.

Team Leader

Отвечает за работу группы специалистов. Team Leader обеспечивает комфортные условия работы коллектива и поддерживает высокий уровень эффективности команды. Этот человек не обязательно должен знать специфику работы команды досконально. Например, Team Leader в группе разработчиков не обязан быть программистом, ему достаточно понимать как организовать работу, понимать процессы, протекающие во время производства.

На практике часто на эту позицию ставят самых прокачанных программистов, что является *классической ошибкой управления*.

Tech Leader

Отвечает за грамотный аргументированный выбор технических решений. Ответственный выбор стороннего ПО для проекта. Рекомендация по выбору конкретного алгоритма или архитектурного решения при производстве ПО. Определение технических особенностей в процессах производства.

Scrum Master

Scrum, Agile, KanBan, гибкие методологии, и прочие теоретические знания, которые крайне бесполезны без практики и опыта.

Scrum Master — это специалист, который помогает команде применять методологию Scrum правильно, объясняет правила методологии, контролирует их выполнение. Сейчас к командам разработки стали прикреплять роль Scrum Master. Он отвечает за грамотное применение той или иной гибкой методологии (бывает, что даже той, которая не касается Scrum вообще).

Project Manager (PjM)

Отвечает за старт, ведение и сдачу проектных работ. Эта роль классического управленца процессами. Работа над проектом начинается с Project Manager'а, ведётся (ставит задачи), контролируется (контроль качества и эффективности) и сдаётся тоже им. В большинстве компаний Project Manager управляет проектным фондом.

Архитектор (Architect)

Ключевая обязанность архитектора — проектирование архитектуры ПО, т. е. принятие ключевых проектных решений относительно внутреннего устройства программной системы и её технических интерфейсов.

Бизнес Аналитик (Business Analyst)

Напрямую общается с заказчиками продукта и выясняет их пожелания и требования. Задача бизнес аналитика на высоком (абстрактном) уровне понять, чего хочет заказчик, как он видит продукт, который будет разрабатывать команда, цель у продукта и какие задачи он будет решать. На момент общения с заказчиком бизнес аналитик может предлагать свои идеи по улучшению продукта и совместно с заказчиком формировать так называемый *vision*.

Системный аналитик (System Analyst)

Занимается, в основном, анализом данных и принятием решений о том, как будет работать система, какие методы будут использоваться, а также написанием основных технических документов (техническое задание или ТЗ, спецификации). Важная часть работы — функциональный анализ, в результате которого выделяется перечень функций, которые должна выполнять система, а также определение требований к системе.

Технический писатель (Technical writer)

Специалист, который занимается составлением документации в рамках разработки различных программ. Это люди, которые призваны помогать нам овладевать новыми технологиями, будь то модное устройство или новая программа. От них отчасти зависит успех новинки, ведь именно им нужно убедить потенциального покупателя в пользе этой новинки и объяснить, как ей пользоваться.

2 Стандартный список ролей коллектива разработчиков

Разработка ИТ-продукта – это длительный и трудоемкий процесс. В каждом проекте участвует большое количество специалистов. В небольших компаниях часто один менеджер отвечает сразу за несколько моментов, выполняя несколько ролей. В крупных корпорациях такое не приветствуется.

Эффективность команды во многом зависит от того, насколько хорошо налажена коммуникация внутри нее, соблюдаются ли дедлайны и насколько четко распределены роли. В каждом отдельном проекте это индивидуально, так как многое зависит от того, выполняется ли разработка на заказ или внутри компании и от многих других факторов.

Однако имеется некий стандартный список ролей, которые используются в каждом проекте. Рассмотрим их подробнее.

Аккаунт-менеджер

Это «правая рука» заказчика. Именно аккаунт-менеджер является гарантом того, что фирма выполнит свои обязательства.

Задачи:

развитие партнерских отношений между фирмой-исполнителем и заказчиком продукта;

курирование команды, работающей над проектом;

оперативное решение нестандартных ситуаций и вопросов между клиентом и командой или внутри команды;

создание плана выполнения заказа;

анализ рисков и гарант дедлайнов;

документооборот по проекту.

hard skills	soft skills
Навыки работы с облачными хостингами (cloud computing) – AWS, GCP, Azure, DigitalOcean, RackSpace, программами документооборота и т. д. – для ведения документации и отчетности пригодится знание данных программ.	Креативность – поможет в том случае, когда клиенту требуется необычный продукт. Также важное умение при подготовке презентации команды.
UX-design – такой скилл будет полезен при контроле результатов, понимании того, насколько интерфейс удобен пользователю и была ли достигнута конечная цель.	Умение убеждать – хороший навык при работе с заказчиком и выстраивании партнерских отношений.
Принятие решений для решения проблем (decision-making) пригодится при работе в команде.	Умение работать в команде – важное качество, которое пригодится для выстраивания коммуникации с командой и внутри нее.
Управление командами (leadership) для контроля за дедлайнами разных этапов разработки.	Адаптируемость – разные проекты требуют разного подхода.
	Тайм-менеджмент – помогает распределить задачи по их значимости и уложиться в дедлайн.

Продакт

Менеджер продукта в команде. Он отвечает за продукт, поэтому может быть не особо осведомлен в том, как развивается проект и на каком он сейчас этапе. Зато о конечном продукте он должен знать все. В его задачи входит:

продумывание стратегии развития продукта;
определение ЦА продукта, понимание ее болей;
анализ рынка и стратегия продвижения продукта;
знание особенностей продукта, его плюсов и особенностей;
презентация продукта.

hard skills	soft skills
Нетворкинг помогает в продвижении продукта на рынке.	Эмпатия – понимание болей клиента, что помогает закрыть их с помощью продукта.
Анализ и цифры пригодится при обдумывании стратегии развития продукта и анализировании рынка, конкурентов, метрики и ЦА.	Умение давать и получать фидбек – важный навык при работе с ЦА и командой. Обратная связь помогает понять необходимость улучшения продукта.
Выбор модели монетизации позволяет провести монетизацию наиболее эффективно и с максимальной отдачей.	Коммуникативные навыки помогают при подготовке и проведении презентации продукта.
Постановка OKR.	Умение продавать идеи.

*Полезные связи или **нетворкинг** (англицизм от networking – букв. плетение сети: net = сеть + work = работать + ing = суффикс, обозначающий деятельность в какой-либо области) — это социальная и профессиональная деятельность, направленная на то, чтобы с помощью круга друзей и знакомых, работающих или имеющих связи в той или иной сфере, максимально быстро и эффективно решать сложные жизненные задачи.*

OKR (от англ. Objectives and Key Results — «цели и ключевые результаты») — методика постановки, синхронизации и мониторинга целей и ключевых результатов на уровне организации, команды и на индивидуальном уровне.

Позволяет:

- повысить мотивацию сотрудников;
- ускорить работу;
- сохранять фокус на приоритетных целях.

Фидбек — это ответная реакция на те или иные события, действия. Он представляет собой отклик на действия компании или человека

Проджект-менеджер

Иными словами, менеджер проекта. Он дублирует в отдельных функциях аккаунт-менеджера, тоже следит за соблюдением дедлайнов и требований по проекту, ведет документацию. Но если аккаунт-менеджер работает напрямую с заказчиком и представляет его сторону, то проджект-менеджер – это представитель со стороны команды разработчиков и работает именно от ее лица.

В его задачи входит: бюджет проекта; сбор требований по проекту и постановка целей по итогам анализа этих требований; делегирование задач; контроль за дедлайнами; общение с заказчиком по конкретным техническим задачам и нюансам ТЗ; распределение зон ответственности между ключевыми специалистами проекта; сбор и контроль метрических данных проекта.

hard skills	soft skills
Навыки работы с облачными хостингами (cloud computing) – AWS (Amazon Web Services), GCP (Google Cloud Platform), Azure (облачная платформа от компании Microsoft), DigitalOcean, RackSpace, Yandex, MTS, Mail.ru, Selectel, Beeline, программами документооборота и т. д. Нужны для ведения документации и отчетности.	Тайм-менеджмент – помогает распределить задачи по их значимости и уложиться в дедлайн.
Scrum и подобные методики.	Лидерство – полезно при управлении командой.
Управление командами (leadership) необходимо для контроля за дедлайнами разных этапов разработки.	Стрессоустойчивость пригодится при решении спорных моментов и налаживании коммуникации.
Ведение переговоров важно для обсуждения бюджета, сроков и выявления нюансов в техническом задании при общении с заказчиком.	Управление рисками – при изменении в проекте поможет не потерять контроль над изменившейся ситуацией.

Ведение багтрекера – для анализа результатов и эффективности разработок.	Эмпатия – для комфортного выстраивания диалога с заказчиком, например, для понимания, можно ли убедить клиента в необходимости расширить бюджет или нет.
--	--

Система отслеживания ошибок (англ. bug tracking system) — прикладная программа, разработанная с целью помочь разработчикам программного обеспечения (программистам, тестировщикам и др.) учитывать и контролировать ошибки и неполадки, найденные в программах, а также следить за процессом устранения этих ошибок и выполнения или невыполнения пожеланий.

Сейл

Проще говоря, это менеджер по продажам. Его основные задачи: найти клиента; произвести на него хорошее впечатление, убедить, что продукт или решение – именно то, что ему в данный момент необходимо и закроет его боли на все 100%; продать, то есть подписать контракт на поставку товаров и/или услуг.

В идеале **сейл-менеджер** – специалист, который умеет убеждать клиента, что продаваемый им товар самый нужный. К сожалению, анализ того, насколько выгоден контракт клиенту, не является задачей сейла.

hard skills	soft skills
Знание техник переговоров поможет продавать эффективно и избежать маркетинговых ловушек.	Дисциплина и самоорганизация важны для сейла, ведь он постоянно в поисках клиентов, и важно ничего не упустить.
Работа в CRM-системе – важный навык для оформления продаж.	Коммуникабельность – умение общаться, убеждать, производить хорошее впечатление.
Владение техниками продаж поможет продать, заключить контракт. Знание техники прогревов или работа с возражениями.	Стрессоустойчивость – важный скилл, особенно, когда приходится сталкиваться с отказами.
Нетворкинг – чем больше связей и знакомств, тем эффективнее продажи.	Эмпатия – умение чувствовать настроение клиента помогает сейлу продавать эффективнее.

Прогрев — это работа с потенциальным клиентом, в процессе которой компания удерживает внимание пользователя и формирует желание купить продукт или совершить другое целевое действие.

Работа с возражениями клиентов — комплекс мер, благодаря которым клиента удаётся убедить в необходимости покупки. Чтобы она была эффективной, важно не «втучать» товар любой ценой, а распознать потребности клиента и мотивы, которые им движут. Возражением называется форма, в которой клиент выражает своё несогласие с необходимостью ему данного товара, качеством, ценой и/или другими характеристиками.

Архитектор (Architect)

Именно этот специалист принимает на проекте решения о том, каким будет внутреннее строение программной системы и ее интерфейсов. Среди других задач архитектора в ИТ-проекте можно выделить: составление конструкции программного обеспечения (ПО), элементов и их взаимосвязи; знание мировых практик разработки ПО и построение системы в зависимости от задач бизнеса, для которого она разрабатывается; проектирование архитектуры ПО; выполнение требований заказчика к системе: быстрое восстановление, быстрая работа, возможность дальнейшего развития.

hard skills	soft skills
Навыки работы с алгоритмами искусственного интеллекта (AI&ML) для понимания этапов разработки и информирования заказчика.	Умение работать в команде для выстраивания коммуникации.
Знание мировых практик разработки ПО для решения бизнес-задач.	Эмпатия – для понимания болей бизнеса и построения системы так, чтобы она максимально подходила для их закрытия.
Технические навыки – знание языка программирования и многие другие.	

Бизнес Аналитик (Business Analyst)

Этот специалист нужен на любом проекте по разработке ИТ-продукта, ведь именно он составляет, так называемый, Vision, или конечную концепцию продукта, который и предстоит

разрабатывать команде. Он должен хорошо представлять себе конечный вид и функционирование системы.

Задачи бизнес-аналитика на проекте: общение с заказчиком и выявление его желаний; выявление целей, для которых разрабатывается продукт, какие задачи он должен решать; предложение собственных идей по улучшению конечного продукта; формирование совместно с заказчиком документации по проекту, в которой подробно описывается, какой именно продукт разрабатывается, каковы его цели и задачи, для какой ЦА предназначен и основные возможности будущей системы.

hard skills	soft skills
Знание основ экономики бизнеса поможет при создании концепции продукта сделать его максимально полезным для бизнеса.	Творчество, интуиция, опыт – навык хорош для улучшения разрабатываемого продукта. В процессе разработки концепции будет нелишним использовать творческий подход при решении отдельных бизнес-задач.
Основы финансового анализа важны для закрытия связанных с этим задач при разработке продукта для бизнеса.	Коммуникативные навыки помогут при выявлении потребностей заказчика.
Знание принципов интеграции систем и владение инструментарием бизнес-аналитика пригодятся при интеграции нового продукта в уже работающую систему компании-заказчика.	Умение ставить задачи программистам и оценивать сами задачи – навык помогает при общении с командой разработчиков и донесении до них пожеланий заказчика и бизнес-целей, под которые разрабатывается продукт.

Системный аналитик (System Analyst)

Как правило, этот специалист работает в паре с бизнес-аналитиком, но в отличие от него обращает внимание не на то, как должна выглядеть и какие задачи решать система. Главное, что волнует системного аналитика – это то, как система должна работать в различных вариантах использования и какие у нее функции. Именно это детально описывает системный аналитик проекта.

Задачи системного аналитика включают в себя: анализ данных и метрик; принятие решений о том, какие именно методы использовать; составление технического задания (ТЗ); разработка и написание спецификации; составление списка требований к системе; функциональный анализ системы.

hard skills	soft skills
Анализ и цифры – поможет при анализировании метрик и результатов работы системы. С помощью этого выполняется функциональный анализ системы.	Понимание целей оптимизации продукта поможет быстрее сориентироваться в том, какие именно функции системы нужно доработать.
Технические навыки – необходимы при составлении ТЗ и написании спецификации к системе.	Тайм-менеджмент – важный навык для соблюдения дедлайнов.
Владение навыками конфигурирования – помощь при составлении требований к системе и выполнении функционального анализа.	Коммуникативные навыки нужны для выстраивания взаимодействия с командой.

Технический писатель (Technical writer)

Без этого специалиста невозможен ни один проект, так как чаще всего именно технический писатель рассказывает о продукте и ведет его в массы.

Основными задачами такого специалиста становятся:

написание инструкции по эксплуатации системы;

составление документации для сотрудников фирмы-заказчика, обеспечивающих поддержку программы в дальнейшем.

hard skills	soft skills
Анализ и цифры – поможет при анализировании метрик и результатов работы системы. С помощью этого выполняется функциональный анализ системы.	Понимание целей оптимизации продукта поможет быстрее сориентироваться в том, какие именно функции системы нужно доработать.
Технические навыки – необходимы при составлении ТЗ и написании спецификации к системе.	Тайм-менеджмент – важный навык для соблюдения дедлайнов.

Владение навыками конфигурирования – помощь при составлении требований к системе и выполнении функционального анализа.	Коммуникативные навыки нужны для выстраивания взаимодействия с командой.
--	--

Проектировщик

Этот специалист нужен в проекте для построения макетов разрабатываемой системы. При этом он обязательно должен учитывать удобство ее использования клиентом.

Основные задачи проектировщика: создание панелей инструментов, меню и кнопок, которые были описаны в ТЗ; создание макета расположения графических элементов; демонстрирует команде/заказчику черновик продукта, например, как будут осуществляться переходы между элементами и страницами (экранами) продукта.

hard skills	soft skills
Технические навыки – знание языков программирования, элементов системы и методов их визуализации.	Коммуникативные навыки необходимы для установления диалога внутри команды и с заказчиком.
Навыки работы с алгоритмами искусственного интеллекта (AI&ML) нужны для понимания этапов разработки и информирования заказчика.	Тайм-менеджмент – помогает распределить задачи по их значимости и уложиться в дедлайн.
	Понимание целей оптимизации продукта поможет быстрее сориентироваться в том, какие именно функции системы нужно доработать.

Дизайнер (Designer)

Этот специалист отвечает за оформление и внешний вид продукта. Именно он «рисует» все элементы продукта, которые видит заказчик в конечном варианте, подбирает цвета и формы.

Задачи дизайнера выглядят так: определение формы и цвета каждого элемента продукта, чтобы вместе они составляли единую картину; прорисовка графических элементов; отрисовка баннеров и логотипов для продукта; конечное оформление продукта.

hard skills	soft skills
Владение профессиональными программами, например, Adobe Photoshop, Adobe Illustrator, GIMP, Figma, Adobe InDesign, CorelDraw, FontLab, Sketch и т. д.	Насмотренность пригодится для создания современного и стильного продукта, отвечающего всем требованиям времени.
Знание основ дизайна – важно понимать, что такое композиция, цвет, типографика, иметь базовое понимание шрифтов.	Тайм-менеджмент – благодаря ему все работы будут завершены в срок и команда избежит срыва дедлайнов.
Визуальные коммуникации – базовое понимание анимации и 3D-моделинга, а также умение создавать шаблоны.	Вкус и стиль – важные черты дизайнера для создания красивого продукта.

Верстальщик (Web developer / Front end developer)

Именно благодаря верстальщику «оживают» макеты, созданные проектировщиками и дизайнерами. Эти специалисты создают верстку страниц так, чтобы они правильно отображались с мобильного телефона и с ПК.

Задачи верстальщика: постановка правил, как браузер должен отобразить тот или иной элемент на web-странице; создание эффектов переходов и кликабельности; выравнивание текста.

hard skills	soft skills
Технические навыки – знание специального языка разметки HTML и CSS. Знание Bootstrap и других фреймворков полезно в процессе разработки.	Эмпатия – понимание того, что хочет видеть клиент.
Знание основ дизайна.	Коммуникативные навыки.
Знание специальных инструментов, например, сборщиков или инструментов-помощников. Git помогает хранить проекты и управлять ими. Docker поможет упаковать проект со всеми окружениями и зависимостями. Командная строка важна для автоматизации и т. д.	Тайм-менеджмент.

Разработчик / Программист (Developer) (backend & frontend)

Этот специалист в команде занимается реализацией требований, которые ранее были прописаны аналитиками.

Задачи разработчика включают в себя:

воплощение в жизнь функций, которые должна иметь система;

создание логики, которая отвечает за то, чтобы все функции системы выполнялись именно так, как это и было задумано изначально.

hard skills	soft skills
Технические навыки – знание языков программирования, фреймворков, работа с искусственным интеллектом. Знание IDE и средств коллективной разработки (Git и/или других).	Умение работать в команде.
Анализ данных пригодится для поиска ошибок системы и их исправления, для создания логической цепочки действий системы.	Нацеленность на результат.

Тестировщик (Testing Engineer)

Именно он первым получает возможность запустить продукт и пользоваться им. Его основные задачи: выявление ошибок и недочетов в работе системы; давать фидбек по продукту.

hard skills	soft skills
Технические навыки – выявление ошибок, багов и дефектов системы.	Работа в команде.
Навык работы с документацией – поможет при составлении отчетности по продукту.	Ответственность и самодисциплина.

Локализатор

Этот специалист часто участвует исключительно в международных проектах, то есть работает только с теми продуктами, которые будут распространяться не только в стране разработки, но и за рубежом.

В задачи локализатора входит: перевод всех слов и команд на другой язык, где планируется распространение и продажа продукта; контроль за внешним видом переведенного продукта.

hard skills	soft skills
Технические навыки – знание языка, на который будет совершаться перевод.	Ответственность и исполнительность.
Знание основ дизайна поможет подобрать размер и тип шрифта для языка перевода, согласовать длину слов и терминов.	Работа в команде.

Тимлид

Это управленец с техническими навыками. Такой специалист, как правило, требуется, если проект масштабный и техническое управление просто необходимо.

Задачи тимлида, обычно включают в себя:

Набор сотрудников в команду, подбор необходимых для выполнения поставленной задачи специалистов. Важно, чтобы в команде была сплоченность и высокий уровень корпоративной культуры.

Принятие решений по вопросам стратегии разработки.

Делегирование задач между специалистами. Контроль за дедлайнами.

Выстраивание коммуникации с другими отделами, а не только внутри команды. Именно тимлид общается с тестировщиками, дизайнерами и прочими специалистами, решая вопросы, не прибегая к помощи заказчика.

В крупных проектах всегда работает два тимлида, со стороны заказчика и со стороны исполнителя.

hard skills	soft skills
Работа с оборудованием – знание оргтехники и умение с ней работать.	Работа с людьми – умение договориться, презентовать свою команду.
Технические навыки – знание программ, языков программирования.	Управление отношениями – умение улаживать конфликты, поиск компромиссных решений.

Умение анализировать данные.	Коммуникативные навыки.
Управление процессами – делегирование задач сотрудникам, контроль дедлайнов.	

В разработке продукта принимают участие большое количество человек, при этом у каждого из них свои задачи. Чтобы добиться эффективного взаимодействия между ними, важно правильно подобрать команду и ведущих специалистов, имеющих определенные hard и soft навыки. Только в таком случае можно добиться качественного и эффективного взаимодействия, продуктивной коммуникации внутри команды. Важно обращать внимание на масштабы проекта, готовится ли он для распространения на зарубежном рынке.

3. Аспекты, влияющие на формирование команды

Уровни

Как и во многих других профессиях, IT-специалисты делятся на уровни, или, на зарубежный манер – *грейды*.

На примере компании Smart World:

Уровни	Описание
Pre-Junior	Сотрудник с начальными техническими навыками, выполняет задачи проекта/команды только под контролем наставника
Junior	Сотрудник с формирующимися навыками ответственности, требует привлечения наставника при необходимости
Junior+	Сотрудник с сформированными начальными техническими навыками
Pre-Middle	Сотрудник с приемлемым уровнем самостоятельности, при котором формируются навыки ответственности и эффективности
Middle	Сотрудник, полностью самостоятельный, достигает результата за счет личной эффективности
Middle+	Сотрудник с отлично развитыми техническими навыками, тактическим мышлением, планированием и контролем поставленных задач
Pre-Senior	Сотрудник является лидером команды, мыслит и принимает стратегические решения на уровне проекта
Senior	Сотрудник является лидером команды, мыслит и принимает стратегические решения на уровне отдела
Senior+	Сотрудник является лидером команды, обеспечивает условия реализации стратегии компании

Делят на хард скил (60%), софт скил (30%) и общие (10%).

Хард скил на примере «Качества процесса проекта»:

Навыки	Индикаторы
Качество процесса проекта - это навык сотрудника в обеспечении качества производственного процесса, его улучшении, предотвращении производственных потерь и выявлении ошибок в процессе разработки ПО.	Проводит активность по тестированию на всех этапах разработки: требования, дизайн, разработка, выкладка
	Контролирует процесс ревью, регулярно проводит инспекцию покрытия функциональности (в том числе в рамках работы над задачами), ведет и обновляет карту покрытия тест-кейсами проекта
	Не пропускает плохо проработанные либо без аналитики эскизы в разработку. - в эскизе всегда присутствует ссылка на аналитику, либо связанная задача по аналитике, либо подзадача по аналитике - после релиза нет задач на координальную переделку функциональности, или заливок хот-фиксов - если что-то не доделано в аналитике, не пускает в разработку фичу/изменения/улучшения
	Контролирует ход главных задач релиза: - следит за соблюдением сроков передачи на тестирование - следит за флпоу передачи задач (#фронт не должен интегрировать не протестированные API)
	Взаимодействует с тимлидами, техлидами и менеджером проекта и корректирует срок выполнения
	Следит за автоматизацией процесса: от деллоя до автоматического выполнения и контроля тестов: - знает и понимает всю структуру pipeline, которая есть на проекте - при необходимости коммуницирует с девопсами (#изменение пайплайна) - своевременно реагирует на упавшие тесты (либо сам правит, либо делегирует)
	Собирает метрики качества проекта, предлагает новые: - собирает ручную метрики - инициирует добавление новых метрик/удаление лишних - участвует в автоматизации сбора метрик (как заказчик или исполнитель)
	Прогнозирует качество результата выбранных решений: задачи, процессы, планы, инструменты, кадровые ресурсы Обучает любыми средствами: guides, лучшие практики, видеоконференции, воркшопы, мастер классы

Code review (CR) - инженерная практика в терминах гибкой методологии разработки. Она заключается в анализе (инспекции) кода с целью выявить ошибки, недочеты, расхождения в стиле написания кода, несоответствие написанного кода и поставленной задачи.

Эпик — это большой объем работы, который можно разбить на несколько историй поменьше.

Hotfix (с англ. горячее исправление) — означает срочное исправление возникшей ошибки или экстренной ситуации. Исправление ошибки не всегда должно быть долгосрочным, т.е. сделал и забыл, чаще всего — ошибка исправляется «наспех», чтобы приложение продолжало работать корректно, а уже после этого вносятся более корректные исправления, которые позволяют навсегда избежать возникновения данной ошибки.

Фи́ча (от английского feature - особенность, необычное свойство) - это определенная деталь или часть продукта, которая имеет специфические характеристики.

User Flow (англ. пользовательский поток, или путь пользователя) — это схема движения пользователя, наглядный разветвлённый сценарий его взаимодействия с конкретным цифровым продуктом: приложением или сайтом.

Деплой (deploy) — это развертывание и запуск веб-приложения или сайта в его рабочей среде, то есть на сервере или хостинге.

Пайплайн в разработке — это последовательность действий или процессов, которые выполняются для достижения заданной цели, например, создания программного продукта или веб-сайта.

Пайплайн состоит из этапов, на каждом из которых выполняется определенная задача или набор задач. Например, пайплайн для разработки игры может включать такие этапы, как разработка идеи, прототипирование, производство, доработка и тестирование.

DevOps (от англ. development & operations) — это методология разработки, которая помогает наладить эффективное взаимодействие разработчиков с другими ИТ-специалистами. Это набор процессов и инструментов, которые позволяют компании создавать и улучшать продукты быстрее, чем при использовании традиционных подходов к разработке программного обеспечения.

Гайд (guide) — это руководство, которое помогает разобраться в определённой теме, решить задачу или проблему. Бывает в формате пошаговой инструкции, списка советов, чек-листа.

Воркшоп (workshop) — это вид мероприятия, в рамках которого группа людей собирается для совместной работы, обучения и достижения определённых целей. Он отличается от традиционных форматов встреч и семинаров тем, что акцент делается на активном взаимодействии участников.

Софт скил:

Навыки	Индикаторы
Ответственность - это навык сотрудника выполнять поставленные задачи вовремя и с надлежащим качеством (без переделок).	Выполняет задачи вовремя, при выполнении нуждается в сопровождении, просит коллег команды проекта
	Выполняет проектные задачи любого уровня сложности, в редких исключениях обращается к более опытному сотруднику
	Самостоятельно выполняет задачи, при поступлении нескольких задач правильно расставляет приоритеты выполнения
	Определяет уровень сложности и срок выполнения поставленных задач
	Перед выполнением задачи прогнозирует полученный результат и максимально быстро и качественно выполняет поставленные задачи
	Организует (собирает и модерирует мероприятия, встречи и другие активности в команде) команду на выполнение задач
	Решает проблемы и задачи, которые напрямую или косвенно мешают развитию проекта/отдела/компании
	Решает проблемы и задачи, которые блокируют развитие компании и решение которых влияет на прибыль компании

Общие навыки:

Навыки	Индикаторы
Общие (10%)	
Проведение встреч - это навык сотрудника регулировать и улучшать рабочий процесс при помощи встреч с коллегами.	Приходит на большинство встреч вовремя
	Готовиться к встрече: много данных - мало вопросов
	Предупреждает об опоздании на встречу за день
	Пропускает встречи только по уважительной причине
	Высказывает свое мнение на встречах, соблюдает повестку встречи, ценит время собравшихся
	Умеет организовать сотрудников на встречу, не забывает подключать удаленных сотрудников
	Умеет проводить встречи, обсуждает открыто ключевые показатели и цели
	Умеет фасилицировать встречу, обладает ораторскими способностями
	Всегда отмечает успех команды и отвечает за последствия принятых решений на встрече

Фасилитация (от англ. facilitate — «помогать, облегчать, способствовать») — стиль управления, отличающийся от простого управления тем, что её способ не директивный, то есть такой, который не выходит за рамки самоорганизации управляемой системы.

Карта развития **QA-инженера** на пути от **джуна** до **синьора**:

