

Карьерный путь

Карьерная ситуация у каждого человека уникальна. Конечно, некоторые ее особенности легко определить, например место работы, должность и примерный размер оплаты труда. Однако другие оценить уже сложнее — взаимодействие с коллегами, возможности для профессионального роста, баланс между работой и личной жизнью и т. д.

Карьерные пути разнообразны, и не существует единого мнения, какой из них можно считать «плохим», а какой «хорошим». У каждого человека он свой, поэтому лучшее, что вы можете сделать, — это понять, что интересно и подходит именно вам.

Способы начать карьеру в разработке ПО также многообразны; можно, например, получить высшее образование по необходимому профилю или же выучиться самостоятельно. У меня был коллега — он двадцать лет проработал инженером-химиком, а потом самостоятельно научился программировать и стал разработчиком.

В этой главе мы рассмотрим следующие вопросы:

1. Типы технологических компаний
2. Типовые карьерные пути в сфере разработки ПО
3. Оплата труда и «уровни» компаний
4. Центры затрат и центры прибыли
5. Альтернативные подходы к оценке карьерного роста

1.1. ТИПЫ КОМПАНИЙ

Не существует единой классификации компаний, но тем не менее можно определить ряд характеристик, важных именно для программиста. Так, выделяют следующие типы компаний.

Технологические гиганты (Big Tech)

Это крупные публичные технологические компании, такие как Apple, Google, Microsoft и Amazon. Они нанимают десятки тысяч программистов, а рыночная капитализация составляет миллиарды долларов.

Многие программисты мечтают работать именно в таких компаниях: из-за высокой оплаты труда, перспективы карьерного роста выше уровня стафф-разработчика и возможности работать над проектами с многомиллионной аудиторией. Кроме того, в процессе работы программисты сотрудничают с лучшими специалистами из различных отраслей.

Средние и крупные технологические компании

Здесь относятся компании, для которых разработка ПО является ключевой составляющей их бизнеса. Они меньше технологических гигантов и нанимают сотни или тысячи программистов. В качестве примера можно привести Atlassian, Dropbox, Shopify, Snap и Uber.

Такие компании, как правило, предлагают оплату труда, близкую к уровню техногигантов, а также гарантируют карьерный рост выше стафф-разработчика. Конечно, база их пользователей несколько меньше, но программисты все равно работают с продуктом, нацеленным на миллионы клиентов.

Скейлапы

Скейлапы представляют собой стабильно развивающиеся компании с венчурным финансированием. Они уже имеют устоявшийся на рынке продукт, и поэтому стремятся к расширению доступа к рынкам. Они даже могут намеренно работать в убыток, чтобы увеличить свою долю на рынке товаров и услуг. К подобным компаниям относятся, например, Airtable, Klarna и Notion.

Развитие таких компаний обычно проходит быстро: они стремятся оправдать свою капитализацию, привлечь дополнительные раунды финансирования или подготовиться к выходу на биржу.

Одной из разновидностей скейлап-компаний являются так называемые единороги — компании с частной оценкой в \$1 млрд и более. В 2010-х годах таких компаний было относительно немного, и их статус «единорога» означал, что компания впоследствии может стать крупным игроком на рынке. На сегодняшний день компаний-единорогов стало больше, и данный статус утратил свою значимость.

Стартапы

Стартапы — это компании с венчурным финансированием, которые еще не привлекли достаточный объем инвестиций и стремятся к тому, чтобы их продукт соответствовал ожиданиям рынка. Подобные компании основаны на какой-либо инновационной идее, привлекающей клиентов.

Деятельность стартапов сопряжена с риском; они зачастую не имеют достаточных доходов и в своей работе зависят от новых раундов финансирования. Именно для этого им и необходимо привлечь внимание потребителей к своему продукту.

Примером успешного стартапа, который вырос до скейлап-компании, является Airbnb в 2011 году. Основанная в 2008 году, компания привлекла начальные инвестиции от фонда Y Combinator. К 2010 году услуги Airbnb начали набирать популярность, и компании удалось привлечь \$7,2 млн в раунде серии А. В 2011 году компания привлекла уже \$112 млн в раунде серии В, продемонстрировав инвесторам свой потенциал.

Примером неудачного стартапа может послужить компания — разработчик Secret, приложения для анонимного обмена сообщениями. Приложение было разработано в 2013 году и позволяло пользователям анонимно делиться своими секретами с другими людьми. Компания добилась хороших результатов и за два года привлекла \$35 млн инвестиций. Однако в 2015 году она прекратила свою деятельность и возвратила инвесторам часть средств.

Стартапы предоставляют своим программистам наибольшую свободу, но в то же время не могут гарантировать им стабильность. Кроме того, работая в подобных компаниях, сложнее всего поддерживать баланс между работой и личной жизнью, ведь успех в данном случае зависит от способности команды выпустить продукт до того, как закончатся деньги. Рабочую атмосферу в коллективе стартапа зачастую определяют его основатели. В некоторых компаниях царит принцип «Work hard, play hard» («Хорошо поработаем — хорошо отдохнем»), в то время как другие предпочитают обеспечить устойчивую рабочую культуру. Деятельность в рамках одного стартапа разнообразна, состав специалистов зачастую разномастный, а возможности для карьерного роста обширны.

Некоторые стартапы предлагают своим сотрудникам долю акций компании — риск таких инвестиций высок, как и их потенциальная доходность. Если бизнес оказывается успешен и в итоге выходит на биржу или выгодно продается, первые сотрудники компании, владеющие значительной долей акций, получают огромную финансовую выгоду. Так и произошло с Airbnb, когда в 2020 году компания вышла на IPO (первичное публичное размещение акций) с рыночной капитализацией в \$86 млрд. Другим примером может послужить и компания — разработчик сервиса для совместной работы в области дизайна Figma, которая в 2022 году была приобретена Adobe за \$20 млрд.

Традиционные компании, не относящиеся к технологическому сектору, но занимающиеся разработкой собственного ПО

Основная деятельность таких компаний мало связана с технологиями. Некоторые из них существуют уже более 50 лет и были основаны до начала бума разработки программного обеспечения. Другие работают в отраслях, где компьютерные технологии не являются главным источником дохода.

Примеры таких компаний включают IKEA (производитель мебели для дома), JPMorgan Chase & Co (финансовые услуги), Pfizer (фармацевтика), Toyota (автомобили) и Walmart (розничная торговля).

В настоящее время многие традиционные компании уже начали процесс цифровизации и разрабатывают собственное программное обеспечение, применимое в их сфере деятельности. Однако компьютерные технологии в подобных компаниях являются центром затрат, а не прибыли, а потому оплата труда в них ниже, чем в технологических гигантах и большинстве скейлапов.

С другой стороны, традиционные компании, как правило, гарантируют более высокую стабильность, что позволяет выстроить разумные границы между работой и личной жизнью, нежели это возможно в большинстве технологических компаний. Однако карьерных возможностей для программистов здесь значительно меньше, а рост выше позиции стафф-разработчика маловероятен.

Традиционные, но технологически подкованные компании

Отдельный подвид традиционных компаний составляют те, где технологии являются ключевым компонентом предлагаемых ими товаров и услуг, например аппаратного или программного обеспечения. На ранних этапах развития такие компании часто демонстрировали выдающиеся результаты и сейчас представляют собой зрелые, надежные и прибыльные предприятия. Однако со временем их рост замедляется, а гибкая структура, свойственная молодому бизнесу, сменяется более неповоротливой и громоздкой.

К таким компаниям относятся, например, Broadcom, Cisco, Intel, Nokia, Ericsson, Mercedes-Benz и Saab. Большинство производителей различного оборудования, а также автомобилей попадают в эту категорию.

Такие компании обычно менее привлекательны для соискателей, чем техногиганты или скейлапы. Предлагаемая ими оплата труда почти всегда ниже, а организационная система недостаточно гибкая, чтобы быстро внедрять новые методы работы, как это могут позволить себе молодые компании.

Однако подобные организации все же способны поставить перед своими специалистами сложные профессиональные задачи, выполнение которых принесет чувство удовлетворения, а масштаб работы может быть сопоставим с деятельностью технологических гигантов. Они также гарантируют высокую стабильность и позволяют выстроить более устойчивый баланс между работой и личной жизнью. Часто программисты работают в одной такой компании достаточно долго, что делает их карьеру предсказуемой и стабильной, а это значительно менее типично для молодых предприятий.

Малые компании без венчурного финансирования

Самофинансируемые компании, семейный бизнес и lifestyle-компании (существующие для обеспечения образа жизни своего основателя) — все это примеры небольших компаний без венчурного финансирования. Это означает следующее:

1. Отсутствие давления со стороны инвесторов.
2. Нужна прибыль, иначе бизнес ждет крах.

Эти особенности приводят к тому, что малые компании редко демонстрируют стремительный рост и могут показаться консервативными в своем подходе к найму и ведению бизнеса. Однако они могут оказаться хорошим вариантом для тех, кто предпочитает поработать подольше в спокойной, нежели суматошной, атмосфере и при этом иметь приемлемый доход и комфортный темп работы.

Государственный сектор

Правительства разных стран постоянно инвестируют в разработку программного обеспечения. К преимуществам работы в государственном секторе можно отнести стабильность, четко оговоренную и прозрачную заработную плату, а также различные льготы в виде пособий и отпусков.

Среди недостатков — низкий уровень гибкости, бюрократия и необходимость поддерживать работу устаревших систем, не поддающихся корректировкам. Кроме того, в некоторых странах могут возникнуть трудности с последующим переходом специалиста из государственного сектора в частный.

Примером правительственной организации с хорошей репутацией является британское подразделение Digital Service, которое разрабатывает и поддерживает многие государственные веб-сайты. Компания публикует свои проекты на GitHub (<https://github.com/alphagov>), что делает ее деятельность прозрачной и общедоступной. Еще одним примером компании государственного сектора с хорошей инженерной культурой является Британская вещательная корпорация (BBC).

Некоммерческие организации

Подобные компании ставят своей целью служение каким-либо общественным интересам, как, например, Code.org, Khan Academy и Wikimedia Foundation.

Некоммерческие организации обычно предлагают меньшую заработную плату, чем компании с венчурным финансированием, и у них принципиально другая миссия, отличная от генерирования прибыли для владельцев и возврата вложений для инвесторов. Профессиональная среда в таких организациях может быть различной; некоторые могут оказаться отличным местом работы для программистов, но технологии там, как правило, являются центром затрат.

Консалтинговые, аутсорсинговые компании и студии разработки

До сих пор мы рассматривали лишь компании, создающие определенные товары или услуги и для этого нанимающие программистов. Однако существует также и значительный спрос на «аренду» экспертных знаний в области программного обеспечения через специальное агентство или поставщика услуг аутсорсинга.

Аутсорсинговые компании предоставляют необходимое число программистов клиенту, который затем уже самостоятельно решает, в каких областях они будут работать. Консалтинговые компании заключают контракты с клиентами на выполнение сложных проектов под ключ, предоставляя программистов, которые будут выполнять конкретную, заранее определенную работу. Кроме того, они обычно несут ответственность за разработку всего проекта в целом.

К примерам консалтинговых и аутсорсинговых компаний можно отнести Accenture, Capgemini, EPAM, Infosys, Thoughtworks и Wipro.

Студии разработки представляют собой малые или средние организации, которые занимаются менее крупными консалтинговыми проектами, такими как, например, создание веб-сайтов, приложений и других подобных проектов по заказу клиентов. Они также могут заниматься обслуживанием клиентских сервисов. Программисты-консультанты получают подневную или почасовую оплату как штатные сотрудники агентства.

Для программистов, желающих работать в консалтинговых, аутсорсинговых компаниях и студиях разработки, преимуществом станет относительная легкость трудоустройства, что особенно ценно для специалистов с небольшим опытом работы. Подобные компании зачастую испытывают высокую потребность в кадрах, но предлагают меньшую оплату труда по сравнению со своими компаниями-клиентами.

Среди прочих преимуществ стоит отметить доступность обучения для менее опытных сотрудников, а также возможность работать над разнообразными проектами, тем самым получая представление о вариантах деятельности на различных рабочих местах.

Однако работа в консалтинговых компаниях имеет и ряд недостатков. Ниже представлены наиболее распространенные.

- С точки зрения карьерного роста подобные компании обычно не предоставляют возможности для продвижения выше уровня стафф-разработчика, что лишь на одну ступень выше сеньора.
- Объем работы устанавливается клиентом. К услугам консалтинговых компаний обычно прибегают для выполнения тех проектов, которые

компания-клиент считает находящимися вне своей основной компетенции.

- Недостаточное внимание уделяется лучшим практикам разработки программного обеспечения. Клиенты платят за получение краткосрочного результата, а не за работу над долгосрочными задачами, такими как, например, сокращение технического долга.
- Возможные трудности адаптации при переходе в компании, ориентированные на продукт. При работе в подобных компаниях — техногигантах, стартапах и скейлапах — особенно ценится инициативность и способность писать простой в сопровождении код. Поэтому деятельность в консалтинге в течение длительного времени может осложнить адаптацию к новой рабочей среде.

Научно-исследовательские институты и центры

Подобные учреждения являются филиалами университетов или же тесно с ними сотрудничают в процессе осуществления долгосрочных исследовательских проектов. Научные институты могут заниматься как прикладными исследованиями, так и фундаментальными.

К преимуществам работы в исследовательских институтах или центрах можно отнести возможность применить ваши навыки в малоизученных областях, а также стабильность работы без какого-либо коммерческого давления.

Какой тип компании лучше всего соотносится с вашими карьерными целями?

Из всего вышесказанного можно сделать вывод, что существует множество различных типов компаний и организаций, которые предстоит рассмотреть программисту при выборе места работы. Так как же понять, что подходит именно вам?

Вероятность того, что на момент поиска работы у вас будет возможность попасть во все десять перечисленных типов компаний, крайне мала. Поэтому будьте реалистичны и суйте критерии поиска, исходя из ваших обстоятельств. Также будет полезно посоветоваться с друзьями, членами семьи или с кем-то еще из вашего круга общения, кто уже работает в сфере разработки ПО. Узнайте, нравится ли им их место работы, расспросите о нюансах.

Учтите, что между компаниями, относящимися к одному типу, также могут существовать значительные различия, и даже внутри одной организации условия в разных группах могут отличаться. Поэтому работа в техническом отделе компании традиционного типа, но при этом в прекрасной команде может приносить гораздо больше удовольствия, чем работа в проблемной команде, но в крупной технологической компании.

1.2. ТИПОВЫЕ КАРЬЕРНЫЕ ПУТИ В СФЕРЕ РАЗРАБОТКИ ПО

Внутри одной компании программисту предлагается два простых варианта развития карьеры: линейный путь или с развилкой (и двумя путями).

Линейный карьерный путь

Подобный карьерный путь для индивидуального контрибьютора (individual contributor, IC) и менеджера обычно выглядит следующим образом:

Уровень	Индивидуальный контрибьютор	Менеджер
1	Разработчик	
2	Сеньор-разработчик	
3	Стафф/принципал-разработчик	Менеджер
4		Директор
5		Вице-президент по разработке
6		Технический директор (СТО)

Типичный линейный карьерный путь. С каждым уровнем растут как ожидания, так и оплата труда

Для программистов, работающих в небольших или нетехнологических компаниях уровень 3 (стафф/принципал-разработчик) фактически является потолком карьерного роста. Дальнейшее продвижение возможно только после перехода на ветвь менеджмента.

Недостатком такого карьерного пути является то, что многие программисты, не желая заниматься менеджментом, увольняются и устраиваются на работу в другие компании. В результате работодатель может потерять своих лучших разработчиков — они либо становятся менеджерами, либо вовсе покидают компанию, не желая заниматься управлением.

Карьера с двумя путями

Прогрессивные компании, в которых работают более 30–50 специалистов, все чаще предлагают своим программистам два варианта карьерного роста, чтобы тем не приходилось в итоге выбирать между сохранением своей должности с заработной платой линейного менеджера и переходом на позицию менеджера. Такой подход обычно выглядит следующим образом:

Уровень	Индивидуальный контрибьютор	Менеджер
1	Разработчик	
2	Сеньор-разработчик	
3	Стафф-разработчик	Менеджер
4	Сеньор стафф-разработчик	Директор
5	Принципал-разработчик	Старший директор (Senior Director) ¹
6	Distinguished engineer ²	Вице-президент по разработке
7	Fellow-разработчик ³	Старший вице-президент по разработке
8		Технический директор (СТО)

Типичная карьера с двумя путями. Как всегда — больше обязанностей, больше и оплата труда

В компаниях, предлагающих карьерный рост с двумя путями, существует несколько возможностей продвижения:

1. Индивидуальный контрибьютор (IC). Продвижение от разработчика начального уровня до сеньора и выше.
2. Менеджер. Переход сеньор- или стафф-разработчиков на позицию менеджера с дальнейшим продвижением по управленческой ветви. При определенной доле везения можно достичь позиции директора и выше.
3. Переход с позиции IC в менеджеры. Возвращение к роли программиста после работы менеджером и, возможно, обратный переход в будущем. И такая ситуация далеко не редкость.

Каждый карьерный путь уникален

В реальности же специалисты в сфере технологий довольно часто меняют место работы. Большинство знакомых мне программистов переходят на

¹ Senior Director — руководящая должность в крупных компаниях, выше уровня Director, но ниже VP (вице-президента). Промежуточное звено между директорами отдельных направлений и топ-менеджментом. В российских компаниях четкого соответствия нет, но возможны варианты: старший директор, исполнительный директор, директор департамента, заместитель вице-президента. — *Примеч. ред.*

² Distinguished Engineer (DE) — почетное техническое звание, присваивается высококвалифицированным специалистам, которые внесли значительный вклад в разработку, архитектуру или технологии компании. — *Примеч. ред.*

³ Fellow — самый высокий технический титул в сфере IT, присваивается выдающимся специалистам. Fellow остается в технической роли, но при этом влияет на стратегию компании, ее технологическое развитие и архитектуру решений на самом высоком уровне. — *Примеч. ред.*

другую работу каждые несколько лет, меняя траекторию своего профессионального развития. Это создает разнообразие возможностей, в чем можно убедиться, посмотрев профили опытных программистов на LinkedIn. Вот несколько примеров.

Карьерный путь Тани Рейли (Tanya Reilly), автора замечательной книги «The Staff Engineer’s Path»¹, на протяжении всех двадцати лет остается довольно линейным, развиваясь в рамках деятельности разработчика:

- системный администратор (Fujitsu, Eircom)
 - разработчик (Google)
 - сеньор-разработчик (Google)
 - системный штафф-разработчик (Google)
 - принципал-разработчик (Squarespace)
 - сеньор принципал-разработчик (Squarespace).

Двадцатилетняя карьера Никки Райтсон (Nicky Wrightson), на момент публикации книги занимающей должность руководителя отдела разработки, развивалась в чисто инженерном русле. Однако в конце концов Никки перешла в менеджмент:

- разработчик ПО (консалтинговая компания)
 - специалист-консультант и разработчик (телекоммуникационные компании)
 - разработчик (BNP Paribas, JP Morgan, Morgan Stanley)
 - шестимесячный саббатикал²
 - принципал-разработчик (Financial Times, River Island, Skyscanner)
 - технический директор (венчурная студия Blenheim Chalcot)
 - руководитель разработки (topi).

Карьерный путь Марка Цимельзона (Mark Tsimelzon), на момент публикации директора отдела разработки WhatsApp, складывался более тридцати лет — с позиции разработчика до основателя стартапов и руководителя проектов:

- разработчик
 - технический менеджер
 - основатель стартапа (позже приобретенного компанией Akamai)
 - продакт-менеджер (Akamai)

¹ Рейли Т. «Карьера разработчика. Штафф — круче, чем senior». Санкт-Петербург, издательство «Питер».

² Саббатикал — длительный отпуск с сохранением рабочего места. — Примеч. ред.

- основатель стартапа (позже приобретенного компанией Sybase)
- директор разработки (Yahoo)
- корпоративный предприниматель (entrepreneur in residence) венчурной фирмы
- вице-президент по разработке (стартап приобретен Yahoo)
- старший директор отдела разработки (Yahoo)
- вице-президент по разработке (Sysopse)
- главный директор по разработке (Babylon Health)
- директор отдела разработки (Meta).

Наиболее распространенные пути развития карьеры

Индустрия разработки ПО динамично развивается, поэтому программисты постоянно получают множество предложений и, как следствие, возможностей построить успешную карьеру. Как это обычно происходит:

1. **Разработчик «пожизненно».** Специалист развивается в рамках одной и той же позиции, постепенно наращивая опыт (становится сеньором, стаффом и принципалом) и периодически меняя место работы. Он также может переходить в другие технологические стеки, постоянно совершенствуя свои навыки.
2. **От разработчика к специалисту.** Сюда относятся разработчики, которые специализируются в определенной области, например в сфере разработки мобильных приложений или бэкенда, и работают в ней длительное время.
3. **Маятник «специалист — универсал».** Такое происходит, когда разработчик, специализирующийся на какой-либо технологии, переходит на позицию, требующую более широкого спектра навыков, и наоборот. Опять и снова, снова и опять.
4. **Разработчик специализируется в нишевой области.** Например, разработчик становится SRE- или дата-инженером, работа которых частично связана с программированием, но тем не менее значительно отличается от стандартной разработки ПО.
5. **От разработчика к подрядчику/фрилансеру.** После достижения уровня сеньор-разработчика программист становится подрядчиком или фрилансером. В результате зарабатывает он зачастую больше, а заботится о внутренних делах компании или карьерном росте значительно меньше.
6. **От разработчика к техлиду.** Программист становится руководителем команды, но при этом может и не брать на себя решение управленческих задач. И даже при смене места работы такие специалисты рано или поздно оказываются на позиции техлида.
7. **От разработчика к техническому менеджеру.** Переход на позицию технического менеджера и последующее развитие по данной карьерной ветви.

8. **От разработчика к основателю/владельцу компании.** Разработчик, длительное время проработавший в какой-либо сфере, основывает (в одиночку или с партнерами) собственную компанию.
9. **Разработчик переходит на должность, не связанную с разработкой ПО.** Например, в другую технологическую область — на позиции специалиста по связям с разработчиками (DevRel), продакт-менеджера, технического продакт-менеджера (TRM), рекрутера и др. Опыт работы разработчиком на данных позициях остается актуален, но, в свою очередь, открывается возможность исследовать интересующие сферы деятельности.
10. **Маятник «разработчик — менеджер».** Разработчик сначала становится техническим менеджером, а затем возвращается на инженерный путь, и так несколько раз. Такой карьерный путь встречается все чаще, а подробнее он описан в книге технического директора Чарити Мэджорса (Charity Majors) «The Engineer/Manager Pendulum»¹.
11. **Комбинация нескольких путей.** Например, разработчик сначала становится техническим менеджером, затем продакт-менеджером, а уже после — основателем собственной компании.
12. **Нелинейный карьерный путь.** Например, разработчик, который перешел на позицию технического менеджера, берет длительный перерыв в карьере — чтобы посвятить себя семье или сменить профессию, а затем возвращается в отрасль на позицию технического директора. Нелинейные карьерные пути всегда индивидуальны.

В этой книге мы рассмотрим наиболее распространенный карьерный путь внутри одной компании — от разработчика начального уровня до уровня стафф+. И хотя такой карьерный путь довольно типичен для ИС, продвигающихся в рамках одной компании, тем не менее он не является общим для всех программистов. Описывая все разнообразие способов развития выше уровня простого разработчика, я надеюсь убедить своих читателей в том, что не существует единственного «правильного» пути. Возможности и предпочтения у каждого человека разные, поэтому вы имеете полное право выбрать то, что подходит именно вам.

1.3. ОПЛАТА ТРУДА И «УРОВНИ» КОМПАНИЙ

Сложно количественно оценить все особенности того или иного места работы, такие как, например, сложность профессиональных задач, гибкость графика и возможность поддерживать баланс между работой и личной жизнью.

¹ <https://charity.wtf/2017/05/11/the-engineer-manager-pendulum>.