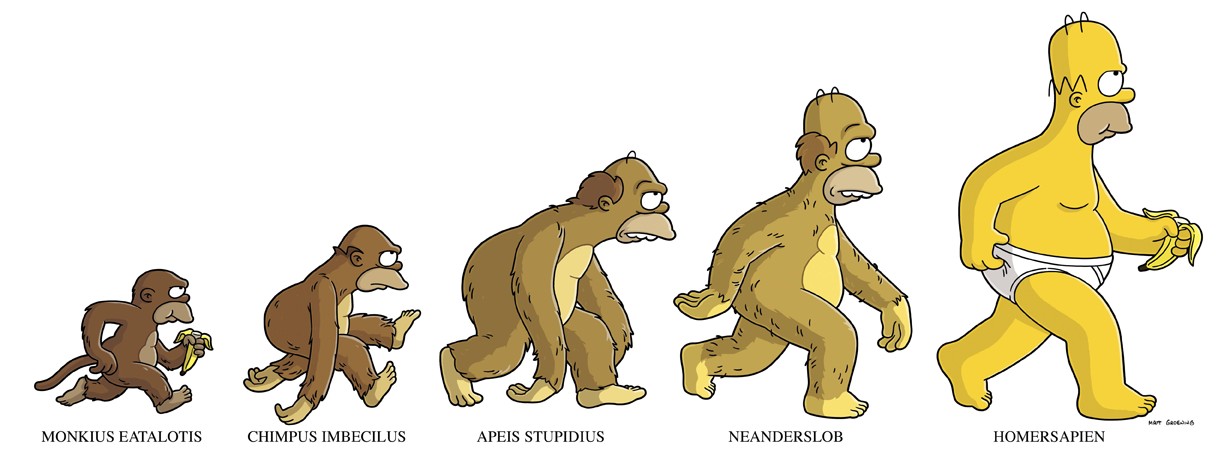
8 советов начинающим программистам или ретроспектива моей карьеры

* [Программирование](https://habr.com/ru/hub/programming/),
* [Карьера в IT-индустрии](https://habr.com/ru/hub/career/)

  
  
Ровно 15 лет назад, в этот самый день, своими дрожащими коленками я переступил порог офиса небольшого провинциального интернет-провайдера. Моя первая настоящая работа в качестве программиста. Знаний почти нет. Тонкий испуганный голосок откуда-то из желудка советует развернуться и убежать подальше, пока окончательно не опозорился. Глубокий вдох-выдох, и вот генеральный директор уже ведет меня в небольшую комнатку-подвальчик, показывает на полуразобранный компьютер, видавший лучшие времена: «Это твое рабочее место. Устраивайся». Так началась моя карьера.   
  
Но перенесемся в настоящее. Сейчас я пишу эти строки, сидя в удобном офисе престижного БЦ в центре Москвы. За плечами работа с крупными международными брендами и разработка сложных fintech приложений. Сотни книг прочитано и десятки статей написано. Мания величия давно вылечена. Менеджерские позиции опробованы и отвергнуты. Душевное равновесие найдено. Любовь к профессии сохранена. Однако это не статья из серии “Какой я молодец. Делай, как я и тоже будешь молодцом”. Эта статья о том, какие ошибки я совершал и что можно было сделать лучше. Эта статья — ретроспектива моей карьеры.

Меняйте компанию, если нет развития

Время от времени я застревал в компаниях, которые явно перерос по уровню. Вроде и деньги платят хорошие, и все тут знакомо и удобно. Приходишь утром, выбираешь привычный режим на кофемашине. Обмениваешься шутками с Катей из бухгалтерии. Садишься в свое кресло, которое давно приняло очертания твоего мягкого места. Открываешь ноут и погружаешься в ежедневную рутину из типичных хорошо знакомых тебе задач. Внимание! Это ловушка!   
  
Работа не должна становиться рутиной. Должны быть вызовы. Не каждый день, может даже не каждый месяц, но если вам слишком легко — пора уходить. Не нужно бояться трудностей. Если на собеседовании вам показалось, что тут ребята решают слишком сложные задачи, и вы точно лажанетесь в первый же день — этот оффер для вас. Ищите максимально сложные задачи, только так вы сможете совершенствоваться, а не деградировать на кухне любимой компании по 4 часа в день

Будьте программистом, а не кодером

Написать код != решить проблему. Код — это только один из инструментов для решения проблем. Ох, как же долог был путь до осознания этого простого утверждения. Как же много времени было потрачено впустую. Нас, программистов, со страшной силой притягивает код. Видя очередную задачу в Джире, мы пробегаем ее глазами по диагонали и скорее-скорее писать код! Как только код написан — мы теряем интерес к задаче и берем новую.  
  
К сожалению, нам очень редко ставят прозрачные и понятные задачи. Перед тем, как открыть IDE, я должен быть на 100% уверен, что понимаю проблему, которую собираюсь решать. Хорошим маркером тут является декомпозиция. Если я могу расписать решение по шагам и знаю, к какому результату приведет каждый шаг — только тогда я открываю редактор и пишу код.   
  
Помните: ваша ценность как программиста не в элегантном коде с экзотическими паттернами от “Банды четырех“. Ни одна компания еще не добивалась успеха благодаря красоте кода (во всяком случае я таких примеров не припомню). Всегда держите в уме конечный результат.

Не бойтесь экспериментировать

Каждый раз, начиная новый пет-проект, я либо начинал его на хорошо знакомых технологиях, либо переходил на них после нескольких недель. Ведь так гораздо быстрее. Какой смысл мучиться с Кафкой, если есть любимый Кролик? Зачем укладывать структуру данных в документы, если она сама легко укладывается в реляционную модель? Каждый раз мысль о том, что нужно побыстрее выйти в прод, мешала мне развиваться. В итоге большая часть моих проектов лежит мертвым грузом на гитхабе, не принеся мне не только мировой известности, но даже новых знаний. Пытаясь угнаться за сроками, которые я сам себе ставил, я просто терял время.  
  
Экспериментируйте. Спешить некуда. Развивайте кругозор. Если вы эксперт в какой-то технологии — используйте эти знания на работе, но вне работы пользуйтесь инструментами, с которыми вы не знакомы. Это сложно психологически, ведь вы будете понимать, что используя хорошо знакомый язык программирования, уже пожинали бы плоды славы от вашего проекта на Мальдивах. А приходится разбираться, почему преобразование чара в инт отдает код символа, а не цифру.

Запускайте пет-проекты

Я давно понял: чтобы быть успешным программистом, мало быть хорошим программистом. Нужно понимать разработку продукта гораздо шире своей специализации. Но как этого добиться? Я могу прочитать сотни книг по маркетингу и они будут бесполезны, если я не применяю эти знания на практике. Я могу подписаться на телеграм каналы лучших продуктологов страны, но что с того, если у меня нет продукта, на котором можно экспериментировать?  
  
С тех пор, как я запустил свой первый пет-проект, мне стало гораздо проще находить общий язык с фронтами, qa, продактами, маркетологами и любыми другими членами команды. Ведь я стал понимать их работу намного лучше. Современное программирование — очень командное, и опыт работы над собственным проектом помогает мне находить точки соприкосновения с другими членами команды. 

Учитесь декомпозировать

У меня всегда были проблемы с внимательностью. Я допускал в коде очень глупые ошибки. Не потому что я что-то не понимал или не знал как сделать. Просто я был невнимателен. Боролся с этой проблемой разными скучными способами (например, следил за минутной стрелкой), но ничего не помогало. Естественно, после каждой такой ошибки моя самооценка резко падала. О каком развитии может идти речь, если самооценка на нуле?   
  
Решение пришло в виде декомпозиции. Меньше задача = меньше кода = меньше вероятность допустить ошибку. Я часто слышу фразу «эту задачу нельзя декомпозировать», но лично я считаю, что любую задачу можно декомпозировать до бесконечности. Если у вас не получается, скорее всего, вы просто не до конца понимаете задачу. 

Изучайте инструменты

Долгое время я пользовался IDE как обычным редактором кода с удобной навигацией. А еще у меня был начальник, который программировал исключительно в mcedit и часто театрально вопрошал: «Кто ты без своего IDE?». Или у меня 24/7 был открыт терминал, но я совсем не умел с ним работать. Я долго жил без статических анализаторов и стайл-фиксеров. Игнорировал горячие клавиши. И не видел в этом никакой проблемы.   
  
Сейчас я подписан на все официальные youtube-каналы своих инструментов. Удивительно, сколько полезного можно из них узнать. Я внимательно смотрю за процессом работы своих коллег и часто перенимаю то, что показалось мне интересным. Также некоторые инструменты показывают нам всплывающие окошки, где рассказывают, как можно улучшить процесс взаимодействия. Не стоит этим пренебрегать.

Участвуйте в opensource

Opensource — это больно. Каждый ваш PR будет рассмотрен под микроскопом. Каждая переменная, каждый символ. Каждый тест будет подвергнут сомнению. И это не ваш тимлид, который: «а, ну ладно, давай так смержим». С этими людьми не договоришься, на них не пожалуешься, их не задавишь авторитетом. И они вам даже не платят. Примерно на двадцать пятом комментарии очень хочется послать все нафиг.  
  
Терпите! Opensource — это отличная возможность прокачать свой скилл, работая с лучшими программистами мира (хотя тут зависит от проекта). Начните с малого. Выберите какой-нибудь несложный инструмент или библиотеку. Зайдите в раздел issues и посмотрите, что можно пофиксить. Надо сказать, что обычно на гитхабе полностью отсутствует токсичность. Если ваш ПР разнесут в клочья, то сделают это очень конструктивно. Во всяком случае, по моему опыту. 

Верьте в себя

Время от времени у меня возникали ситуации, когда хотелось все бросить и пойти работать на стройку. Таскать кирпичи. Этой ситуацией могла быть задача, которая никак не заканчивалась. У которой ежедневно менялись требования. Или технология, которую я не хотел изучать. Или интеграция стороннего сервиса, у которого полностью отсутствовала документация. Или просто функциональность, для понимания которой требовалось образование финансиста, а все продуктологи говорят исключительно на английском. Я психовал, прокрастинировал, увольнялся и срывал злость на котейке.  
  
Есть один прием, который сейчас сильно помогает мне с такими задачами. Просто я всегда помню, что в итоге я справлюсь. Всегда справлялся. И всегда буду. Нужно просто сесть, поизучать, подумать, разложить по полочкам, задать правильные вопросы правильным людям. Главное, верить в себя и оставаться спокойным. Вообще, пожалуй, это самый главный совет, которым я и закончу статью. Keep calm and believe in yourself. Вы справитесь.