

Совет 9

Не клади все свои яйца в чужую корзину

Во времена, когда я руководил группой разработки приложений, я поинтересовался у одного из своих подчиненных: «Что ты думаешь о своей карьере? Кем бы ты хотел стать?» Его ответ меня ужасно разочаровал: «Я хочу быть J2EE-разработчиком». Я спросил, почему не «проектировщиком Microsoft Word» или не «установщиком Real-Player»?

Этот парень собирался строить карьеру на базе *конкретной* технологии, созданной компанией, *сотрудником которой он не являлся*. Что будет, если эта компания исчезнет? Что делать, если привлекательная в данный момент технология устареет? Как можно доверять свою карьеру технологической компании?

Мы, как и отрасль в целом, склонны обманывать себя, считая, что *лидер рынка* — это то же самое, что *стандарт*. В итоге некоторые полагают, что можно сделать продукт сторонней фирмы частью своих характеристик. Хуже всего, когда человек строит свою карьеру на основе продукта, не являющегося лидером рынка, упорствуя в этом, пока его карьера не подходит к концу столь плачевным образом, что остается только переосмыслить проигрышную стратегию.

Давай на секунду остановимся и вспомним, что карьера должна восприниматься, как бизнес. В принципе, построить бизнес, паразитирующий на другом бизнесе, возможно (например, фирмы, производящие продукцию для удаления шпионского ПО и латания дыр в системе безопасности браузеров от *Microsoft*), но для отдельного человека это крайне рискованный путь. Фирма сможет среагировать на изменение в расстановке сил на рынке, например неожиданный рост безопасности браузеров от *Microsoft* (или решение *Microsoft* выйти на рынок программ по удалению шпионского ПО), в то время как индивид вряд ли обладает способностями или излишками финансов, позволяющими резко изменить направление или фокус своей карьеры.

Ориентировка на производителя, как правило, недалководидна.

Ставить во главу угла производителей не имеет смысла еще и потому, что детали реализации программного обеспечения, как правило, держатся в секрете. Ты можешь изучать патентованное программное обеспечение ровно до *границы, за которой начинается сервисное обслуживание*. Это искусственный барьер, призванный не дать пользователям самостоятельно разобраться с возникающими проблемами. Компании возводят его, чтобы получить выгоду, продавая такую услугу, как техническая поддержка. Иногда это делается сознательно, а иногда является результатом защиты интеллектуальной собственности (когда, к примеру, не публикуется исходный код).

Поэтому, хотя целенаправленные инвестиции в конкретную технологию являются *плохой идеей*, если по каким-то причинам ты *вынужден* так поступать, постарайся выбрать вариант с открытым исходным кодом. Даже если ты не можешь или не хочешь использовать подобное решение на рабочем месте, пусть оно станет платформой для глубокого погружения в технологию. Например, у тебя может появиться желание стать экспертом по серверам J2EE-приложений. Не имеет смысла детально изучать способы конфигурирования и развертывания коммерческого сервера (в конце концов, поиграть с параметрами конфигурационного файла может *кто угодно*). Загрузи сервер JBoss или Geronimo с открытым исходным кодом и выдели время, чтобы не только разобраться с механизмом работы этих серверов, но и для изучения их внутреннего устройства.

Вскоре ты обнаружишь, что твои взгляды естественным образом поменялись. В J2EE (или что ты там выбрал в качестве области специализации) на самом деле нет ничего особенного. Разобравшись в деталях реализации, ты поймешь принцип работы высокоуровневых концептуальных моделей. Ты начнешь осознавать, что как на Java, так и на любом другом языке или платформе распределенная архитектура в масштабе предприятия остается распределенной архитектурой в масштабе предприятия. Твой кругозор станет шире, а ум начнет открываться. Ты ощутишь, что концепции и шаблоны, которые понял и осмыслил твой мозг, куда более масштабируемы и универсальны, чем любые технологии производителей. И скажешь себе: «Производители могут приходить и уходить, я знаю, как спроектировать систему!»

Действуй!

1. Начни небольшой проект, создав две реализации. Сначала используй внутреннюю технологию фирмы, в которой ты работаешь, а затем конкурирующую технологию, постаравшись по возможности воспроизвести ее особенности.

Совет 10

Полюби или уходи

Возможно, этот лозунг больше напоминает название какой-то глупой мыльной оперы, но не упомянуть о нем нельзя. Если ты хочешь *достичь высот* в своей работе, ты должен быть влюблен в нее. Если же тебе все равно, это непременно станет заметно.

Переезд в Бангалор вызвал у меня настоящий восторг. Впервые за свою карьеру я ждал, что найду компанию технических специалистов, мыслящих так же, как я, и с такой же страстью к обучению. Я надеялся, что после работы мы будем собираться и вести глубокие философские дискуссии о методологиях и приемах разработки программного обеспечения. Я думал, что индийская Силиконовая долина переполнена мастерами, жаждущими достичь высот в деле разработки ПО.

Но в реальности я обнаружил *огромное количество* людей, интересующихся в основном зарплатой, и *крайне небольшое число* влюбленных в свое дело энтузиастов.

Все как дома.

Разумеется, в тот момент я не понимал, насколько схожи ситуации в двух странах. В США я сменил несколько мест работы, но каждый раз мне казалось, что я просто попал в неудачный город или в плохое окружение. Я считал результаты моих первых попыток трудоустройства исключением из правил и был уверен, что *большинство разработчиков программного обеспечения по праву занимают свои места*. А я просто *пока не оказался в нужном окружении*.

Начал я с работы на кафедре информационных технологий нашего университета, куда меня рекомендовал мой друг Вальтер. Он до-

статочно часто видел, как я работаю за компьютером, и был уверен, что я справлюсь с обязанностями лучше большинства пользователей нашего университета, нуждающихся в помощи. Я в это не верил, так как не имел нужного образования. Я был всего лишь саксофонистом, увлеченным видеоиграми. Но Вальтер заполнил за меня заявление и договорился о собеседовании. Меня взяли, не задав ни одного профессионального вопроса, и мне пришлось сразу приступить к делу.

Сначала я боялся, что коллеги мгновенно распознают во мне шарлатана. *Что этот саксофонист делает среди квалифицированных специалистов?* Ведь я оказался среди людей, обладавших учеными степенями. А я со своим музыкальным образованием пытался делать вид, что знаю еще что-то.

Через несколько дней до меня стала доходить ужасная правда. *Эти люди понятия не имели, чем, черт возьми, они занимаются!* Более того, некоторые смотрели, что я делаю, и *конспектировали!* И это были люди с *учеными степенями в области вычислительной техники!*

В первый момент мне показалось, что меня окружают идиоты. Ведь у меня вообще не было специального образования. Ночи я проводил, играя музыку в барах, а дни — играя в компьютерные игры. Я научился работать с компьютером просто потому, что мне это было интересно. Скажу честно, я учился писать программы, чтобы создавать собственные компьютерные игры. Я возвращался домой после оглушительного вечера в баре и до восхода солнца исследовал Gopher-сайты¹ с обучающими материалами. Затем я ложился спать, просыпался и продолжал изучение до момента, когда снова нужно было отправляться в бар. Я перемежал учебу моими любимыми компьютерными играми, ел и возвращался к экспериментам с протоколом Gopher и компиляторами, которые мне удавалось заполучить.

Оглядываясь назад, можно сказать, что у меня появилась зависимость. Но в хорошем смысле. Основой моего стремления к созиданию

¹ Протокол Gopher представлял собой систему обмена документами, имеющую ту же цель, что и Всемирная паутина. С появлением интернета его популярность резко пошла на убыль.

послужили те же самые побуждения, которые раньше заставляли меня писать классическую музыку или играть джазовые импровизации. Я был одержим идеей изучить все, что только можно. И это не было желанием сделать карьеру в новой области. Более того, многие друзья-музыканты считали, что я безответственным образом отвлекаюсь от своей настоящей карьеры. Но я занимался этим просто потому, что *не мог* иначе.

Именно это отличало меня от моих чрезмерно образованных, но недостаточно продуктивных коллег. Энтузиазм.

Эти люди не имели понятия, *почему* они оказались в IT. Кого-то на эту стезю привела информация о хороших зарплатах у программистов, кого-то толкнула родительская воля, а кто-то в колледже не смог придумать для себя лучшей специализации. К сожалению, все это отразилось на производительности их работы.

Работай потому, что
не можешь не работать.

Вспомнив прочитанные биографии или просмотренные фильмы о людях, достигших высот в своей области, мы обнаружим тот же самый маниакальный энтузиазм. Великий джазовый музыкант Джон Колтрейн упражнялся в игре на саксофоне, пока его губы не начинали кровоточить.

Разумеется, способности в большой степени зависят от природного таланта. Далеко не каждый сможет стать вторым Моцартом или Колтрейном. Но мы можем избежать посредственности, найдя работу, которая нас по-настоящему увлечет.

Это может быть технология или область бизнеса, приводящая тебя в восторг. При этом существуют технологии и области бизнеса, способные тебя полностью деморализовать. Негативным фактором может стать даже тип организации. Возможно, тебе нужно работать в маленьких или, наоборот, в крупных коллективах. Может быть, ты предпочитаешь нетворческий процесс, или, наоборот, лучше себя чувствуешь, когда есть пространство для маневра. Потрать время и найди собственный рецепт.

Ты можешь некоторое время притворяться, но недостаток энтузиазма обязательно скажется как на тебе, так и на твоей работе.

ДЕРЖИ НОС ПО ВЕТРУ

Джеймс Дункан Дэвидсон, программист и фотограф

Моя карьера с самого начала развивалась нетрадиционно. Я всего лишь использовал все подворачивавшиеся мне возможности. Первая возможность появилась, когда я был еще в школе и писал дипломную работу по архитектуре. В возрасте 15 или 16 лет я решил, что хочу стать архитектором, и тратил много времени, вкладываясь в свое будущее. Но основой моей реальной карьеры послужило мое юношеское увлечение электронными досками объявлений. Я был из тех детей, которые обожали модем на 300-бод в семейном компьютере. В конце концов, это привело меня в Сеть, сначала в Gopher, а затем во Всемирную паутину.

Сеть меня захватила. Я быстро создал несколько сайтов, используя преимущества всех имевшихся в моем распоряжении технологий, которые изучал по мере необходимости. В то время я считал эту работу экспериментом по киберархитектуре. Сейчас это звучит чересчур напыщенно и даже в какой-то мере глупо, но именно в таком мире мы жили в первые дни становления интернета. Мы пытались понять, что может принести с собой будущее.

Разумеется, реальная работа по построению будущего интернета происходила совсем не в архитектурных мастерских, а в сфере бизнеса. Достаточно быстро благодаря тому, во что я превратил свою персональную страницу, со мной связалась молодая компания, занимающаяся созданием сайтов для таких организаций, как «Хилтон» или бюро по совершенствованию бизнеса. Они видели плоды моего творчества, и внезапно оказалось, что я обладаю именно теми навыками, которые им требуются. Мне предложили работу с кладом, который по тем временам казался до нелепости огромным. Я принял это предложение, решив, что на некоторое время имеет смысл оседлать волну удачи, подкопить денег и через несколько лет вернуться к учебе.

Был 1995 год. Я слабо представлял, во что все это выльется и куда меня приведет желание освоить новую для меня область. Помогая в создании первой версии сайта отелей «Хилтон», позволяющей бронировать номера в реальном времени, я научился применять различные серверные технологии. Через несколько месяцев мое ученичество закончилось, и я занялся созданием фреймворков на стороне сервера. Сейчас это кажется нелепым, но в тот момент требовался именно такой подход. Я увидел возможность, воспользовался ею и поставил на кон все свои резервы, переосмысливая себя по мере необходимости.

Одно цеплялось за другое. В 1997 году я пришел в JavaSoft, где начал работать над приложениями, исполняющимися на стороне сервера, и, в конце концов,

ДЕРЖИ НОС ПО ВЕТРУ (продолжение)

стал отвечать за спецификацию сервлетов. К сожалению, мои усилия финансировались в недостаточном объеме, кроме того, у меня не было команды, помогающей в решении актуальных задач, в том числе построении эталонной реализации. Однако меня это не остановило, итогом был JavaServer Web Development Kit. Немногие помнят эту программу. А вот о ее следующем выпуске знает большинство «серверных» Java-программистов: Tomcat был выпущен Apache Software Foundation одновременно со служебной программой Ant. История выхода этой версии достойна отдельной книги. Упомяну лишь тот факт, что все это сопровождалось удивительным количеством случайностей, которыми я смог воспользоваться.

Поработав на Sun четыре года и столкнувшись с вопросом «А дальше что?», я решил стать независимым. И начал писать книги для издательства O'Reilly. Разрабатывать программное обеспечение для Mac. Я создал изрядное количество программ, так и не дошедших до выпуска. В конце концов, я занялся разработкой Ruby on Rails. Мне нравится быть независимым разработчиком программного обеспечения, и у меня это хорошо получается. Фактически хобби, которым я увлекался, превратилось в профессию.

Из мечтающего стать архитектором студента я превратился в серьезного технического специалиста, увлекающегося фотографией. Основам фотоискусства меня научила бабушка, а родители всячески поддерживали мое хобби. В итоге, сколько себя помню, рядом со мной всегда была камера. Она стала атрибутом моей жизни. Более того, невыпущенная программа, которую я писал для себя после ухода из Sun, была предназначена для работы с фотографиями.

В 2005 году, через десять лет после того как я прекратил учиться на архитектора и переключился на разработку программного обеспечения, мне позвонили друзья из группы, занимающейся организацией конференций для O'Reilly. Им требовался человек, который мог бы запечатлеть проводимое мероприятие, и меня спросили, не хочу ли я прийти и сделать несколько снимков. Я принял предложение, но несколько вышел за рамки данных мне инструкций. Я с удовольствием снял все важные заседания и с изумившей всех быстротой выложил фотографии на сайт Flickr. Меня снова пригласили на аналогичное мероприятие, и в течение следующих четырех лет благодаря большому количеству клиентов мне удалось открыть свое дело.

Сейчас, когда я пишу эти строки, я все еще программирую время от времени и даже занимаюсь программным обеспечением некоторых клиентов. Но по большей части я зарабатываю как фотограф. Впрочем, ситуация вполне может

ДЕРЖИ НОС ПО ВЕТРУ (продолжение)

снова измениться. Заранее этого никогда не знаешь. Сложно сказать, что принесет нам будущее.

Я всего лишь ловлю возможности. Увидев нечто интересное и вызывающее восторг, я включаюсь в это дело и предпринимаю все, чтобы не упустить свой шанс. Обычно это означает приобретение новых навыков и новых способностей. Некоторым это может показаться тягостным, но по какой-то причине мне нравится учиться чему-то новому. В конце концов, новые способности дают новые возможности. И я никогда не характеризовал себя, описывая навыки. Мне кажется, куда лучше меня характеризует то, что я уже сделал и собираюсь сделать. А навыки — всего лишь способ достижения цели.

Действуй!

1. Найди работу, которая тебе по-настоящему интересна.
2. Начиная со следующего понедельника на протяжении двух недель веди журнал. Каждый будний день, проснувшись, отмечай уровень энтузиазма по десятибалльной шкале: 1 означает, что ты предпочел бы заболеть, лишь бы не идти на работу, а 10 — что высокая температура не может удержать тебя в постели, так тебя захватила идея по поводу решения очередной задачи.

Через две недели внимательно посмотри на результаты. Есть ли на графике пики? Какие тенденции он показывает? Он прижат вниз или стремится ввысь? Какую бы ты поставил себе оценку, если бы это был школьный тест?

На протяжении следующих двух недель каждое утро думай над тем, каким образом ты собираешься сегодня работать на 10. Планируй *сегодняшнее* рабочее расписание так, чтобы на следующий день ты не мог дожидаться момента, когда приступишь к работе. Каждый день фиксируй *вчерашний* уровень энтузиазма. Если и после двух недель ничего коренным образом не поменяется, имеет смысл задуматься о смене работы.