

КАК СТРОИТЬ СВОЮ КАРЬЕРУ В СФЕРЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Прокопчук В.И.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Славинская О.В. – канд. пед. наук, доцент

В работе рассматриваются основные аспекты построения карьеры в сфере информационных технологий, критерии карьерного успеха и пути прохождения собеседования на желаемую работу. В связи с развитием цифрового общества в нашей стране и, соответственно, развитием IT-сферы, изменением требований нанимателей этой сферы к потенциальным сотрудникам, становится актуальным для каждого программиста, находящегося в поиске работы, вне зависимости от опыта, понимание алгоритма построения своей карьеры.

Карьера, как продвижение человека вверх по «лестнице» успеха, может рассматриваться им в разных областях жизни (социальной, профессиональной, личной, научной, в спорте и т.п.). Мы в своей статье остановимся на построении профессиональной карьеры в IT-сфере.

Согласно современной теории менеджмента, карьера – это осознанная позиция человека, стимулирующая его соответствующее поведение, в трудовой деятельности, предполагающей должностной или профессиональный рост.

Вопрос, который интересует каждого, когда он думает об устройстве на работу: «Откуда я возьму опыт работы, если я только учусь, работал в другой сфере?». Это вполне закономерный вопрос, потому что работодатели хотят нанять профессионала, который сразу с высокой степенью вероятности решит поставленные задачи. Где же взять этот опыт, если Вы пока не работали? Дадим несколько советов, проверенных на личном опыте.

1. Участие в проектах с открытым исходным кодом. Мои знакомые участвуют в проектах с открытым исходным кодом. Они создавали свои проекты или поддерживали уже существующие. В этих проектах присутствует и обратная связь от пользователей, и работа в команде, и выпуск версий, и планирование.

2. Просмотр кода проектов с открытым исходным кодом. Я следил за развитием нескольких проектов, просматривал код, который писали другие разработчики. Внимательно разбирался в уже написанном коде и модульных тестах, пытаюсь при этом понять принципы, по которым строились приложения.

3. Курсовые и диплом. Если вы учитесь, то у Вас есть отличная возможность взять проект сложнее и интереснее. Вы можете пробовать различные подходы и по несколько раз переписывать свои программы. Учеба в университете – это уникальное время, когда у вас есть все для саморазвития и индивидуальный консультант в лице преподавателя. К нему всегда можно обратиться в случае затруднений. Главное – правильно рассчитать свое время, чтобы выбранный сложный проект не выбил в колею неуспевающих.

4. Расспрашивайте своих преподавателей. Студенты находятся в самом выгодном положении. У студентов есть человек, который будет отвечать на их вопросы – преподаватель. Но готовы ли Вы их задавать? Во время педагогической практики в университете я преподавал в компании и с удивлением заметил, что лишь малая часть обучающихся по максимуму пользуются этой привилегией. Почему-то большая часть студентов стесняются спрашивать и уточнять. Если вы еще не уверены, стоит ли спрашивать наставника или нет, то приведу известную фразу: «Если спросите сейчас, то будете дураком 5 минут, а если нет – всю жизнь».

5. Продумайте свои цели. Менеджеру по персоналу, менеджеру проекта или ведущему разработчику надо понимать ваши мотивы, которые сподвигли вас искать работу. Любой менеджер думает так: хороший кандидат – это мотивированный кандидат. На собеседовании всегда спрашивают о целях кандидата. Цели должны быть очерчены на год вперед, а в идеале на 5-10 лет.

6. Резюме. Первое, что увидит работодатель, будет Ваше резюме. Резюме должно быть исключительно по делу. Работодатель после просмотра резюме должен знать ответы на вопросы: «Чего Вы хотите?», «Какой у Вас был опыт?».

Найдите готовые резюме. Соберите и проанализируйте как можно больше резюме, которые сейчас есть на рынке. Это даст Вам представление о той области, куда Вы хотите попасть.

Подготовьте резюме. Да, Вы уже присылали резюме. И все-таки принесите еще одну копию с собой. После собеседования можете оставить Ваше резюме, если есть визитка, то вместе с ней. Не стесняйтесь рассказать о том, чего Вы хотите добиться. Скажите о целях явно, без намеков и хождений вокруг да около. Расскажите о Ваших достижениях. Все это, возможно, уже описано в Вашем резюме, но блеск в глазах при рассказе о собственных достижениях не оставит равнодушным собеседующего с Вами.

Соберите требования из вакансий. Еще более мощный инструмент – соберите как можно больше вакансий, из тех, на которые Вы хотите попасть. Проанализируйте требования к кандидатам и сразу поймете, чего Вам не хватает, на что обращать внимание при составлении резюме.

7. Собеседуйте компанию. Вы пришли на собеседование не только для того, чтобы себя показать. Вам нужно убедиться, что Вы захотите работать в этой компании, если Вам предложат должность. Задавайте вопросы о компании, о целях компании, планах. Вы узнаете много интересного, плюс эти вопросы выделяют Вас среди десятка других кандидатов, которые приходили за неделю.

8. Опыт на фрилансе. Я думаю, что многие из программистов, сделали хотя бы один проект для своих друзей, родственников или через сайты. Работа на фрилансе дает возможность экспериментировать с разными языками, делая небольшие проекты или дорабатывая уже существующие. За небольшие деньги, делая проекты с низкими рисками, Вы можете набраться опыта в нужной области. Сейчас на фрилансе представлены все языки программирования и платформы, так что эту возможность нельзя игнорировать.

9. Стажировки. В настоящее время многие компании с удовольствием берут начинающих разработчиков к себе на стажировки. Вы можете поработать в крупных компаниях просто попав к ним на стажировку. На стажировках, конечно, не дают участвовать в критичных и крупных проектах, но Вы сможете пообщаться с опытными разработчиками, возможно, даже поработать с ними в паре. По результатам стажировки могут и на работу пригласить.

10. Пишите тем, кто пишет статьи и книги, и ходите на конференции. Люди, которые пишут публичные статьи или книги, будут очень рады получить обратную связь от Вас. Задайте им вопрос или напишите уточнение по статье, и Вам обязательно ответят.

Перейдем к вопросу «Как правильно строить карьеру, когда Вас приняли в компанию?».

Работа разработчика предполагает не просто кодирование изо дня в день. У каждого, кто занимается разработкой программного обеспечения, есть, как минимум, 10 направлений для развития (Requirements, Design, Construction, Maintenance, Configuration management, Engineering management, Engineering process, Engineering tools and methods, Quality). Если Вы хотите оставаться востребованным специалистом, то Вам нужно быть в курсе IT-мира и постоянно совершенствоваться. Это может звучать банально, но нужно уделять много времени саморазвитию.

Я отдаю предпочтение бумажным книгам, которые заказываю в интернет-магазинах. Читаю примерно 1-2 книги в месяц. Бывает больше, но это – норма. Чтение статей помогает оставаться в курсе событий и тенденций IT-мира. Написание статей помогает упорядочить мысли в собственной голове. Если вы не хотите публичности, то все-равно стоит писать статьи в закрытом блоге или просто в текстовом редакторе.

Нужно узнавать у коллег, в чем Вам еще стоит «подрости». Это самый эффективный, но психологически сложный способ «вытащить себя за волосы вверх». Спросить у своих коллег, чего мне не хватает, как мне стать лучше. Обязательно поинтересуйтесь — узнаете много интересного.

Выстраивая свою карьеру, разработчики часто думают: сначала я поработаю разработчиком, потом стану менеджером проекта. Встречали таких? Конечно, такая схема работает, но есть подводные камни, которые не очевидны для начинающих этот путь.

Менеджер не то же самое, что и разработчик. Хороший разработчик не превратится в хорошего менеджера. Разные книги, разные подходы. Для менеджера наиболее интересны такие области, как психология, управление проектами, методологии, управление рисками, управление персоналом, управление требованиями. Для разработчика – другие. Это и разные границы ответственности. Если говорить про иерархию в организации, то менеджер проекта и разработчик стоят на одной ступени. Отличие только в границах ответственности.

Поэтому нужно четко определить, Вы кем хотите стать.

И поэтому, я еще раз предлагаю Вам явно определить свои цели. Кем Вы себя видите через 1-2 года? В зависимости от этого надо подбирать книги, конференции и RSS-ленты. Есть такая тенденция: начинающие разработчики много внимания уделяют технологиям (языки программирования, платформы). Через несколько лет работы над коммерческими проектами намного больше внимания начинают уделять подходам и принципам разработки. Становится понятно, что «серебряной пули» как не было, так и не будет, что выбор языка программирования зависит от прикладной задачи, как и платформа. Технологии меняются очень быстро, а правильные подходы только накапливаются со временем. Наиболее верным решением будет инвестировать большую часть своего времени в изучение подходов и лучших практик.

Список использованных источников:

1. *How to Make a Resume for a Job – Professional Writing Guide [Электронный ресурс]. – Режим доступа: <https://zety.com/blog/how-to-make-a-resume>.*

2. *Directions in Information Technology [Электронный ресурс]. – Режим доступа: <https://it.unt.edu/directions-information-technology>.*