

IT-эксперт

Филимонов Игорь

---

# Жизнь и карьера успешного веб-разработчика



[www.i-filimonov.com](http://www.i-filimonov.com)

## Глава 1

.....

### **Кодеры и настоящие веб-разработчики**

Количество людей в России, видящих себя веб-разработчиками, из года в год неуклонно растет.

Это связано с престижностью профессии веб-разработчика, высокими зарплатами в этой сфере, возможностью путешествовать, при этом работая и оставаясь на связи из любой точки мира. Этим пестрит современный интернет, с примерами, видео и красочными фото, где программист работает на пляже, в перерывах купается и пьет свежесжатый сок в обнимку с красоткой. Давайте вместе разбираться, так ли это на самом деле, или все подобные истории – лишь миф и пиар?

Начнем с системы обучения. Стать отличным программистом, закончив университет и больше не прикладывая усилий, невозможно, так как большинство программ обучения сильно устарели, да и сами преподаватели не горят желанием реально учить молодого студента. Здесь, на мой взгляд, основных причины две. Во-первых, у них маленькая зарплата, а за такие деньги даже не хочется читать по бумажке, не говоря уже о серьезном обучении и развитии потенциала и конкурентоспособности студентов. Во-вторых, они сами не могут применить то, о чем рассказывают, это очевидно, так как, если бы они это могли сделать, то просто работали бы программистами за 100 000 – 150 000 в месяц и наслаждались жизнью. Я это знаю не по наслышке, так как сам сейчас являюсь аспирантом.

Какие еще есть варианты? Правильно, учиться самому. Здесь большинство начинающих веб-разработчиков выбирают самый сложный путь и поэтому сталкиваются со множеством трудностей, начиная от изучения уже устаревшей информации и заканчивая отсутствием комплексной и проверенной годами системы обучения. К сожалению, интернет переполнен информацией, которая часто противоречит друг-другу, по моему опыту, лишь только 1-2% информации от всего объема является полезной. Поэтому у начинающего программиста в среднем уходит 5-8 лет, чтобы во всем этом разобраться. И здесь исхода два: или он станет профессионалом, что удастся единицам, или он превратится в «горе-программиста» или кодера, которыми переполнен современный рынок. И этого человека ждет отнюдь не сказочное будущее, а работа в душном офисе по 8 часов в день, зарплата в

40-60 тысяч и отпуск 28 дней в месяц. Вот и получается, что жизнь программиста не такая беззаботная и красочная или это не так?

Давайте вместе разбираться, кто такой кодер и чем он отличается от программиста? И, главное, что нужно делать, чтобы случайно или преднамеренно не попасть в их ряды?

Кодер не обязательно самоучка-программист, это скорее человек, который изучил не те инструменты и не в то время. Что значит не те и не в то время? То и значит, что такой подход к написанию веб-приложений либо устарел, либо код и примеры просто взяты с интернета и они, к сожалению, не соответствуют стандартам написания профессионального кода. О стандартах я расскажу подробнее в одной из следующих глав книги.

Теперь главное, как начинающему разработчику стать профессионалом и жить той жизнью, о которой кодер может только мечтать?

Вопрос очень интересный и неоднозначный, так как путь от разработчика к разработчику может отличаться, при этом сохраняются и прослеживаются общие черты.

Во-первых, нужно изучить правильные инструменты, которые актуальны в наше время и которые еще не успели устареть. Единственное, это не так просто сделать, так как придется самостоятельно изучать документацию различных библиотек на английском языке, она не проста и порой не подробная, ряд моментов просто приходится додумывать, поэтому это не самый простой путь.

Какой путь тогда самый простой? Правильно. Можно поступить хитрее и найти себе хорошего наставника, который проведет Вас по пути изучения IT-технологий самым быстрым и безопасным путем. Здесь тоже не все так просто и однозначно, так как высококвалифицированный программист отлично зарабатывает, имеет много опыта и знает себе цену. Поэтому индивидуальные занятия у такого специалиста не могут стоить менее 2-2,5 тысяч рублей за час. Если он соглашается на меньшее, знайте, перед Вами не тот, кто способен сделать из Вас настоящего гуру. Вот и получается, что начинающий веб-разработчик, стараясь сэкономить на своем развитии, в итоге попадает в категорию "Кодера". Не попадайтесь в эту ловушку.

Еще вариант, немного более долгий, но в тоже время довольно эффективный: пройти курсы по профессиональной веб-разработке и программированию. Единственное, к выбору курсов стоит отнестись очень серьезно, так как любая некачественная информация, которую вы усваиваете, делает Вас слабее, учитывайте, пожалуйста, этот факт. Чтобы это понять, у меня ушло несколько лет, Вам повезло больше, так как у Вас есть эта книга, которая предостережет Вас от различного рода подводных камней и расскажет о моих ошибках, которые не стоит повторять. Один совет: выбирайте курсы не по рейтингу школы, а по достижениям и отличным отзывам у преподавателя, так как Вам учиться не у самой школы, а у преподавателя, от которого будет зависеть более 50% результата. Почему только 50% и от кого зависит вторая половина успеха? Правильно. От Вас и больше не от кого. Теперь самое сложное. Как понять, что отзывам преподавателя можно верить? Это не просто определить, согласен, так как в сети, к сожалению, имеется огромное количество «фейковых» отзывов. Варианты докопаться до истины есть, но их немного. У меня, например, есть клуб выпускников, где уже более 350 человек, с каждым из них можно пообщаться и узнать о качестве моего обучения.

⋮  
⋮  
⋮ **Выводы. Помните, что половина успеха зависит от Вас и половина от Вашего наставника/преподавателя, и чем опытнее он будет, тем выше Ваш шанс стать именно веб-разработчиком (а не кодером), которых сейчас очень мало и которые получают все те преимущества, с которых начиналась эта история. Выбор за Вами! Действуйте и будьте готовы инвестировать серьезные деньги в свое профессиональное развитие.**  
⋮  
⋮  
⋮

## Глава 2

.....

### **Карьера в зарубежной компании. Интервью из сердца Вьетнама**

#### **Расскажи немного о себе и о своей компании.**

Меня зовут Денис Леонов и я Software Engineer. Я пишу код с 2010 года, большую часть времени работаю с web-технологиями. В начале 2016 года устроился работать во вьетнамскую IT-компанию «Côt Côt» (cossos.com) и переехал жить в Ханой. Это достаточно крупная IT-компания для Вьетнама. Наши основные продукты — это браузер и поисковая система, которые разрабатываются с учетом культурных особенностей страны.

Корпоративный язык общения — английский, однако в компании работает много русскоязычных сотрудников. Это моя первая работа за пределами РФ. На примере своего опыта пока еще сложно делать однозначные выводы о работе за границей, хотя кое-какое мнение уже есть. К тому же, за пару лет приобрел множество знакомств с людьми, которые достаточно давно приняли решение работать только за рубежом, и имеют неплохой опыт. В общем, несколько интересных вещей рассказать смогу.

#### **Чем на твой взгляд отличается работа в российской и зарубежной компании?**

Технологии часто повторяются, однако культура страны и региона в целом существенно отличается. Работая над различными задачами в России, я привык менять лишь только стек используемых инструментов, переехав в Юго-Восточную Азию пришлось частично поменять еще и образ мышления.

Сам подход к определению целей строится на специфике проживания в регионе. Поначалу многие вещи казались необычными, со временем проникаешься. В целом, в зарубежных компаниях зарплатная вилка выше, чем среднее по РФ. Однако это не гарантирует, что в конце месяца будет оставаться больше денег.

Многие коллеги, работающие в Европе и США, рассказывают про крайне высокий уровень корпоративной культуры. Что важно: отсутствуют постоянные затяжные переработки (в РФ с этим все плохо).

## **В плане уровня специалистов, как отличаются навыки и опыт веб-разработчиков российских и зарубежных компаний?**

Зарубежные компании приглашают к себе только опытных специалистов. Никто не будет перевозить и обучать джуниора — это не выгодно.

Состоявшихся разработчиков обычно нанимают по двум причинам: местные специалисты аналогичного уровня либо стоят существенно дороже, либо просто отсутствуют (быть нанятым по второй причине приятнее:)).

В случае с Юго-Восточной Азией хорошие местные разработчики с достаточным опытом рано или поздно оказываются в Сингапуре, Гонконге или открывают свою компанию с привлечением инвестиций.

## **Отличается ли уровень собеседований в России и за рубежом?**

На мой взгляд, цель такого собеседования — понять не только уровень разработчика, но и насколько быстро он сможет присоединиться к текущим проектам и взять под свою ответственность ряд задач.

Еще интервьюер обращает внимание на то, готов ли человек преодолевать трудности, связанные с переездом. Как оказывается позже, многие люди не готовы к радикальным переменам в климате/культуре/привычках и после испытательного срока просто уезжают.

В остальном собеседования похожи. Да и в целом, если человек разбирается в своих технологиях и имеет достаточный опыт разработки, то ничего крайне сложного на собеседовании не встретит.

## **С какими трудностями сталкивается веб-разработчик, выбирая иностранную компанию? Какие преимущества он при этом получает?**

Самое главное преимущество — это непосредственно возможность поработать в иностранной компании, решая качественно-новые задачи, с которыми никогда не столкнешься в РФ. Даже работая на западную компанию и при этом проживая в России — такого опыта достичь не получится.

Трудности в основном бытовые и связаны с погружением в новую культуру. За пару месяцев вполне привыкаешь. А еще есть языковой барьер. Во Вьетнаме крайне плохо разговаривают на английском. Но вот когда встречаю экспатов из других стран, или вьетнамцев, выросших за границей — понимаю, что плохо уже разговариваю я.

Для текущей работы моего уровня английского хватает. Для переписки уж точно. Иногда общаемся с коллегами лично, вслух. Путаю грамматику, иногда подбираю слова. Всем все равно:), и это никак не сказывается на выполнении рабочих задач. Тут главное — побороть начальный барьер в общении (это делается за один день).

**Есть какие-то подводные камни при трудоустройстве за пределами РФ, с которыми тебе пришлось столкнуться или твоим коллегам?**

Почти всегда договор с иностранным разработчиком заключается на английском языке. Если собственный уровень не позволяет — надо попросить кого-то из знакомых крайне внимательно проверить, перевести и уточнить значение каждого термина.

Также я бы крайне рекомендовал изучить такие вопросы:

- Налоговая система в стране;
- Форма, валюта и метод оплаты труда;
- Каким образом отчитываться о доходах за границей перед органами РФ (это зависит от наличия дополнительных соглашений между странами).

**Помогают ли, на твой взгляд, ИТ курсы у уже опытного разработчика или лучше учиться самому?**

Безусловно помогают.

Самостоятельно обучение крайне важно и без него обойтись нельзя. Однако общение с профессионалом позволяет существенно ускорить процесс освоения технологии или языка. Подобная инвестиция в себя и свое время достаточно быстро окупается.

Без постоянного обучения в ИТ вообще никак нельзя. Наша сфера настолько быстро развивается, что каждые полгода желательно проводить ревизию собственных



знаний и соответствие требованиям рынка. Это относится прежде всего к специалистам, которые хотят активно расти профессионально. Я ни в коем случае не имею ввиду, что необходимо менять технологии. Важно просто быть в курсе новых практик и инструментов.

Опытному специалисту крайне полезно пройти курсы из смежной сферы: например, управление проектами или развитие продукта. А вот фронтендщику неплохо бы заглянуть на курсы по разработке бекенда (и наоборот). Это позволяет коллегам из разных отделов лучше понимать работу друг друга.

### **Расскажи немного о своих целях на ближайшее время**

Сейчас мне интересен полный цикл разработки проекта, от формирования требования и постановки задач на продукт до непрерывной разработки и системы поддержки клиентов. При этом стараюсь не завязываться на конкретный язык.

### **Один совет, который ты хочешь дать начинающему веб-разработчику?**

Можно несколько советов? Я бы хотел это услышать лет 5 назад.

1. Изучайте английский язык, с упором на разговорный. А писать «сочинения» позже будет проще научиться.
2. Инвестируйте в себя: хорошие курсы помогут существенно сэкономить время и уберечь себя от «велосипедостроения».
3. Придумайте и начните свой веб-проект прямо сейчас. Это даст отличную фору при собеседовании.



## Глава 3

.....

### **В вечной погоне за новыми технологиями в IT**

В IT сфере очень быстро одни технологии сменяются новыми, это похоже на отражение луны в ночной реке. Ты ее видишь, она, кажется, близко, на расстоянии вытянутой руки, при этом она так далеко.

Почему одни разработчики постоянно тянутся к звездам, изучая постоянно что-то новое, а другие годами остаются на популярных языках программирования, технологиях, библиотеках и фреймворках. При этом и те, и другие чувствуют себя комфортно и хорошо зарабатывают. Давайте вместе разбираться, какой путь наиболее выигрышен для веб-разработчика в перспективе.

---

*Знание — это абсолютная ценность нашего времени...*  
М. Горький

---

#### **Путь первый. Изучать постоянно что-то новое**

Такой разработчик видит основную ценность в самих знаниях, для него, несомненно, важна оплата его труда, однако она не является приоритетной. Он всегда изучает что-то новое для себя, чтобы быть на волне (в тренде), иногда, чтобы самоутвердиться и показать всем, сколько он знает и как он велик. Отличить разработчика, который изучает для себя, от второго типа можно по его проектам и карьере. Если карьера идет в гору, а в портфолио появляются новые инновационные проекты – это первый тип.

---

*Мы точно знаем, только когда мы знаем мало; с ростом знания возрастает сомнение.*  
И. Гёте

---

Обычно такие люди долго не задерживаются на одной должности, так как очень часто есть определенный стек технологий, и без роста нельзя перейти на новый уровень, где необходимы инновационные знания, навыки и умения.

В итоге, если программист не растёт, то независимо от количества языков программирования, изученных фреймворков, библиотек, на практике применяется лишь 20-30% знаний, остальные 70-80% тормозят регламенты и уже устоявшийся стек технологий в компании. Давайте посмотрим минусы и плюсы постоянного изучения нового.



### **Из минусов:**

- Есть один большой минус, который, к сожалению, не лежит на поверхности. Он заключается в том, что чем больше знаний имеет разработчик, тем больше времени он тратит на принятие решения и тем больше он сомневается в правильности своего выбора. Это нормально.

Когда я знал 2 языка программирования (Си и PHP), я всегда приложения для компьютера (десктопные приложения) писал на Си, а для веб-разработки использовал PHP. Затем в мою жизнь пришли фреймворки (Yii, Simfony, позже Laravel), и выбор стал намного сложнее. Очевидно, что фреймворки сильно ускоряют процесс разработки, однако их много, и не всегда очевидно, какой из них лучше выбрать для конкретного проекта. Затем в моей жизни появились новые языки программирования, такие как Java и Python (с фреймворком Django), выбор стал еще более неоднозначный.

- Второй минус – рутина. Первое время жизнь проста и интересна, однако по завершении 2-3 проектов, стек технологий становится привычным, и работа разработчика превращается просто в повторение уже известных шагов для достижения определенного результата. Со временем разработчика все чаще начинают посещать вопросы, связанные со смыслом жизни и правильности выбора своего пути.

#### **Из плюсов:**

- Очень часто к человеку, который владеет более обширными и уверенными знаниями. более лояльно относится руководство компании, да и коллеги чаще обращаются за советами или за помощью при выборе технологий.
- Проще стать руководителем команды или отдела разработки, так как разработчики хотят видеть своим руководителем подкованного, знающего и умеющего применять эти знания на практике руководителя.
- Учиться очень интересно. О рутине не может идти и речи, так как сегодня ты изучаешь React.js, а завтра уже реализуешь проект на Vue.js.

#### **Путь второй. Изучить один стек технологий для всех проектов**

В этом подходе тоже есть свои плюсы и минусы.

#### **Из плюсов:**

- Относительно много свободного времени, так как не нужно его тратить на постоянное получение новых знаний и изучение новых библиотек и фреймворков.
- Уверенные знания одного стека технологий, так как технологии одни и те же от проекта к проекту. Изменения и отклонения минимальны.
- Высокая продуктивность и скорость разработки, так как не нужно разбираться в новых инструментах и технологиях. Все привычно и уже ни один раз проверено и опробовано на практике.

---

*Привычка –  
вторая натура.  
Аристотель*

---

### **Из минусов:**

- Для освоения новых технологий требуется время. Первое время на разработчика действует сила привычки, из-за этого нужна раскочка перед началом интенсивной учебы, особенно, если до этого ничего не менялось годами.
- Если такой разработчик возглавляет команду, ее участники с более уверенными знаниями могут начать покидать проект, иногда даже компанию.
- Сопротивление изменениям. Это как история с крабами в ведре. Один краб может преодолеть все препятствия и выбраться на свободу. Если крабов несколько, то остальные непроизвольно тянут более инициативных крабов обратно в ведро. В итоге, получится шикарный ужин из всех крабов.

### **Какой путь я выбрал для себя**

Долгое время я развивался по второму пути, изучил один стек технологий (PHP/Yii + JS/jQuery + HTML/CSS) и несколько лет использовал его для реализации проектов, этих технологий более чем достаточно.

Затем был период в моей жизни, когда я практически не программировал и все время посвящал учебе. Так я переместился на первый путь развития.

За это время я узнал много нового: Python с его удивительным фреймворком Django, библиотеку React.js, Vue.js и т. д.

Начал применять новые знания на практике, однако в тот период они оказались не очень востребованными. Я был вынужден на несколько лет вернуться к старому, уже проверенному годами, стеку технологий.

Только сейчас я начал применять уже давно полученные знания в полную силу, так как теперь это стало новым стандартом для ряда компаний.

Отсюда я сделал вывод, что учиться нужно волнами: один год что-то изучаешь, затем год-два применяешь знания на практике, затем опять садишься «за парту».

Сейчас у меня идет как раз волна изучения нового, на этот раз я серьезно взялся за Big Data и аналитику. И скоро встрою новые инструменты в свою работу, что сделает мои результаты еще более эффективными.

⋮  
⋮  
⋮ **Выводы. Нельзя однозначно сказать, какой путь лучше, так как любой выбор имеет свои плюсы и минусы. На мой взгляд, эффективнее всего учиться волнами: год изучаешь новое, один-два года используешь новые знания.**  
⋮  
⋮  
⋮

”

Жизнь коротка, поэтому не жди удачного случая, он никогда не наступит.

Действуй сейчас, и другие веб-разработчики уже совсем скоро станут удивляться твоему профессиональному развитию и начнут равняться на тебя.

Филимонов Игорь

IT-эксперт, ведущий преподаватель, автор курсов