# Как расти в IT: почему усердие компенсирует талант



Оглавление

[Как расти в IT: почему усердие компенсирует талант 1](#_Toc96097598)

[Совет #1: разбирайтесь, как работает создаваемый продукт end-to-end 1](#_Toc96097599)

[Совет #2: качайте скиллы, изучайте предметные области, которые нужны для реализации задачи и всё, что происходит на рынке 2](#_Toc96097600)

[Совет #3: чем короче срок выкатки минимального результата, тем лучше 3](#_Toc96097601)

[Совет #4: усердие компенсирует талант 3](#_Toc96097602)

[Совет #5: каждый день задавайте себе вопрос: «За что сегодня я получил зарплату?» 4](#_Toc96097603)

[Совет #6: позвольте себе и коллегам ошибаться 4](#_Toc96097604)

[Совет #7: от ИТ должны гореть глаза 5](#_Toc96097605)

[Используемые ресурсы 5](#_Toc96097606)

## Совет #1: разбирайтесь, как работает создаваемый продукт [end-to-end](https://ru.wikipedia.org/wiki/Сквозное_шифрование)

Развитие узкой специализации — круто. Но нужно постоянно изучать новое. Если сейчас вы чего-то не знаете, то учитесь и делайте, растя скиллы итеративно. Самый ценный специалист — тот, кто может работать полностью самостоятельно, даже если его на годы оставить одного с продуктом.

Желательно разбираться, как работает создаваемый сервис end-to-end: от уровня железа и понимания всего стека выше до продуктового подхода. Важно знать, кому и зачем нужно создаваемое решение, а также как другие игроки на рынке делают схожие продукты.

«Это не моя задача и не моя зона ответственности, у меня нет ресурсов», — по этим словам можно определить посредственного специалиста. Немало проектов в прошлом я делал практически в одиночку. Например, когда только вышел продактом в новое место, и выделять дополнительный ресурс для меня никто не планировал.

Была задача — сделать продукт, и я занимался закупкой и подключением серверов, [деплоем](https://guides.hexlet.io/deploy/), настройкой сети (с помощью коллег-сетевиков) и базовыми настройками. После этого описанием процессов поддержки, подготовкой фактуры для маркетинговых материалов, обучением коллег из продаж и запуском сервиса в продуктив.

Пример в части ИТ. В основе сервиса лежали высокопроизводительные серваки, которые проектировались вендором как однопользовательские. Но нам это не подходило — я сделал многопользовательский режим, обеспечивая мультитенантность. При этом в kvm, например, ещё не было поддержки такого сервака — патчу kvm (и потом патчу каждый апдейт[[1]](#footnote-1) kvm и драйверы вокруг него). Но патчить все время вручную нельзя — делаю автоматизацию обновления без даунтайма[[2]](#footnote-2) и провожу тесты. Если апдейт неуспешен, то должно сохраниться работающее состояние.

Самостоятельное создание мощного решения с нуля и выход за рамки зоны ответственности дало мне опыт, который пригождается до сих пор. Как дальше развивался сервис? Мы продали все его ресурсы за три месяца. На построение следующего решения компания выделила команду, потому что была уверена и во мне, и в проекте.

## Совет #2: качайте скиллы, изучайте предметные области, которые нужны для реализации задачи и всё, что происходит на рынке



У вас могут быть пробелы в знаниях, мало того, они всегда будут с развитием технологий. ИТ постоянно меняется, и многое устареет уже через пять лет. Нужно постоянно изучать лучшие подходы и практики. Пробовать делать самому, имплементировать, ошибаться, делать лучше.

Однажды мы общались с другом после конференции. Сейчас это суперуспешный человек в финансовой индустрии, а раньше был отличным разработчиком. Разговаривали часов пять подряд, — пролетело как пять минут. Среди обсуждаемого я спросил: «Чтобы докачаться до твоего уровня, на каких книжках стоит сфокусироваться?» И перечислил внушительное количество трудов российских и зарубежных авторов.

Он ответил мне: «Слушай, да я давно прочитал все эти книжки, и даже больше». Это ещё раз подтвердило для меня, что успех невозможен без процесса изучения, обобщения и применения лучших практик. Так работает прогресс.

От новых людей в проектах часто слышу: «Ну, мы — не они. У нас нет ресурсов, понимания начальства, опыта» (можно дополнять). Не ставьте себе преграды. Попробуйте ответить на вопрос —  что нужно сделать именно мне, чтобы быть лучше сегодня? Отвечайте на этот вопрос каждый день. Если каждый в своей части будет делать всё от него зависящее, — мы не будем «ими», мы будем лучше. Что мешает прямо сейчас нам быть лучше? Находите и устраняйте барьеры.

Что касается меня, то я просто не думаю о том, что чего-то сделать нельзя. Куда конструктивнее думать о поиске решения. Не только в технологиях и разработках, но и в организационных моментах, например, в возможности платить хорошие зарплаты сотрудникам, в конце концов, в возможности выдать каждому разработчику моего центра лучший ноут на рынке.

## Совет #3: чем короче срок выкатки минимального результата, тем лучше

Лучше сделать максимально просто и быстро собрать обратную связь, чем делать год никому ненужное решение. Такой подход может встречать сопротивление особенно у опытных техлидов. Понятно, что техдолг надо исправлять и рефакторинг рулит, но сперва реализуйте минимальные сценарии.

Работающий продукт — главная ценность. Мой самый провальный проект был в начале карьеры, когда я пилил софтину два года. Когда я её выпустил, она уже не была нужна. Противоположный мой опыт связан с mvp биллинга, который реально работал, и который я сделал за четыре дня.

 Работающий продукт — главная ценность

Если mvp выстреливает, то дальше всегда найдется время сделать лучше. Доставляйте то, что делаете, как можно раньше до бизнеса и потребителей. И улучшайте: собирайте обратную связь и делайте именно то, что нужно вашим пользователям. Важно постоянное итеративное развитие и учёт [фидбека](https://ru.wikipedia.org/wiki/Обратная_связь_(коммуникации)). К сожалению, ваши долгие идеальные решения на пути к релизу в итоге могут оказаться невостребованными.

## Совет #4: усердие компенсирует талант

Я считаю себя абсолютно средним, не выдающимся человеком. Но точно знаю, что привычка решать задачи, сложность которых превышает твои текущие скиллы, позволяет расти. Уверен, что можно хорошо делать работу почти в любой сфере, если очень хочешь и стараешься.

Перейдя в седьмой класс в математическую школу, я попал в новый для себя мир. Если в обычной школе дети просто учились, а в свободное время развлекались, то здесь для меня началась сплошная зона дискомфорта. По сути, мне постоянно «предлагали» перейти на новый уровень, который превышал мои возможности. Занятия проходили даже летом (в формате решения задач из заочной школы МГУ).

Именно тогда я открыл для себя, что можно сильно постараться, потратив, возможно, в десять раз больше сил и времени, чем более умные (в моем понимании) одноклассники, —  и получить классный результат. Это знание использую до сих пор.

## Совет #5: каждый день задавайте себе вопрос: «За что сегодня я получил зарплату?»

Примечание: я не задаю этот вопрос своим коллегам и никогда в компании не слышал такого вопроса. Задаю его только сам себе.

Этот вопрос часто задавал нам генеральный директор (он же собственник бизнеса) небольшой компании, которая жила только на выручку от клиентов, без инвестиций. Он мог уволить любого сотрудника за один день, если не получал ответа. Этот урок я запомнил на всю жизнь. Это мощнейший мотиватор. Вообще, довольно страшно в один день оказаться без работы, особенно, если у тебя семья и двое прекрасных детей.

До сих пор, даже будучи директором большого центра разработки, я каждый день задаю себе этот вопрос. И получаю ответ. Советую делать то же самое, —  не важно, это ваш первый проект как стажёра или вы помогаете строить бизнес, которого до вас не существовало.

## Совет #6: позвольте себе и коллегам ошибаться

Никто не знает всего, и каждый может ошибаться. Нужно принять это и внимательно относиться к коллегам. Советуйтесь, спрашивайте, слушайте и помогайте. Отношения «я начальник — ты дурак» — большое зло. Даже самый неопытный сотрудник может родить крутую идею или решение, которое может изменить отрасль. И дураком будете именно вы, если не будете общаться с командой на равных и решать вопросы консенсусом.

Поработав практически на всех позициях ниже своей текущей, я понимаю, насколько это важно. Горящие глаза очень просто затушить безразличием. Демотивированные коллеги работать хорошо просто не смогут. Не понимаю начальников, которые так делают.

Отношения «я начальник — ты дурак» — большое зло

Для того чтобы команда развивалась, люди должны иметь возможность влиять на проекты, которые делают. В нашей команде каждый может максимально применить свои скиллы, брать больше ответственности на себя.

## Совет #7: от ИТ должны гореть глаза

Для того чтобы расти, нужно работать. Как в спорте: только преодолевая себя, вы становитесь сильнее. Спать нужно, иначе у вас будут проблемы со здоровьем. Но work life balance, кажется, не существует. Либо хобби совпадает с ИТ (и не важно — это работа со сложнейшими системами или переустановка Windows), либо вы выбрали не то занятие.

От ИТ, не смотря на все сложности, должны гореть глаза. Если это не так — у меня нет для вас совета. Вас должны драйвить постоянные изменения и всё новое в ИТ, позволяющее эффективнее решать старые задачи. (Кстати, у нас нового достаточно — сейчас мы с ребятами в #CloudMTS работаем над собственными решениями, которые, как мы думаем, поменяют облачный рынок.)

## Используемые ресурсы

* <https://tproger.ru/articles/kak-rasti-v-it-pochemu-userdie-kompensiruet-talant/>
* <https://guides.hexlet.io/deploy/>
* <https://ru.wikipedia.org/wiki/Сквозное_шифрование>
* <https://ru.wikipedia.org/wiki/Обратная_связь_(коммуникации)>

1. Обновление базы или части базы поисковой системы [↑](#footnote-ref-1)
2. Время, когда система не работает [↑](#footnote-ref-2)