

Светлана А. Савельева
AT Software
s.saveljeva@at-software

Задача как мотивирующий фактор

... один программист кладет перед другим однострочную программу и либо гордо рассказывает, что она делает, а затем спрашивает: "А ты можешь закодировать это меньшим количеством символов?" - как будто это имеет какую-то практическую ценность, - либо просто спрашивает: "Угадай, как это работает!".

Эдсгер В. Дейкстра, «Смиренный программист»

Вступление.

Удержание квалифицированного разработчика ПО в компании при современной ситуации на рынке труда является одной из наиболее актуальных задач в сфере управления персоналом. Если еще несколько лет назад средний срок работы на одном месте для квалифицированного разработчика оставлял порядка 3-5 лет, то сейчас, по оценкам представителей hr-служб, смена места работы раз в 1-2 года считается нормой.

Проблемой удовлетворенности трудом мотивации занимались такие ученые, как Ф. Тейлор, А. Файоль, Э. Мейо, Г. Форд, А. Маслоу, Д. Мак-Грегор, К. Альдерфер, Д. Мак-Клелланд, Ф. Херцберг, Л. Портер, Б.Ф. Скиннер и др. Эти ученые создавали различные теории и модели, выработали практические рекомендации, широко применяющиеся в современной практике управления персоналом.

Изучая вопрос мотивации и удовлетворенности трудом в IT-сфере, написали свои работы такие авторы как Джоэл Спольски, Эдсгер Дейкстра, Эдвард Йордон и ряд других авторов.

Безусловно, профессиональная деятельность разработчика ПО имеет свою специфику с точки зрения факторов, определяющих уровень удовлетворенности трудом и, как следствие, уровень лояльности к работодателю. Если сравнивать так называемые «карты мотиваторов» среднестатистического разработчика ПО и, например, таковые менеджера по продажам, мы получим совершенно разные картины: активность и заинтересованность разработчика и продавца определяют различные факторы.

На практике, анализируя результаты job interview, и в частности, причины, по которым разработчик покидал предыдущие проекты или компании по собственному желанию, можно выделить три основные. Чаще всего ими являются:

- **неудовлетворительное качество менеджмента проекта**
- **неудовлетворительная оплата труда**
- **неинтересная задача**

Порядок с точки зрения приоритетности в каждом отдельном случае может быть различным, но все три причины, в большинстве случаев, так или иначе, озвучиваются в процессе прохождения интервью при найме.

Ниже мы более подробно рассмотрим один из перечисленных факторов, определяющий удовлетворенность трудом разработчика ПО и качество выполняемой работы, а именно характер выполняемой задачи и, как следствие, **отношение разработчика к задаче**. Для получения качественного ПО важно, чтобы разработчик принимал задачу, чтобы она была ему интересна и соответствовала его личностным особенностям.

Небольшое замечание: у нас нет точных статистических данных, но эмпирический опыт показывает, что чем выше квалификация разработчика, тем более значим для него фактор «интересности» задачи при принятии того или иного предложения о найме.

1. Какими могут быть задачи?

1. Если попытаться описать характер существующих задач, то можно получить ту или иную типологизацию. Например, при классификации задач можно выделить следующие критерии:
 - а) Степень самостоятельности при выполнении задачи (от реализации небольшой готовой задачи до самостоятельного создания архитектуры и декомпозиции для других участников команды)

- b) Предметная область в зависимости от функциональной части системы (GUI, базы данных, и пр.)
- c) Коммуникация с клиентом (общается напрямую с заказчиком или нет)
- d) Степень «рутинности» задачи (монотонная однообразная задача или поддержка; Start-up's; исследования или инновации; регулярная смена задач или даже технологий)

Безусловно, это далеко не единственные возможные критерии, на основании которых можно сделать те или иные описания. Но их, на наш взгляд, можно зафиксировать как одни из наиболее «ежедневных», встречающихся в повседневной практике ведения проекта по разработке ПО. На основе подобной классификации могут быть составлены небольшие опросники, которые могут быть использованы для более четкого понимания предпочтения разработчиков в отношении задачи (в качестве примера может быть представлен опросник, данный в приложении 2). Такие методы исследования позволяют дать прогноз на поведение разработчика в проекте в отношении принятия задачи и эффективности ее реализации. Они показывают, какая задача будет для разработчика наиболее «приемлемой» и где его труд будет наиболее эффективен.

Это позволяет более качественно и целенаправленно осуществлять управление человеческими ресурсами при разработке ПО.

2. Отношение разработчика к задаче и практика hr-менеджмента

Если говорить об общих процессах управления человеческими ресурсами компании (human resources), то, на наш взгляд, важно учитывать фактор принятия задачи и уделять отдельное внимание подбору задачи в соответствии с ожиданиями и предпочтениями разработчика. Нужно стараться привести в соответствие «вкусы» разработчика и тип задачи, которую он получает для реализации (хотя далеко не всегда это возможно). Если этого не происходит, у разработчика довольно часто снижается производительность труда, он делает меньше работы, а ее качество становится хуже. На фоне утраты интереса к задаче происходит снижение лояльности к работодателю, и, как следствие, повышается «текучка», что автоматически снижает стоимость компании

На наш взгляд, важно выяснить ожидания и предпочтения разработчика в следующие моменты его работы в компании

- на стадии первичного интервью при найме.

Приглашая нового сотрудника в компанию, нужно четко понимать, под какую задачу он берется и в какой степени эта задача соответствует его ожиданиям.

- На стадии включенности в проект. Регулярный мониторинг на предмет удовлетворенности трудом во время проекта позволяет прогнозировать уход и «играть на опережение»

- На моменте перерыва в работе при переходе в новый проект

Здесь нужно быть особенно внимательным, так как осознание собственного отношения разработчика к задаче в данном случае особенно велико и потребность в соответствии своим ожиданиям наиболее актуализирована. Т.е. это тот момент, когда у разработчика, в силу возникшей паузы, есть возможность подумать не только о задаче, но и о себе.

Реализация подобных практик может быть осуществлена через внутренние документы, фиксирующие процессы службы персонала. Могут быть разработаны и утверждены графики мониторинга, исследования можно привязывать к регулярной аттестации, если таковая существует в компании, необходимость проведения периодического опроса на предмет удовлетворенности задачей может быть зафиксирована в положении о персонале и т.д.

3. Задача и управление ожиданиями клиента.

При аутсорсинговой (сервисной, а не продуктовой) модели бизнеса, проблема качества и характера задачи, над которой будет работать разработчик, встает особенно остро. Задачи, а иногда и состав команды, утверждает непосредственно заказчик, лишая руководителя проекта возможности свободно выбора задачи для того или иного программиста или произвольной ротации.

Соответственно, полноценный процесс управления человеческим капиталом в такого рода бизнесе включает в себя и аспект управления ожиданиями клиента. Клиент должен осознавать, что однообразная и неинтересная задача снижает мотивацию и удовлетворенность трудом у команды. Довольно часто, по нашей практике, со стороны заказчика высказываются опасения относительно любых перестановок внутри команды, а увольнение разработчика по

собственной инициативе рассматривается как настоящая проблема. Но, на наш взгляд, одной из основных задач, лежащих на стыке hr- и клиент-менеджмента, являются выстраивание осознанных ротаций внутри проектной группы и вовлечение в обсуждение этого вопроса клиента. Этот процесс должен быть плановым и управляемым. Перемещения, вводы и выводы людей из команды определяются заблаговременно, обсуждаются и согласовываются с заказчиком. До клиента важно донести, что стратегия простого «силового» удержания разработчика в течение долгого времени на одной и той же задаче, которая данному разработчику не интересна, ведет к снижению мотивации, ухудшению качества создаваемого кода и, в конечном итоге, потере разработчика как участника команды.

Понятие «развития команды» в данном случае включает в себя с одной стороны, плановое изменение в характере задач путем обсуждения таких возможностей с заказчиком, с другой стороны, с плановыми перестановками внутри команды в соответствии с ожидаемыми задачами. Вполне понятно, что «интересных задач» на всех не хватает, и, осуществляя аутсорсинг, заказчик зачастую пытается вынести за пределы своей компании как раз наиболее рутинные и скучные, с точки зрения выполнения, работы. Но, как показывает наш опыт, если клиент настроен на долгосрочное сотрудничество, он готов к обсуждению такого рода вопросов и к конструктивным действиям в этом направлении.

Заключение.

Таким образом, повышение удовлетворенности трудом у developer's с целью сохранения его мотивации удержания в компании требует (помимо прочего):

- *Диагностики и понимания предпочтения разработчика в отношении задачи.*

Важно адекватно и тщательно провести ее еще на моменте найма нового сотрудника в компанию и учитывать его особенности при дальнейшем распределении в проекты. Диагностика может быть осуществлена различными методами, от неформальных бесед общего характера до использования специально разработанных тестовых методик и опросников.

- *Регулярного мониторинга удовлетворенности разработчика задачей.*

Безусловно, сам по себе такого рода мониторинг несет

определенный риск: его проведение с большой вероятностью ведет к осознанию разработчиком неудовлетворенности задачей и актуализацию потребности в смене задачи, а работодатель не всегда имеет возможность предложения интересной задачи, интересных задач на всех не хватает. Поэтому, планируя мониторинг, нужно помнить о возможности возникновения подобной проблемы. Но, не смотря на то, что такого рода диагностика несет определенную опасность, ее полное отсутствие так же может обернуться возникновением сложностей. Т.е. решение о проведении мониторинга либо об отказе от него должно быть комплексным и взвешенным, учитывающим все влияющие факторы и возможные последствия.

- *Регулярной ротации в случае неудовлетворенности задачей*
- *Простроенных (проактивных) отношений с клиентом в смысле обсуждения качества предлагаемых задач, а так же выведения «уставших» разработчиков в другие проекты.*

Приложение 1.

Ниже мы приведем несколько примеров из практики, которые показывают, насколько важно понимание и учет ожиданий разработчика в отношении предложенной ему задачи и как это влияет на ход проекта в целом.

Пример 1.

В одном из проектов при работе с западным заказчиком, надо сказать, весьма чувствительным к выводам людей из проекта, один из участков работы, рассчитанный на одного разработчика, постоянно являлся «камнем преткновения» с точки зрения «текучки». После небольшого срока работы, обычно после 2-3 месяцев, разработчики

отказывались от выполнения данной работы с просьбой перевести их на другую задачу или проект. При внимательном рассмотрении вопроса выяснилось, что предложенная заказчиком архитектура, в рамках которой приходилось работать нашим девелоперам, была весьма специфичной и «противоестественной» с точки зрения большинства современных программистов. Она была разработана заказчиком довольно давно и в силу внутренних причин сохранялась в этой части проекта, хотя в остальных частях системы уже была изменена. Учитывая важность заказчика для компании, был проведен серьезный поиск подходящего человека, который был бы готов работать в таких условиях. Т.е., требовался специалист, имеющий достаточную толерантность к «неудобной» архитектуре, готовый выполнять мелкие четко ограниченные заказчиком задачи, не имея при этом права на самостоятельные технические решения за пределами поставленных задач. По результатам полученного «портрета» был найден соответствующий разработчик. Не смотря на то, что его поиск занял довольно много ресурсов, в конечном итоге, проблема «текучки» на данном участке работ была решена, а отношения с заказчиком сохранены на должном уровне.

Пример 2.

Start-up's

При прохождении exit-interview один из разработчиков, обсуждая причины своего увольнения, в качестве основной причины назвал частое привлечение его к start-up's новых проектов. Менеджер, проводивший интервью, был крайне удивлен, т.к. увольняющийся сотрудник всегда показывал хороший результат и считался вполне успешным разработчиком в «амплуа» такого рода. В процессе разговора стало понятно, что, не смотря на то, что девелопер был успешен в подобной деятельности и у него это хорошо получалось, сам он крайне тяжело переживал такие ситуации. Новые проекты давались ему ценой большого напряжения и переживались как серьезный стресс. Поскольку, как уже было сказано, результат он демонстрировал вполне успешный, ни менеджер проекта, ни служба персонала вовремя не поинтересовались глубинной сутью происходящего. Не исследовались общая удовлетворенность трудом, не обсуждались те или иные предпочтения в отношении задачи а так же в отношении роли в проекте. Присутствие некоторых особенностей личности, и, как следствие, негативное отношение к участию в start-up's, выяснилась только на exit-interview. Разработчик, как сотрудник компании, был потерян.

Пример 3.

Небольшой проект по разработке ПО для западной телекоммуникационной компании отличался на редкость рутинной задачей и крайне высокими требованиями к уровню профессионализма девелоперов со стороны клиента. Требования к квалификации разработчиков были явно завышенными, но, тем не менее, такие разработчики были заказчику предоставлены. С одной стороны, желание клиента иметь как можно более квалифицированный персонал для выполнения своих работ вполне объясним и понятен. С другой стороны, столь явное несоответствие этого высокого уровня довольно простому характеру задачи привело к ряду проблем в проекте. Через полгода от момента начала работ, из проекта, по собственному желанию, вышел первый человек, а через год ситуация начала ухудшаться, т.к. о своем желании уйти заявили еще несколько разработчиков. Все это время, начиная от момента start-up, с клиентом велись разговоры о том, что его требования к квалификации явно завышены и ситуация over qualified рано или поздно даст о себе знать. Но переговоры не привели к какому-либо решению вопроса. В результате удалось достигнуть только договоренности о смене всей команды, к счастью, она не была большой, но при этом новая команда должна была иметь такую же высокую квалификацию, как и предыдущая, не смотря на все ту же простую задачу. Очевидно, что взаимодействие с клиентом в данном случае нуждается в улучшении и дальнейших переговорах.

Пример 4.

В компании в течение длительного времени существовал большой проект. Довольно многочисленная группа разработчиков высокой квалификации работала практически в одном составе на одного заказчика в течение нескольких лет. Задачи всегда были однородными и похожими одна на другую, менеджеры со стороны заказчика не менялись. Когда по объективным причинам проект был закрыт, компания, рассчитывая на освобождение серьезного трудового ресурса, столкнулась со следующей проблемой: за время долгой работы над одним проектом, разработчики, не смотря на свою высокую квалификацию, потеряли навык адаптации и вхождения в новый проект. Первые попытки запуска проекта этой группой оказались неудачными. Отсутствовал навык start-up'a. Пришлось разделять группу на отдельные части или выводить отдельных людей и включать их в существующие проекты. Таким образом, выполнение долгой однородной задачи и отсутствие действий по ротации разработчиков привело к общей своеобразной деформации квалификации (фактически, к снижению) и невозможности дальнейшего использования всей группы как единой команды.

Приложение 2.

Ниже находится опросник, который можно использовать для исследования предпочтений в области задачи в повседневной менеджерской практике. На наш взгляд, рабочий инструмент для исследования предпочтений разработчика должен быть крайне прост в заполнении и однозначен в интерпретации. Сложные развернутые опросники не будут, на наш взгляд, иметь успеха, т.к. ни технические специалисты, ни менеджеры не мотивированы сами по себе на выполнение больших сложных заданий, цели и смысл которых для них не очевидны.

Среднее время ответа на список вопросов, представленных ниже, составляет порядка 10 мин.

Как правило, мы выдаем подобный опросник на первичном интервью при найме на открытую позицию.

Опросник для выяснения профессиональных предпочтений разработчика ПО.

1. Свое дальнейшее профессиональное развитие Вы видите как менеджер проектов или технический гуру?
2. Какие технологии разработки ПО Вам будут наиболее интересны в ближайшие 2-3 года?
3. Какая предметная область в зависимости от функциональной части системы Вам наиболее интересна?
 - a) GUI
 - b) DB
 - c) Системный уровень
 - d) Что-то еще
4. Как Вы оцениваете наиболее комфортную для Вас степень предоставленной самостоятельности при принятии решения?
 - a) получить небольшие готовые задачи
 - b) реализовывать определенный функционал, самостоятельно прописывая конкретные задачи
 - c) заниматься декомпозицией
 - d) архитектурой системы в целом
5. Коммуникация с клиентом.
 - a) готов общаться с клиентом напрямую по любым вопросам
 - b) готов общаться только в рамках своей узкой задачи
 - c) хотел бы вести все общение только через менеджера проекта или team-lead
 - d) не важно
 - e) что-то еще
6. «Рутинность» деятельности.
Предпочитаю:
 - a) Однообразная стабильная задача
 - b) Поддержка (maintenance)
 - c) Разнообразные задачи
 - d) Start-up's
 - e) исследования и разработки
 - f) не важно
 - g) что-то еще
7. Отношение к «чужому» коду.
 - a) готов работать
 - b) предпочел бы не работать
 - c) что-то еще
8. Какой процент повседневных задач, на Ваш взгляд, должен содержать «вызов интеллект», т.е. требовать от Вас приложения серьезных интеллектуальных усилий для их решения? _____

Спасибо за заполнение анкеты, мы надеемся, что Ваши ответы позволят сделать наше сотрудничество более эффективным.

Список литературы:

1. Дейкстра Э., «Смирный программист», 1972
2. Йордон Э., «Путь камикадзе», Москва, Лори, 2008
3. Рейнвогел Х., «Как пасти коты. Наставление программистам, руководящим другими программистами», СПб, 2006
4. Спольски Дж., «Джоэл о программировании» Символ-плюс, Санкт-Петербург, Москва, 2006
5. Шелдрейк Дж. Теория менеджмента: от тейлоризма до японизации / Пер. с англ. под ред. В.А.Спивака. - СПб: Питер, 2001