
ОСОБЕННОСТИ ОРГАНИЗАЦИИ РАБОТЫ ПЕРСОНАЛА ДЛЯ КОМПАНИЙ СФЕРЫ ИТ

О.В. ПОНОМАРЕВА

ООО «Сибирские информационные системы»

E-mail: ponom-olga@yandex.ru

Рассматриваются особенности организации работников сферы информационных технологий, методы поощрения сотрудников. Представлены способы решения проблем проектов, реализуемых компанией, за счет правильной организации работы персонала.

ВВЕДЕНИЕ

В настоящее время в сфере информационных технологий распространена ситуация, когда руководство компании заботится обо всех своих процессах: технологических процессах производства, бизнес-процессах и т.д., но не уделяет должного внимания процессам по управлению персоналом и кадровым проблемам.

По причине особой специфики производства программных продуктов сотрудники воспринимаются руководством как некоторая система компьютеров или точных информационных систем. Такое отношение к процессу управления персоналом приводит к неудовлетворительной реализации проектов компании. Организация работы сотрудников ИТ-компаний заслуживает особого внимания.

1. ОСНОВНЫЕ ОШИБКИ ПРИ ОРГАНИЗАЦИИ РАБОТЫ И МОТИВАЦИИ ПЕРСОНАЛА

При организации работы и мотивации персонала часто допускаются следующие ошибки:

1. Не учитывается «человеческий фактор». То есть руководство часто забывает, что подробно расписанный график работ является всего лишь примерным и, самое главное, желаемым сценарием проведения работ. На

деле всегда возникают обстоятельства, препятствующие оптимальной реализации проекта, а именно:

а) Рутинная работа, если она не чередуется с творческой, всегда выполняется менее качественно. Поэтому следует чередовать виды работ даже в пределах одного проекта (например, тестирование, подготовку документации и работу в команде с программистами по выяснению причин дефекта и способов его устранения).

б) При распределении должностных обязанностей следует учитывать, что люди, занимающие руководящие должности, должны обладать не только высокой степенью профессиональной подготовки, но и определенными личностными качествами, такими, как умение быстро принимать решение и брать на себя ответственность, стрессоустойчивость, коммуникативность, умение быстро разбираться в новом материале и т.д.

В настоящее же время менеджерами проекта становятся просто лучшие программисты. В какой-то мере это оправдано, так как человек, обладающий отличными профессиональными качествами, может в определенный момент «спасти» проект, заменив собой ведущего разработчика. Но он может и создать проблемы при реализации проекта, не сумев правильно распределить роли между разработчиками, не уделив должного внимания всем аспектам разработки (а не только программированию) и т.д.

2. К каждому сотруднику нужен индивидуальный подход. То есть следует знать сильные стороны каждого сотрудника, способы его мотивации к лучшей работе и использовать их при работе с персоналом.

Можно условно разделить все способы мотивации персонала, применяемые в современных компаниях, на следующие категории:

- а) материальное поощрение (денежное премирование);
- б) повышение в должности;
- в) поощрение льготами (например, премирование дополнительным отпуском);
- г) поощрение грамотами и благодарственными письмами различного рода.

К каждому сотруднику нужен индивидуальный подход, но зачастую выбор сводится к стандартным способам мотивации. Наиболее распространена практика повышения заработной платы и премий всем сотрудникам на одинаковое количество процентов. Эффективнее было бы ввести более сложный коэффициент повышения денежного вознаграждения.

2. РАСЧЕТ ПОВЫШЕНИЯ ЗАРАБОТНОЙ ПЛАТЫ СОТРУДНИКА

Рассмотрим пример расчета денежного вознаграждения сотрудника отдела контроля качества. Сначала введем следующие обозначения:

SP (Salary Pay) – текущая заработная плата сотрудника.

SP_{new} (*Salary Pay new*) – новая заработная плата сотрудника, это и есть искомая величина, которую следует высчитать для сотрудника.

CS (*Company Seniority*) – коэффициент стажа работы в данной компании. Следует учитывать тот факт, что сотрудник, давно работающий в компании, вызывает больше доверия и имеет право на предпочтительное отношение за счет своих заслуг перед компанией в период своей работы. Величина CS рассчитывается по формуле

$$CS = \frac{\text{количество_лет}}{100}. \quad (1)$$

GS (*General Seniority*) – коэффициент общего стажа работы по специальности. Вполне логично, что работник, имеющий большой стаж работы по специальности, является в подавляющем большинстве случаев лучшим специалистом, такому сотруднику можно доверить более ответственную работу. Поэтому наличие у сотрудника большого стажа работы должно быть отмечено со стороны руководства. Величина CS рассчитывается по формуле

$$GS = \frac{\text{количество_лет}}{100}. \quad (2)$$

VPC (*Valuation Personal Coefficient*) – персональный оценочный коэффициент. Данный коэффициент предоставляется каждому сотруднику его непосредственным руководителем и отражает субъективную оценку сотрудника. Такая оценка также необходима, так как кроме фактических оценок (стаж и опыт работы в компании) необходима оценка непосредственного руководства отдела, заинтересованного в скорейшем и наилучшем выполнении задач, поставленных высшим руководством. Данный коэффициент предоставляется по усмотрению руководителя и может принимать значения от 0 до 0,5.

PC (*Project Coefficient*) – коэффициент проекта, в котором работает сотрудник. Данный коэффициент отражает, насколько важен и сложен проект, в котором работает данный сотрудник. Чем важнее и прибыльнее проект, тем больше ответственность сотрудников, задействованных в нем. Данный коэффициент высчитывается по формуле

$$PC = FLC + Pfc,$$

где FLC (*Foreign Language Coefficient*) – коэффициент за работу на иностранном языке. Данный коэффициент является логической величиной и может принимать одно из двух значений:

$FLC = 0$, если проект русскоязычный, так как работа на родном языке не отягощена дополнительными сложностями (как, например, в случае работы на иностранном языке);

$FLC = 0,01$, если проект иностранный.

Коэффициенты достаточно малы (сотые доли единицы), так как, как правило, один сотрудник участвует в нескольких проектах – это распространенная практика. Поэтому, чтобы не раздувать стоимость проекта за счет высоких коэффициентов (и, как следствие, завышенных заработных плат) сотрудников, коэффициенты выбраны подобным образом.

PfC (Profit Coefficient) – коэффициент прибыльности. Данный коэффициент также является составляющей коэффициента проекта и отражает долю проекта в финансовом состоянии компании. Он рассчитывается как доля проекта (или, говоря иначе, часть денег, приносимых данным проектом, разделенная на 100% дохода);

Новая заработная плата сотрудника получается путем суммирования текущей заработной платы и величины, полученной в результате перемножения текущей заработной платы на сумму всех коэффициентов, с учетом формул (1) и (2), а именно:

$$SP_{\text{нов}} = SP + SP * (CS + (GS - CS) + VPC + \sum_{i=1}^n PC_i), \quad (3)$$

где $i = 1 \dots n$ – номер проекта, в котором работает сотрудник, т.е. происходит суммирование коэффициентов всех проектов, в которых участвует сотрудник (один работник, как было сказано выше, может быть вовлечен в несколько проектов).

Все коэффициенты подобраны в соответствии со степенью важности их влияния на коммерческую стоимость проекта.

Теперь рассмотрим формулу расчета повышения заработной платы сотрудника отдела контроля качества на конкретном примере.

Допустим, заработная плата сотрудника в настоящий момент времени составляет 20 000 рублей, стаж работы в компании составляет 1 год, а общий стаж работы по специальности 2 года. Данный сотрудник занят в трех проектах в компании. Два из них – иностранных заказчиков, доля каждого из них в состоянии компании составляет по 1/5 (т.е. прибыль компании на 2/5 состоит из денежных средств, приносимых данными проектами). Доля русскоязычного проекта составляет 1/3 в состоянии компании. Руководитель, несмотря на небольшой стаж сотрудника, оценивает его работу как отличную и ставит высокий персональный оценочный коэффициент, равный 0,5. Таким образом, получается:

$$\begin{array}{ll} SP = 20\,000; & PfC_1 = 0,03; \\ CS = 1/100 = 0,01; & PfC_2 = 0,02; \\ GS = 2/100 = 0,02; & PfC_3 = 0,02; \\ VPC = 0,5; & PC_1 = 0,03; \\ FLC_1 = 0; & PC_2 = 0,03; \\ FLC_2 = 0,01; & PC_3 = 0,03. \\ FLC_3 = 0,01; & \end{array}$$

По формуле (3) имеем:

$$SP_{new} = 20000 + 20000*(0,01 + (0,02 - 0,01) + 0,5 + 0,09) = 23200.$$

Это новая заработная плата сотрудника.

Процесс проверки системы и поиска ошибок представляет собой череду творческой работы по поиску скрытых дефектов и выполнение рутинной работы по осуществлению дымового тестирования (т.е. тестирование только основных функций программного продукта, их число, как правило, не превышает 5 основных) и проверки всех элементов GUI (Graphical User Interface – набор элементов и функций программы, видимых для ее пользователя).

3. ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ПУТЁМ ОРГАНИЗАЦИИ РАБОТЫ ПЕРСОНАЛА

Все проблемы, возникающие в проекте только по причине недостаточно высокого качества работы персонала, можно разделить на три группы:

- 1) срыв сроков сдачи проекта;
- 2) большое количество дефектов в полученном программном продукте;
- 3) несоответствие полученного программного продукта требованиям, заявленным заказчиком в спецификации.

Указанные проблемы обычно решаются путем увеличения штатов сотрудников. Рассмотрим данный способ решения проблем подробнее.

Программные проекты чаще «разваливаются» из-за нехватки календарного времени. Причина такой ситуации в современных IT-компаниях заключается в следующем:

1. Слабо развитая система оценок, т.е. оценка ликвидности проекта неадекватна и редко совпадает с реальным прогнозом, который можно было бы получить, проведя более подробный анализ и оценку возможностей компании.

2. Методы оценки ошибочно путают достигнутый прогресс с затраченными усилиями, т.е. если для решения задачи пришлось приложить большое количество средств и усилий, то она сразу же переходит в разряд сложных и значимых, хотя на самом деле таковой не является, просто для ее решения у компании не было необходимых средств и опыта.

3. Сложности в общении с заказчиком из-за того, что менеджеры проектов не уверены в своих оценках. Не обладая достаточными средствами для проведения оценочных мероприятий, менеджеру проекта трудно быть убедительным и привести серьезные доводы в защиту своего проекта временного и финансового плана.

4. Недостаточный контроль выполнения графика работ. К сожалению, данный аспект полностью попадает в разряд «человеческого фактора» и

в большой степени зависит от личностных качеств менеджера проекта. Кроме того, управление группой разработчиков, занимающихся производством программного обеспечения (ПО), несколько отличается от управления другими категориями работников и имеет ряд своих особенностей, о которых будет сказано ниже.

5. Стремление команды, работающей над проектом, к скорейшей сдаче очередного этапа проекта, не задумываясь о качестве продукта и процессов разработки в целом. Это выражается в том, что когда остается мало времени до сдачи очередного прототипа системы, определенная часть функций программы на данный момент не является стабильной, поэтому все элементы программы, увеличивающие риск провала испытаний, убираются из прототипа. Таким образом, над проектом нависает угроза деградации, и очень часто данная угроза переходит в разряд действительности.

6. Очень часто программный продукт передается заказчику, не являясь продуктом надлежащего качества. Причина этого явления заключается в следующем: неправильное распределение проектного времени повлекло за собой тот факт, что программный продукт был отдан на тестирование в отдел контроля качества не своевременно. Причем нежелательна как слишком поздняя передача версии, так и слишком ранняя.

Лучше всего сравнить различные варианты выбора времени начала тестирования, используя графики зависимости времени от найденных дефектов (рис. 1).

Рассмотрим график 1 на рис. 1. Прототип системы был передан на тестирование, когда еще не все функции программы были реализованы программистами в полном объеме. Так как программисты вынуждены тратить большую часть своего времени на реализацию обязательных функций системы, то у них остается очень мало времени на устранение дефектов, най-

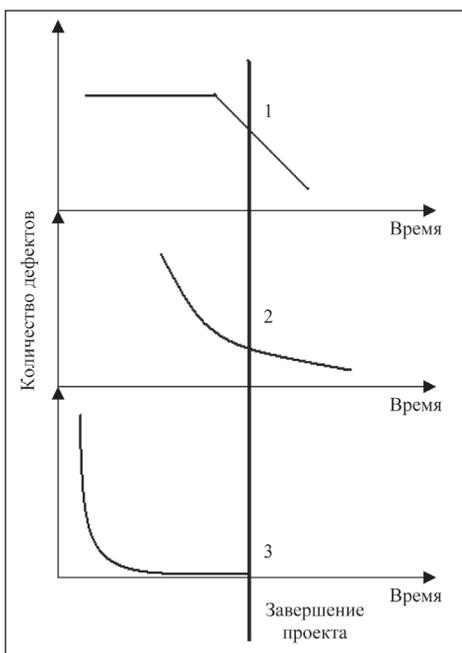


Рис. 1. График зависимости количества найденных дефектов от времени

денных тестерами. Поэтому процесс устранения происходит медленными темпами. Это видно из графика: сначала количество дефектов остается на прежнем уровне достаточно большой период времени, а потом, когда у программистов появляется время для устранения ошибок (когда все основные функции программы реализованы), начинается процесс исправления ошибок, но, к сожалению, тогда остается уже слишком мало времени, поэтому к моменту завершения проекта, т.е. когда необходимо сдавать систему заказчику, программа еще фактически не готова к эксплуатации. Не следует отдавать прототип системы на тестирование слишком рано, так как тестирование «сырой» системы только отнимает время у тестера, поскольку данные ошибки являются скорей недоработками по причине отсутствия времени, о которых программисту и так известно. У программиста, в свою очередь, в этот момент нет времени на исправление найденного тестером большего числа дефектов, так как еще не до конца реализованы основные функции.

Рассмотрим график 2. Самая распространенная ситуация, когда тестирование начинается непосредственно перед сдачей (иногда сроки особенно «экстремальны»), например, тестирование начинается за 1-2 дня до сдачи проекта заказчику с расчетом на то, что программисты все сделали без ошибок), но так как ошибки возникают всегда (невозможно создать идеальную систему с первого раза), то времени на устранение дефектов остается слишком мало, даже несмотря на то, что программисты полностью свободны во времени для исправления ошибок (так как написание обязательных функций они время теперь не тратят). По сути, исполнитель на свой страх и риск отдает заказчику неоттестированную версию продукта.

Рассмотрим график 3. В ситуации, когда у команды, работающей над проектом, есть достаточно времени на разработку прототипа и фаза активного тестирования наступает вовремя – сразу после завершения работ по разработке, и тестеры, и программисты имеют достаточное количество времени для совместного обнаружения и устранения дефектов.

К сожалению, такая идеальная ситуация – очень большая редкость в современных IT-компаниях. Поэтому задача руководства – правильно рассчитать сроки сдачи проекта и распределить силы таким образом, чтобы график сходимости для данного конкретного проекта был максимально приближен к графику 3, изображенному на рис. 1.

При обнаружении отставания от графика или несоответствия получаемой продукции заявленным требованиям общепринятой реакцией является увеличение числа сотрудников. В этом заключается основная ошибка аппарата управления персоналом. Теория о том, что такие величины, как «число месяцев» и «число разработчиков», могут быть взаимозаменяемыми, неверна для современного процесса разработки программных

продуктов. Зачастую задачу нельзя разбить на части, выполняемые параллельно и независимо, и отдать на реализацию разным исполнителям, так как модули системы связаны и реализация одного модуля невозможна до завершения реализации другого модуля. Если задачу нельзя разбить на части, то увеличение затрат не оказывает влияния на график. Большинство задач коммерческого программирования как раз носят такой характер. Даже если и существует возможность условно разбить большую задачу на несколько частей для реализации несколькими сотрудниками, то все равно требуется время на коммуникацию, обмен информацией и согласование действий между сотрудниками, реализующими разные модули будущей системы.

Таким образом, руководство само загоняет себя в тупик, так как увеличение штата сотрудников влечет за собой увеличение временных затрат на коммуникацию между сотрудниками и согласование действий по разработке продукта. К тому же большее количество сотрудников требует больших усилий по управлению как персоналом, так и процессами. Большинство задач, с которыми сталкивается разработчик, могут быть разбиты на части, но требуют обмена данными между подзадачами. Кроме того, каждого нового сотрудника надо обучить технологии, целям проекта, общей стратегии и плану работы. Это обучение нельзя разбить на части, поэтому данная часть затрат изменяется линейно в зависимости от числа занятых. Поскольку создание программного продукта является по сути системным проектом – практикой сложных взаимосвязей, затраты на обмен данными велики и быстро начинают преобладать над сокращением сроков, достигаемым в результате разбиения задачи на более мелкие подзадачи [1]. Таким образом, на определенном этапе количество времени, потраченного на проект, прямо пропорционально количеству разработчиков, занятых в данном проекте.

Более эффективным решением будет следующее. Необходимо создать небольшую специальную группу, состоящую из разработчиков, тестера и аналитика (примерно 1/5 от численности команды, работающей над проектом). Основная группа программистов на ранних этапах разработки занимается созданием нового функционала, тестировщики и инженеры по качеству из основной группы на данном этапе ведут активную работу с аналитиками по разработке системы, а специальная группа в это время занимается тестированием и доработкой уже разработанного функционала в режиме экстремального программирования. Работники, входящие в состав специальной группы, должны быть лучшими специалистами отдела, иметь способность быстро и эффективно решать поставленные задачи.

Если специфика проекта и профессионализм персонала позволяют разделить проект на несколько автономных задач, тогда есть возможность

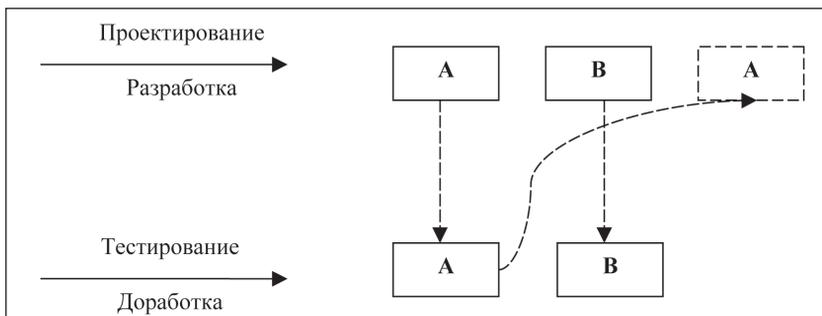


Рис. 2. Организация работы команды

разбить команду на две равнозначные группы и организовать работу по такому принципу: команда А разрабатывает новый функционал, затем приступает к его тестированию и доработке. Когда команда А занимается тестированием и доработкой, команда В разрабатывает свой новый функционал, и, когда команда В приступает к тестированию и доработке своего функционала, команда А снова приступает к разработке нового функционала. Схема такого взаимодействия представлена на рис. 2.

По данной схеме работы требуется усиленный контроль со стороны менеджера проекта, но зато при этом исключается ситуация, когда команда сначала в авральном режиме разрабатывает новую программу, а потом неоттестированный продукт передается заказчику, или разработка нового продукта откладывается по причине того, что команде необходимо время на стабилизацию системы.

ЗАКЛЮЧЕНИЕ

Процесс управления персоналом является столь же важным, как и технологические процессы производства. К тому же это мощный инструмент повышения качества производимой компанией продукции. Организация работников – специалистов сферы ИТ имеет свои особенности, грамотно используя которые, можно значительно повысить эффективность труда.

ЛИТЕРАТУРА

1. Брукс Ф. Мифический человеко-месяц, или Как создаются программные системы. Forbidden Reality, 1995. 171 с.