

С. Архипенков

Руководство командой разработчиков программного обеспечения

Прикладные мысли

Аннотация

Сегодня умение руководить людьми становятся ключом к конкурентному преимуществу. Это искусство решения сложных задач при помощи других людей. Известно, что производительность программистов может отличаться в десятки раз. Задача руководителя команды разработчиков - сделать воспроизводимой высокую продуктивность интеллектуальной деятельности. Эффективный руководитель проекта программной разработки – это в первую очередь, лидер, который, получил от команды признание своей профессиональной компетентности и своих исключительных человеческих качеств. Основываясь на последних наработках в областях управления программными проектами, психологии, соционики, социологии и личном опыте, автор излагает принципы эффективного руководства командами разработчиков программного обеспечения. Рассматриваются вопросы индивидуальной эффективности программистов, эффективных коммуникаций, разрешения конфликтов, лидерства в командах разработчиков, оценки, развития и мотивации участников проекта.

Об авторе

Стаж в разработке ПО более 30 лет. Занимался созданием имитационных моделей сложных космических систем в Центре управления полетами. Руководил коммерческой разработкой ПО и проектами организационного развития в компаниях PriceWaterhouseCoopers, Luxoft, CBOSS. Выполнял проекты по заказу Европейского космического агентства (ESA), Даймлер-Бенц Аэроспейс (Германия), Боинг (США), ЦБ РФ, ОАО «Газпром». Является автором книг, статей и учебных курсов по информационным технологиям и управлению проектами разработки ПО. С автором можно связаться по e-mail: SArkhipenkov@gmail.com

Оглавление

Предисловие	5
Для кого?	5
Что внутри?	5
Благодарности.....	7
Введение	7
В какое время мы работаем?.....	7
Изменение жизненной парадигмы	8
Почему прежние методы управления людьми не работают?	9
Глава 1. Профессиональные психологические особенности разработчиков ПО	10
Специфика разработки программного обеспечения.....	10
Тип личности и темперамент программистов	13
Глава 2. Личная эффективность	22
Ступени роста.....	22
Управляем своей жизнью	23
Эффективный программист.....	26
Глава 3. Эффективное взаимодействие	30
Думать и действовать в духе «выиграл/выиграл»	30
Коммуникации	32
Конфликты	41
Глава 4. Руководство командами	43
Группа и команда	43
Командные роли	46
Этапы формирования команды.....	50
Лидерство и управление.....	53
Роли и стратегии лидера	54
Проблемы неисполнения	58
Глава 5. Практики демотивации.....	60

Классификация антипаттернов руководства.....	60
Антипаттерны некомпетентности.....	61
Антипаттерны мнительности	62
Последствия применения антипаттернов.....	62
Глава 6. Мотивация.....	63
Гуманистическая теория мотивации.....	63
Мотивация и тип личности	65
Мотивация и опыт.....	66
Глава 7. Подбор и развитие команд.....	69
Набор сотрудников	69
Оценка и развитие	73
Сколько надо платить программисту?.....	74
Стандарт People CMM	76
Заключение или Что надо программисту для счастья?.....	77
Литература.....	78

Предисловие

Для кого?

Книга адресована в первую очередь менеджерам проектов и лидерам команд разработчиков программного обеспечения (ПО). Книга будет полезна руководителям отделов и служб ИТ, поскольку позволит им глубже понять особенности разработки программных систем и учесть эту специфику при построении эффективных производственных процессов в подразделении. Думаю, что изложенные мысли могут оказаться полезными в практике HR-служб, для грамотного руководства улучшениями в области подбора, оценки, развития и закрепления наиболее эффективных сотрудников. И, наконец, все остальные участники проектов разработки ПО смогут применить изложенные здесь принципы для повышения личной эффективности: адекватной постановки индивидуальных целей, стратегического планирования личного профессионального и карьерного развития, успешного решения своих задач на основе эффективного взаимодействия с другими участниками команды.

Что внутри?

Программирование только один из видов творческой деятельности. Все изложенное в книге применимо и в других областях коллективного творчества. Правда, у разработки ПО, по сравнению, например, с кинематографом, театром, рекламой или другими отраслями коллективного творчества, есть свои законы и специфические особенности, которые еще мало изучены в силу молодости программной индустрии.

Жизнь сложилась так, что большую ее часть я занимаюсь разработкой программ, а последние 20 лет, в основном, руковожу этим процессом. Все эти годы мне приходилось искать ответы на множество вопросов, начинающихся со слова «почему». Почему сверхвысокого IQ недостаточно для того, чтобы эффективно руководить программистами? Почему только менее 20% проектов разработки ПО завершаются в срок и укладываются в бюджет, а почти треть проектов аннулируется до их завершения [1]? Почему одни программисты могут быть на порядок, а, порой, и на два порядка эффективнее других? Почему сотрудники не выполняют мои поручения? Почему правильно подобранные «лебедь, рак и щука» могут оказаться гораздо эффективнее «родственных душ»? И много других. В результате у меня сложилось собственное представление («карта мира») о правильном подходе при поиске ответов на эти вопросы. Оказалось, что использование лучших языков и технологий программирования, самых совершенных инструментов разработки и систем качества не гарантируют успешность программного проекта. «Именно человеческие качества обеспечивают успех тому или иному проекту, именно они являются фактором первостепенной важности, основываясь на котором надо строить прогнозы о проекте» [2].

Произошел сдвиг парадигмы, заставивший меня увидеть на новой карте мира множество белых пятен, которые пришлось заполнять, используя самообразование и переосмысление приобретенного опыта. Белые пятна находились, в основном, в области гуманитарных знаний: философии, прикладной психологии, социологии и соционики, и не были связаны с информационными технологиями. Полученные знания оформлялись и оттачивались в ходе практической работы, в многочисленных дискуссиях по поводу публикаций, а так же на лекциях и семинарах, в которых мне приходилось отстаивать свои взгляды. Результатами этой работы я и хотел бы поделиться с коллегами по ремеслу. Буду рад, если мои мысли окажутся для кого-то практически полезными.

Адекватная картина мира – правильные ответы на вопросы, начинающиеся со слова «почему», а не со слов «что» и «как», дает нам принципы и критерии выбора верного решения в постоянно меняющихся условиях, вместо попыток записаться рекомендациями на все случаи жизни. Чтобы иллюстрировать, излагаемые мысли, я использовал ряд «историй», которые почти все основаны на реальных наблюдениях.

Первая и главная мысль, которую мне хотелось бы донести до читателя, состоит в том, что творческими командами разработчиков ПО невозможно управлять, их можно только направлять и вести. А для этого не достаточно быть эффективным управленцем, необходимо еще получить признание от команды в качестве лидера. Ответ на вопрос, почему прежние методы управления людьми не работают, дан во Введении.

Следующая моя мысль о том, что все люди разные, требуются терпимость и умение принимать людей такими, какие они есть. Недостатки людей – это, как правило, обратная сторона их достоинств. Следует эти достоинства разглядеть и постараться использовать их с максимальной отдачей для общего дела. В главе 1, на основе последних наработок в типологии Майерс-Бриггс и соционики, обсуждаются профессиональные психологические особенности разработчиков ПО, которые необходимо учитывать при формировании команд, организации их деятельности и их мотивации на достижение общего успеха.

Лидер – это главным образом состоявшаяся личность, поэтому глава 2 посвящена личностному росту и развитию эмоционального интеллекта. В главе 3 рассмотрены вопросы эффективного межличностного взаимодействия и конструктивного разрешения производственных конфликтов на основе доверия и взаимовыгоды, без которых не бывает эффективных команд.

Для российского менталитета командная работа достаточно органичный вид коллективной деятельности. Сельская община, профессиональные артели, рабочие бригады, временные научные творческие коллективы – все это страницы нашей истории. Что такое команда, ее отличие от рабочей группы, динамика ее становления, командные роли участников и стратегии руководителя на разных этапах развития команды – все это обсуждается в главе 4.

Главная проблема успешного командообразования – это создание и сохранение высокой степени мотивации ее участников на общий успех. Но не может быть эффективной мотивации, если руководитель не исключил из своего управленческого арсенала демотивирующие практики. В главе 5 приведен обзор наиболее часто используемых антипаттернов руководства командами, которые приводят к фатальной демотивации исполнителей и делают невозможным создание самоорганизующейся и самоуправляемой команды. Работники приходят в компанию, как правило, не потому, что привержены ее миссии. И не для того, чтобы заработать еще больше денег для владельцев бизнеса. Работая в конкретной компании, участвуя в конкретном проекте, каждый работник стремится к достижению своих индивидуальных целей. «Лучшей заботой о компании будет вовремя сданный проект» (с) анонимный пост на rsdn.ru. Поэтому, в главе 6 рассмотрены вопросы мотивации участников команды на достижение общего успеха в совместной работе, на основе достижения личных целей каждого.

Создание и закрепление эффективной команды – это стратегическое приобретение компании, поэтому последняя глава 7 посвящена вопросам подбора, развития и сохранения эффективных команд.

Мои мысли, надеюсь, принесут вам пользу. Но это произойдет только в том случае, если у вас самих уже возникли аналогичные вопросы «почему». Так как, если вопрос поставлен правильно, то это половина решения проблемы. А если таких вопросов у вас нет, то мои ответы вам, вряд ли, удастся куда-либо приложить.

И еще. Бесполезно пытаться узнать у шахматиста его лучшие ходы. Если вы хотите найти в этой книге набор «приемчиков», освоение которых обеспечит вам эффективность в руководстве командой разработчиков, то разочарую вас – их здесь нет. Все психологические уловки и приемчики легко будут распознаны любым зрелым человеком, а большинство программистов относятся именно к этой категории. Их применение будет воспринято как попытка манипулирования с вашей стороны и навсегда подорвет доверие команды к вам. По словам У.Д. Джордана (цитируется по [3]): «Человек постоянно излучает свою сущность – то, каков он есть, а не то, каким он хочет казаться». Чтобы изменить свою жизнь, стать эффективным руководителем, надо изменить себя изнутри, изменить свое видение мира.

И последнее. Представленная мной картина мира не есть истина в последней инстанции. Вы видите мир по-другому? Это естественно. Все люди разные. И это хорошо! Поскольку сопряжение разных взглядов на один и тот же объект обеспечивает стереоскопичность зрения, позволяет

увидеть в изучаемом объекте дополнительное измерение, а, следовательно, и новые возможности.

Благодарности

Безусловно, все, что написано в книге, отражает мое личное видение проблем программостроения. Многочисленные цитаты приводятся лишь для того, чтобы подчеркнуть тот факт, что высказанные мысли это не только мое личное мнение. Вместе с тем, моя картина мира формировалась не столько на основе изучения трудов авторитетных предшественников, сколько непосредственно путем анализа долгого опыта работы в области информационных технологий. Поэтому, не могу не высказать благодарности, в первую очередь, своим старшим коллегам, работникам ЦУПа, у которых мне посчастливилось учиться. Так же хочу поблагодарить многочисленных братьев по ремеслу, совместно с которыми (на которых?) я набирал свой опыт, решая непростые задачи разработки программных систем. Поименно хочу поблагодарить тех из них, кто непосредственно принял участие в создании данной книги и взял на себя труд найти время, прочесть черновик рукописи и высказать свои замечания и предложения по ее улучшению, которые я с постарался учесть в окончательном варианте. Хочу выразить искреннюю благодарность менеджеру проектов компании EPAM Systems *Владимиру Аркадову*, руководителю направления ИТМиВТ РАН *Владу Балину*, руководителю направления компании «Verysell Проекты» *Игорю Гундареву*, разработчику ПО компании «jNetX» *Кириллу Заборскому*, начальнику отдела компании «Межрегиональный ТранзитТелеком» (МТТ) *Александрю Лебедеву*, менеджеру проектов компании «Intel, Inc.» *Александрю Орлову*.

Особая благодарность членам сообщества RSDN.ru, в первую очередь, господам под никами bkat, Garerton, Spidola, Курилка. Мысли и статьи участников я регулярно и с большим интересом изучаю, творчески переосмысливаю и стараюсь использовать в своей работе.

Введение

В какое время мы работаем?

Скажу сразу, что ответ: «с 9:00 до 18:00» - неправильный. Мы работаем в «век кибернетики и атомной энергии», который еще называется постиндустриальное общество. Отличительные особенности этой новой общественно-экономической формации следующие.

Эпоха перемен. Все в мире стало непрерывно и стремительно изменяться. Изобилие стало причиной острейшей конкуренции. Инновации - неотъемлемый атрибут нашего времени. «Если у вас медленный доступ в Интернет, вы можете навсегда отстать от развития информационных технологий». Практика должна постоянно перестраиваться применительно к новым и новым условиям. Пример. Hewlett-Packard получает большую долю прибыли на товарах, которые год назад даже не существовали [3].

Глобализация. Всеобщая взаимозависимость и взаимосвязанность. Транснациональные компании. Бизнес идет туда, где дешевле рабочая сила. Интернет. Конкуренция без границ. Пример. Google. За ночь любой из нас в принципе может создать многомиллионную компанию у себя в гараже. С помощью Интернета вы можете выйти на рынок, на котором более 100 млн. потребителей.

Все решают таланты. Простая мобилизация средств и усилий уже не может обеспечить прогресс. Вспомним Ф. Брукса [4], «Если проект не укладывается в сроки, то добавление рабочей силы задержит его еще больше». Идею богатства теперь связывают не с деньгами, а с людьми, не с финансовым капиталом, а с «человеческим». Рынок труда превращается в рынок независимых специалистов и его участникам все больше известно о возможных вариантах выбора. Работники интеллектуального труда начинают самостоятельно определять себе цену.

Человечеству известны два вида деятельности. *Репродуктивная* деятельность (труд) является слепком, копией с деятельности другого человека либо копией своей собственной деятельности, освоенной в предшествующем опыте. Такая деятельность, как, например, труд токаря в любом

механическом цеху, или рутинная повседневная деятельность менеджера-управленца на уровне раз и навсегда усвоенных технологий. *Продуктивная* деятельность (творчество) - деятельность, направленная на получение объективно нового или субъективно нового (для данного работника) результата.

Репродуктивная деятельность уходит в прошлое. В постиндустриальном обществе интеллект - основная производственная сила. Сегодня от 70 до 80% всего, что сегодня делается людьми, производится при помощи их интеллекта [5]. В любом товаре, сделанном в США, доля зарплаты составляет 70 процентов (в России, пока, - только 30, очень надеюсь что *лока*). Но это в среднем по всем товарам. Что касается разработки ПО, то почти все, что в этой отрасли производится, создается при помощи интеллекта.

Для того, чтобы выжить в условиях постоянных изменений, глобальной взаимосвязанности и взаимозависимости, необходимости постоянных инноваций, нужны: проекты, команды и лидеры.

Проекты. Небольшой быстрый и конкретный шаг, а вернее скачок в направлении бизнес-цели, реализующий стратегию. Сделать то, до чего другие компании еще не додумались, сделать это как можно быстрее, иначе это сделают другие. Предложить потребителю более качественный продукт или такой продукт, потребность в котором потребитель даже не может пока осознать. Анализ достигнутых результатов, уточнение целей и стратегий - обратная связь.

Команды. Ясность общих ценностей и целей, самоорганизация и самоуправление совместной деятельностью, взаимный контроль, взаимопомощь и взаимозаменяемость, коллективная ответственность за результаты труда, всемерное развитие и использование индивидуального и группового потенциалов. Лучшая аналогия – пожарная команда, которая, как и разработчики ПО, работает в условиях неполной и неточной информации, постоянных изменений, сжатых сроков, ограниченных ресурсов и общей взаимозависимости. «*Коллектив* пожарных приступил к тушению пожара» - или оксюморон, или трагедия.

Лидеры. Современный руководитель – это целеустремленный организатор, у которого есть вдохновение, который создает силовые поля, притягивающие таланты, а не просто служащих, стремящихся занять рабочие места. Просьба не путать таланты со «звездами». Работа лидера не нанимать, а зажигать звезды. Никому уже не нужны работники, которые просто умеют хорошо работать, эффективные руководители ищут тех, кто не умеет работать плохо. Никто больше не верит в руководителей, которые всегда правы и притворяются, что знают больше, чем подчиненные. «Людьми не надо «управлять». Задача – направлять людей. Цель – сделать максимально производительными специфические навыки и знания каждого отдельного работника» [6].

Изменение жизненной парадигмы

Одновременно со сменой общественно-экономической формации происходит изменение жизненной парадигмы человека. В индустриальном обществе человек сначала учился, чтобы хорошо работать, а потом старательно работал, чтобы счастливо жить. В постиндустриальном обществе нет больше разницы между учебой, работой и жизнью. Это один неразрывный процесс. В центре жизнь, а не карьера, учеба или семья. Делай то, что ты любишь! Или, в крайнем случае, полюби то, что делаешь. *Делай то, к чему у тебя есть талант, что у тебя получается лучше других, и докажи, что это нужно всем.*

Сегодня наши знания устаревают быстрее, чем мы успеваем отпраздновать окончание учебного заведения. Обучение – это процесс, который длится теперь всю жизнь. Жизнь - это вереница проектов и великих начинаний. Чем эффективней этот процесс, тем более счастлив человек. Абрахамсон Маслоу один из основоположников гуманистической психологии утверждает [7], что «каждый из нас имеет импульс к самосовершенствованию, к более полному воплощению наших возможностей в действительность, к самоактуализации, или к полной человечности, или к самоосуществлению». Более того, он уверен в том, что сегодня только самоактуализирующиеся люди могут считаться полностью психологически здоровыми. Эйнштейн как-то заметил, что тот, кто ощущает свою жизнь, лишенной смысла, не только несчастлив, но и вряд ли жизнеспособен.

Постиндустриальное общество – это в первую очередь эпоха возможностей. Сегодня поступки и жизнь человека определяют не внешние обстоятельства, как учили психологи-бихевиористы, и не биологические потребности, из которых психоаналитики основное внимание уделяли сексуальной потребности. Жизнь современного человека управляется «экзистенциальными потребностями» [8]. Стремление к поиску и реализации человеком смысла своей жизни В. Франкл считает врожденной мотивационной тенденцией, присущей всем людям и являющейся основным двигателем развития личности. «Не человек ставит вопрос о смысле своей жизни - жизнь ставит этот вопрос перед ним, и человеку приходится ежедневно и ежечасно отвечать на него - не словами, а действиями. Смысл не субъективен, человек не изобретает его, а находит в мире, в объективной действительности, именно поэтому он выступает для человека как императив, требующий своей реализации!»

«Быть», а не «обладать», становится сегодня главной жизненной целью. Творческое бытие – это собственное созидание ценностей своей жизни; это гибкое единство, последовательность, цельность и открытость в отношениях с собой и другими; это жизнетворчество и духовное меценатство. Творческое бытие требует мужества, «смелости быть», а так же воли для преодоления трудностей и своих слабостей.

Почему прежние методы управления людьми не работают?

Прежние методы управления происходят из нашего индустриального и аграрного прошлого, где один день был очень похож на другой. Эффективность зависела от согласованной работы масс. Работа не требовала большой изобретательности. «Тотальный контроль», «нормирование», «пряник и кнут», «человеческий ресурс – «винтик», который легко заменить», - вот главные принципы эффективного менеджмента предыдущих эпох. Чем больше взмахов веслами в единицу времени, тем быстрее идет галера, тем эффективнее работают гребцы!

А. Маслоу [7] утверждает, что «творческий человек – это особый тип человека, а не обычный человек, приобретший новые дополнительные возможности, некое новое умение вроде умения кататься на коньках, овладевший чем-то, что осталось внешним по отношению к его базовой природе». В программировании мало повторяющихся задач, нет места для репродуктивной деятельности. После третьего повторения однотипного действия программист пишет утилиту, которая это действие автоматизирует раз и навсегда. Никто не знает, каким местом программист думает и как он этим местом это делает. Интеллектуальное творчество нельзя нормировать и контролировать. Бессмысленно сажать за спиной программиста нормировщика-контролера с секундомером. Что он увидит и измерит? *Все, кто пытается примерить методы управления фаст-фудом к разработке ПО, обречены на неудачу.* Что? В вашей компании эти методы работают? Сочувствую. Это означает, что вы наняли не тех, кого следует.

Современное предприятие обязано относиться к своим работникам так же, как к своим лучшим клиентам. Главный капитал современной компании – это знания. Большая часть этих знаний неотъемлема от их носителя – человека. Те предприятия, которые этого не поняли этого, не выживут потому, что не смогут быть эффективными. Сегодня эффективное предприятие – это сервис. Предприятие, с одной стороны, предоставляет услуги и продукты своим клиентам, а с другой, - рабочие места для наемного персонала. Принципы «Одно предприятие на всю жизнь», «Работай продуктивно, а предприятие о тебе позаботится» - быстро уходят в прошлое. Посмотрите на рынок рабочей силы в ИТ - правила устанавливают профессионалы. Мало кого интересует, в каких компаниях вы работали, зато всех интересует, в каких проектах вы участвовали и ваш вклад в их успех.

Цель предприятия, которое стремится к эффективности, сделать счастливыми не только своих клиентов, но и своих работников. У проекта разработки ПО сегодня не три, а четыре фактора успеха:

1. Выполнен в соответствие со спецификациями.
2. Выполнен в срок.
3. Выполнен в пределах бюджета.

4. Каждый участник команды уходил с работы в 18:00 с чувством успеха.

Этот четвертый фактор успеха должен стать воспроизводимым, если предприятие хочет быть эффективным. Для успешного проекта характерно постоянное ощущение его участниками чувства удовлетворения и гордости за результаты своей работы, чувства оптимизма. Нет ничего более губительного для проекта, чем равнодушие или уныние его участников.

Эффективность это отношение полученного результата к произведенным затратам. Нельзя рассматривать эффективность, исходя только из результативности: чем больше ты производишь, чем больше делаешь, тем выше твоя эффективность. Так можно «зарезать на ужин курицу, несущую золотые яйца». Затраты не следует путать с инвестициями. Оплата аренды, электроэнергии, коммунальные платежи – затраты. Создание и закрепление эффективной команды – это стратегическое приобретение компании. Обучение участников проекта – инвестиции. Вложение в людей – это увеличение числителя в формуле эффективности. Уход из компании всех профессионалов после проекта, выполненного по принципу «любой ценой» – затраты, причем очень тяжело восполняемые. Нарастающая конкуренция указывает на совершенно четкий тренд в мировой экономике – персонал – это форма инвестиций, активов, которые нужно уметь наращивать, управлять и сохранять. Сегодня люди – это капитал.

В индустриальную эпоху производительность ручного труда увеличилась более чем в 50 раз. Задача менеджеров XXI века – сделать нормой высокую эффективность интеллектуальной деятельности. Признанный эксперт в области управления проектами разработки ПО Том Демарко в своем интервью [9] сказал: «Управление благородно по своей сути. Менеджмент ничего не имеет общего с деятельностью бюрократа, сидящего наверху. Менеджер делает самую необходимую в компании работу. Благодаря ему, становится возможным выполнение самых грандиозных проектов. Нет более почетной работы, чем та, которую делает настоящий менеджер».

Глава 1. Профессиональные психологические особенности разработчиков ПО

Специфика разработки программного обеспечения

Ф. Брукс более 30-ти лет назад написал [4]: «Программист, подобно поэту, работает почти непосредственно с чистой мыслью. Он строит свои замки в воздухе и из воздуха, творя силой воображения. Трудно найти другой материал, используемый в творчестве, который столь же гибок, прост для шлифовки или переработки и доступен для воплощения грандиозных замыслов». Некоторые психологи, которые работают с программистами, идут дальше и даже утверждают, что программирование – это высшая форма творчества [10].

Творчество в программировании начинается с определения целей программы и заканчивается только тогда, когда в ее коде, написанном на каком-либо языке программирования, поставлена последняя точка с запятой. Попытки разделять программистов на творческую элиту, архитекторов и проектировщиков, и нетворческих программистов-кодеров не имеют под собой объективных оснований. Даже если алгоритм программы строго определен математически, два разных программиста его закодируют по-разному, и полученная программа будет иметь разные потребительские качества.

Программирование было, есть и останется в обозримом будущем творческой деятельностью. Хочу отметить, что это утверждение вызывает, как правило, наибольшие дискуссии. Порой меня упрекают в том, что я «выделяю программирование, как вид супер-творческой деятельности, что негативно отражается не только на эффективности команд, фирм, компаний, но и на всей отрасли ИТ в целом и смежных отраслях». В защиту своей позиции могу сказать, что творчеству есть место и в приготовлении домашнего обеда. Все зависит от задачи, которую мы решаем. Творчество – это получение нового результата. Какова доля новых результатов и творчества в проектах разработки ПО? У меня таких данных нет. Есть личное ощущение, что очень большая. Косвенно об этом свидетельствуют высокая неопределенность при разработке программных систем, разброс производительности программистов на порядок и доля провальных программных проектов более 80%. А это уже объективные данные. Существующее состояние Software

Engineering напоминает мне большую поваренную книгу с многочисленными описаниями рецептов однажды успешно приготовленных блюд из ингредиентов, которых у меня никогда в будущем не будет. Завтра в моей новой системе будут другие вычислительные машины, технологии, языки программирования, инструменты и базовое ПО, новые проблемы взаимодействия с которыми мне обязательно придется решать.

Творчество - это интеллектуальная деятельность человека, законы которой нам не известны. Если бы мы знали законы творчества, то и картины, и стихи, и музыку, и программы уже давно бы создавали компьютеры. Творческое начало это то, что роднит программирование с искусством и наукой.

Программирование это не искусство, в том смысле, что оно не является «творческим отражением и воспроизведением действительности в художественных образах». Об искусстве в программировании можно и должно говорить только в смысле умения, мастерства, знания дела, как и в любой другой профессии. И как в любой другой профессии программистское мастерство может доставлять истинное эстетическое наслаждение, но только для людей, причастных к этой профессии.

Программирование это не наука. *В программировании нет системы знаний о закономерностях создания программ.* Нарботки математиков в области логики, теории информации, численных методов, реляционной алгебры, теории графов и некоторых других дисциплинах на долю процента не покрывают многообразие программистских задач. Даже выдающиеся программисты не возьмут на себя смелость утверждать об архитектуре новой программной системы то, что она будет успешной. Хотя в программировании уже накоплен определенный опыт провалов, который может позволить искушенному программисту увидеть в архитектуре новой системы антипаттерны - источники серьезных будущих проблем. Но не более того.

Программирование - не искусство и не наука, это ремесло - «рукодельное мастерство, ручной труд, работа и умение, коим добывают хлеб», (с) «Толковый словарь живого великорусского языка В. Даля». Сегодня мы так же далеки от индустриальной разработки программ, как и 50 лет назад. Представляется, что до выхода из «детского» периода программированию еще очень далеко. Пока, на мой взгляд, сложность и масштаб программистских задач растет быстрее, чем взрослеет программное производство. А поскольку это ремесло, то человек, научившийся писать программы на C ++, будет также далек от профессионала, как ученик третьего класса средней школы, научившийся писать по-русски, от А. С. Пушкина или Ф. М. Достоевского. Путь к мастерству в ремесле лежит только через опыт. Нельзя научиться программированию, читая книги. Как нельзя по книгам научиться писать романы, картины, стихи, музыку. А еще программистам нужен постоянный труд самоусовершенствования и саморазвития. Поэтому далеко не все, кто пишет программы, становятся профессионалами. Производительность труда программистов с одинаковым уровнем знаний и опытом в обозримом будущем, по-прежнему, будет отличаться на порядок даже при решении сходных по сложности задач. Хотите, меряйте производительность в строках исходного кода, хотите – в функциональных точках. Творчество неразрывно связано с вдохновением, а это субстанция капризная и непредсказуемая. Помните знаменитый сон Д. И. Менделеева, про периодическую таблицу элементов его имени? Без вдохновения в программировании не обойтись. И, чем сложнее задача, тем труднее извлечь это вдохновение из подсознания. Иногда для этого требуются часы, а иногда недели.

Профессиональное творчество программиста принципиально отличается от творчества в науке и искусстве. Программистские задачи с каждым годом становятся все сложнее и объемнее, а сроки, за которые требуется решить эти задачи, наоборот, с каждым годом сокращаются. Поэтому современные программные системы создаются коллективами от нескольких до тысяч программистов, в то время как творческие деятели науки и искусства работают, в большинстве своем, в одиночку. Проблемы *коллективного* творчества, практически, остались без внимания у психологов и социологов.

Правда, существует еще прикладные науки, гражданское строительство, космическое машиностроение, авиастроение, автомобилестроение и другие высокотехнологичные отрасли промышленности, в которых над созданием новых изделий трудятся сотни тысяч человек. Очень велик соблазн провести аналогию с этими отраслями и говорить об индустриализации подхода к разработке ПО. Не получается.

Упрощенно, путь от идеи до ее реализации в этих отраслях выглядит следующим образом: научно-исследовательская работа (НИР), опытно-конструкторская работа (ОКР), производство. В верхней части этой пирамиды находятся отраслевые НИИ, которые производят идеи и занимаются проектированием новых изделий. На втором этаже пирамиды работают конструкторы в конструкторских бюро, в задачу которых входит реализация нового проекта в чертежах деталей и технологиях изготовления и сборки. На нижнем уровне находятся производственные мощности - заводы, на которых инженеры и рабочие воплощают «в железе» чертежи и технологии.

Если проводить аналогию, то программисты работают исключительно на вершине описанной пирамиды. Программирование – это идеи и проектирование. Роль конструкторского бюро для программного проекта выполняют компилятор и сборщик программ. А программистским аналогом завода, который переводит конструкторскую документацию в продукт, доступный потребителю, служит вычислительный комплекс, на котором разворачивается и выполняется созданная программа.

А теперь давайте вспомним, сколько НИР так и остались на бумаге, не дойдя до ОКР, а сколько еще ОКР закончилось закрытием тематики. Я думаю, что процент инноваций, дошедших до производства, от общего числа исследовательских проектов, выполненных в отраслевых НИИ, будет сравним с процентом успешных программных проектов. И давайте еще учтем, что ученые в НИИ опираются на достаточно хорошо изученные законы математики, физики и химии, а программирование, как мы отмечали выше, пока остается лишь ремесленным производством.

Для коллективного программистского творчества скорее уместна аналогия с созданием художественного фильма, театрального спектакля или исполнения музыкального произведения оркестром. Я полагаю, что количество провальных проектов в этих областях ничуть не меньше, чем в программировании. Дай Бог, если хотя бы пятая часть завершенных фильмов не «ложится на полку» после первого показа. И еще одна аналогия программных проектов с кинематографом. Наличие даже самых звездных актеров не обеспечивает успех фильма. Только талантливый режиссер способен организовать и вдохновить актеров на создание шедевра, открыть новые звезды. А талантливых режиссеров, как, впрочем, и талантливых менеджеров программных проектов, к сожалению, не так много, как хотелось бы.

Поэтому, в обозримом будущем большая часть проектов разработки ПО будет завершаться со срывами сроков и перерасходом бюджета, а часть проектов не будет завершена никогда. Бизнес должен помнить, что инвестиции в разработку ПО по-прежнему будут связаны с высокими рисками. Но эти риски компенсируются тем, что прибыли от одного успешного программного проекта могут порой покрыть убытки от десятков менее удачных.

Есть еще нечто, что отличает труд профессионального программиста от ученого, художника, композитора и поэта. Предметом деятельности ученых являются упрощенные модели, в которых они могут абстрагироваться от большинства деталей реального мира, не существенных для их целей. Математик, доказывая новую теорему о тензорах, не заботится ни о чем, кроме системы постулатов, положенных в основание дифференциальной геометрии. Физик, описывая динамику жидкости в трубе, абстрагируется от того, как движутся и сталкиваются молекулы и от того, как движутся планеты вокруг Солнца. Деятели искусства тоже во многом оперируют абстракциями. Поэту, композитору, художнику достаточно лишь сделать намек, набросок объекта творчества, и на этом его работа закончена. Остальное пусть додумывает читатель, слушатель, зритель.

Программист тоже работает с абстракциями, но ему приходится держать в голове гораздо больше абстракций, чем любому ученому. Абстракции сопутствуют программисту на всех уровнях разработки программы от описания ее целей до исполняемого машинного кода. И этих уровней могут быть десятки. И на каждом уровне абстракций их деталей становится все больше и больше. «Я бы сказал, что отнесение программирования к "инженерным" дисциплинам сослужило ему плохую службу, создав иллюзию "вещности" и "объективности" результатов труда программистов. Тогда как по роду своему программы ближе всего к словам языка (к "Именам"), чем к традиционным объектам материального мира. И к конструкциям над этими словами - песням, сказкам, метафорам. И эта отнесенность касается всех уровней программирования: от системных библиотек до больших прикладных комплексов» [11].

Дополнительно к абстрактному мышлению, программист должен обладать сильно выраженным системным мышлением, чтобы удерживать многочисленные взаимосвязи, существующие на всех уровнях программистских абстракций, а также взаимосвязи между этими уровнями. Еще одной сложностью является то, что все эти абстракции и взаимосвязи между ними изменяются во времени, и программист обязан учитывать эту динамику.

Кроме того, программист должен обладать маниакальной усидчивостью, сосредоточенностью и упорством для перебора всех возможных вариантов поведения своих абстракций и доскональной проработки всех деталей. Проработка должна быть абсолютно точной и не содержать ни одной ошибки, неправильного, лишнего или отсутствующего символа исходного кода (а это сотни тысяч строк, а порой и миллионы). Инструменты программирования: синтаксические анализаторы, компиляторы и проч., - могут лишь незначительно облегчить эту работу.

Еще одна особенность, которая присуща программистскому творчеству, это постоянное обновление информационных технологий, которые программисту необходимо знать и успешно применять в своей работе. Поэтому профессиональный программист должен, как сказал один из наших прежних вождей, «учиться, учиться и учиться». Программист должен удерживать в голове, постоянно пополнять и активно применять на практике гигабайты профессиональной информации. Это устройство компьютеров, компьютерных сетей и сетевые протоколы. Это операционные системы и языки программирования. Это программные интерфейсы промежуточного ПО и прикладных библиотек с особенностями и багами их реализации в конкретных версиях продукта. Это технологические стандарты, технологии разработки и инструменты, которые их поддерживают. Это архитектуры программных систем, паттерны и антипаттерны проектирования и много-много другой информации.

Еще в начале 70-х замечательный советский ученый академик А.П.Ершов сказал: «Программист должен обладать способностью первоклассного математика к абстракции и логическому мышлению в сочетании с эдисоновским талантом сооружать все, что угодно, из нуля и единиц. Он должен сочетать аккуратность бухгалтера с пронизательностью разведчика, фантазию автора детективных романов с трезвой практичностью экономиста». Можно утверждать, что профессиональный программист обязан сочетать в себе легкость и полет таланта Моцарта с усидчивостью и скрупулезностью Сальери.

Тип личности и темперамент программистов

Человек всегда будет субъективно неповторим, но имеет смысл различать в нем устойчивое ядро природных качеств, определяющее основную линию его поведения в рабочей команде. Поведение человека в каждый момент времени определяется множеством факторов: интеллектом, воспитанием, опытом, окружением, ролью, который человек на себя принял, мотивацией и др. Однако ядро – тип личности - обеспечивает относительное постоянство ответных реакций человека на окружающую среду. «Знать характер человека - это знать те существенные для него черты, из которых вытекает, которыми определяется весь образ его действий. Черты характера - это те существенные свойства человека, из которых с определенной логикой и внутренней последовательностью вытекает одна линия поведения, одни поступки и которыми исключаются, как не совместимые с ними, им противоречащие другие» [12].

Тип - абстрактная модель личностных характеристик, присущих определенной совокупности людей. Для определения индивидуальных типов личностей в мире было разработано более 150 моделей. Индикатор типов личности Майерс-Бриггс (Myers-Briggs Type Indicator, MBTI) является наиболее популярным и широко применяемым на протяжении последних 40 лет.

Этот индикатор основан на выявлении двух противоположных способов пополнения запаса энергии и сосредоточения внимания (шкала экстраверсия-интроверсия), двух противоположных способов сбора информации (шкала ощущение-интуиция), двух различных способов принятия решений (шкала мышление-чувствование) и двух различных способов организации своего взаимодействия с миром (решение-восприятие). Таким образом, существуют четыре основных шкалы предрасположенностей, три из которых основаны непосредственно на типологии К.Юнга, четвертая (решение-восприятие) - добавлена на основе исследований И.Майерс [13].

В бывшем Советском Союзе типология К.Юнга получила известность и развитие в восьмидесятые годы благодаря работам Аушры Аугустинавичюте (1983). Она, как бы заново, независимо от западных ученых, открыла его типологические идеи. В результате критического анализа доступной литературы по проблеме типологии А.Аугустинавичюте предложила свою оригинальную модификацию типологии К.Юнга, получившую название «соционика» [14].

Таким образом, у дерева с общим корнем (типология К.Юнга) появилось две ветви, которые строят модель одного и того же явления – тип личности, используя разный подход и разные языки. Сопоставление этих двух направлений в определении и интерпретации типов уведет нас далеко от основного предмета нашего обсуждения. Заинтересовавшийся читатель легко найдет в русскоязычном Интернете массу статей по этому вопросу. Я же постараюсь, привести в этом разделе то, что представилось мне практически полезным (а точнее то, что наилучшим образом описывает мой переосмысленный личный опыт коллективной работы) в обоих этих подходах.

Хотелось бы отметить один символический, по крайней мере, для меня, момент. Первая информация на русском языке по типологии MBTI появилась в переводе книги, посвященной именно психологии программирования [15].

Итак, последователи К.Юнга выделяют следующие четыре характеристики типа личности (Таблица 1)

Таблица 1. Четыре характеристики типа личности

№	MBTI (в переводе [15])	Соционика	Значение
1	E – Extroverted (экстраверты) / I – Introverted (интроверты)	Экстраверты / интроверты	Направление энергии (психической активности)
2	S – Sensing (ощущение) / N – iNtuitive (интуиция)	Конкретное восприятие / интуиция	Сбор информации
3	T – Thinking (мышление) / F – Feeling (чувствование)	Логика / этика	Принятие решений
4	J – Judging (решение) / P – Perceiving (восприятие)	Рациональность / иррациональность	Способ взаимодействия с внешним миром

Следует помнить, что здесь речь идёт не о строгой дифференциации, а о *предпочтениях* [16]. Для сравнения: возьмём левшей и правшей. Если вы правша, это не означает, что вы вообще ничего не делаете левой рукой. Вы просто *предпочитаете* действовать правой. Ваше предпочтение может быть сильным – и тогда вы относительно мало пользуетесь левой рукой; а может быть слабым или вовсе отсутствовать, и тогда вы владеете обеими руками почти одинаково.

В Интернет доступны многочисленные онлайн тесты (например, здесь: <http://www.tests-tests.ru/>), которые позволяют определить тип личности. Наибольшее распространение получили тесты MBTI и Кейрси. Следует заметить, что последователи соционики не признают тестирование в необходимой степени достоверным и определение типа считают возможным только с помощью специалиста.

В реальной жизни, человека, с которым приходится работать вместе, не отправишь на тестирование или на собеседование со специалистом по соционике. Ниже приведена Таблица 2, которая описывает ключевые понятия, наиболее применимые к людям, обладающим той или иной характеристикой типа, и которые могут помочь вам в типизации вашего партнера по взаимодействию.

Таблица 2. Ключевые понятия для разных характеристик типа личности

E – Extroverted (экстраверт)	I – Introverted (интроверт)
взаимодействие внешний облик широта обширный круг знакомых расходование энергии внешние условия общительный сначала говорить, потом думать	чувство территории сосредоточенность внутренние свойства глубина узкий круг знакомых экономия энергии внутренние ощущения задумчивый сначала думать, потом говорить
S – Sensing (конкретное восприятие)	N – iNtuitive (интуиция)
последовательность интерес к настоящему реалистичность упорство действительный приземленный факты практичный конкретный	случайность интерес к будущему отвлеченность вдохновение теоретичный витающий в облаках фантазии оригинальный общий
T – Thinking (логика)	F – Feeling (этика)
объективный верный своим убеждениям законы настойчивость точный ясность	субъективный милосердный обстоятельства убеждение человеческий гармония

критика	понимание
политика	общие ценности
беспристрастный	сочувствующий
J – Judging (рациональность)	P – Perceiving (иррациональность)
решительный	выжидающий
окончательный	предварительный
твердый	податливый
контроль	приспособляемость
завершенность	открытость
структура	изменчивость
точный	экспериментальный
действующий по программе	непредсказуемый
срок	что такое срок?

Особенности поведения человека также свидетельствуют о доминировании той или иной характеристики личности (Таблица 3). Внимательное наблюдение этих особенностей позволяет определять тип личности.

Таблица 3. Особенности поведения человека в зависимости от характеристики типа личности

E – Extroverted (экстраверт)	I – Introverted (интроверт)
Любят разнообразие и активную деятельность.	Любят тишину, позволяющую сосредоточиться на работе.
Нетерпеливы при выполнении долговременных проектов.	Способны работать долгое время над одним проектом, не отрываясь.
Интересуются работой окружающих.	Интересуются фактами и идеями, лежащими в основе работы.
Часто действуют быстро, без предварительных размышлений.	Сначала думают, а потом (может быть) действуют.
Воспринимают телефонные звонки как приятные перерывы в работе.	Воспринимают телефонные звонки как помехи работе.
Генерируют идеи путем обсуждения.	Генерируют идеи путем уединенных размышлений.
Любят работать в окружении людей.	Любят работать в уединении.
Обычно интересуются другими людьми.	
Склонны предпочитать устное общение письменному.	Испытывают затруднения, вспоминая имена и лица людей.

Предпочитают учиться, общаясь с коллегами.	Склонны предпочитать письменное общение устному. Предпочитают учиться на основе чтения литературы, а не обсуждений.
S – Sensing (конкретное восприятие)	N – iNtuitive (интуиция)
<p>Любят решать проблемы, используя прошлый опыт и стандартные подходы.</p> <p>Любят применять уже освоенные навыки и знания.</p> <p>Склонны не доверять своей интуиции и игнорировать ее.</p> <p>Редко ошибаются в фактах. Любят работу с практическим уклоном.</p> <p>Говоря о работе, сначала сообщают конкретные детали.</p> <p>Предпочитают работать установившимся способом, оттачивая свое мастерство.</p> <p>Обычно действуют методично, продвигаясь шаг за шагом к цели.</p> <p>Обращают внимание на специфику каждой ситуации.</p> <p>Склонны упрощать решаемую задачу.</p> <p>Принимают существующую ситуацию как данность, с которой надо работать.</p>	<p>Любят решать новые, сложные проблемы.</p> <p>Больше любят осваивать новые навыки и знания, чем применять их.</p> <p>Склонны следовать своей интуиции, удачно или неудачно.</p> <p>Могут допускать ошибки в фактах.</p> <p>Любят работу с новаторским уклоном.</p> <p>Говоря о работе, сначала представляют ее общий обзор.</p> <p>Любят изменения своей работы, порой весьма радикальные.</p> <p>Действуют по настроению, в соответствии с приливами сил.</p> <p>Обращают внимание на новые возможности и интересные проблемы.</p> <p>Склонны усложнять решаемую задачу.</p> <p>Задаются вопросом о причинах сложившейся ситуации.</p>
T – Thinking (логика)	F – Feeling (этика)
<p>Делают заключения на основе логического анализа.</p> <p>Могут работать при отсутствии гармоничных отношений с окружающими.</p> <p>Могут задевать чувства людей, не подозревая об этом.</p> <p>Склонны принимать решения на безличной основе, иногда уделяя недостаточно внимания желаниям людей.</p> <p>Обычно бывают трезвомыслящими и, если надо, критикуют других.</p> <p>Обращают внимание на закономерности, проявляющиеся в ситуации.</p>	<p>Делают заключения на основе ценностных суждений.</p> <p>Лучше всего работают при гармоничных отношениях с окружающими.</p> <p>Любят делать людям приятное, даже по мелочам.</p> <p>При принятии решений стараются учитывать все симпатии и антипатии, как свои, так и окружающих.</p> <p>Склонны сочувствовать другим и не любят, избегают говорить людям неприятные вещи.</p> <p>Обращают внимание на человеческие ценности, проявляющиеся в ситуации.</p>

<p>Чувствуют себя вознагражденными, когда работа сделана хорошо.</p> <p>Лучше откликаются на идеи окружающих, чем на их чувства.</p> <p>Обладают способностями анализа проблем или ситуаций.</p> <p>Хорошо предвидят логические последствия принятых решений.</p> <p>Хотят справедливой оценки своих заслуг.</p>	<p>Чувствуют себя вознагражденными, когда удовлетворены потребности людей.</p> <p>Лучше откликаются на чувства окружающих, чем на их идеи.</p> <p>Обладают способностью понимания интересов людей в разных ситуациях.</p> <p>Хорошо предвидят влияние принятых решений на людей.</p> <p>Периодически нуждаются в похвале и признании.</p>
J – Judging (рациональность)	P – Perceiving (иррациональность)
<p>Лучше всего работают, когда могут планировать работу и придерживаться планов.</p> <p>Любят приводить в порядок и завершать свои дела.</p> <p>Могут не замечать новые возникающие дела, которые необходимо сделать.</p> <p>Испытывают удовлетворение, приняв решение по поводу ситуации или человека.</p> <p>Быстро принимают окончательное решение</p> <p>Стремятся к упорядоченной структуре и планированию времени.</p> <p>Используют записи при планировании деятельности.</p> <p>Приступая к новой работе, стремятся узнать о ней только существенные моменты.</p> <p>Не любят отрываться от работы ради более срочного дела.</p> <p>Планируют работу так, что каждый шаг выполняется вовремя.</p>	<p>Любят гибкость в работе.</p> <p>Любят оставлять работу незавершенной, чтобы иметь возможность вносить изменения.</p> <p>Могут откладывать неприятные дела, которые необходимо сделать.</p> <p>Любознательны и приветствуют новый взгляд на человека и ситуацию.</p> <p>Откладывают решение. Пытаются найти новые альтернативы и данные, чувствуя, что никогда не будут обладать всей информацией.</p> <p>Склонны начинать сразу несколько дел, но с трудом их завершают.</p> <p>Хорошо приспосабливаются к изменяющейся ситуации и чувствуют себя стесненно без перемен.</p> <p>Используют записи как напоминание обо всем, что предстоит сделать.</p> <p>Приступая к новой работе, стремятся узнать о ней все.</p> <p>Склонны начинать сразу несколько дел, но с трудом их завершают.</p> <p>Оставляют много дел на последний момент, завершая работу вплотную к крайнему сроку.</p>

Различные возможные комбинации четырех базовых характеристик формируют 16 психологических типов. В. Гуленко [14] приводит набор псевдонимов для этих 16 типов, который, на мой взгляд, наиболее практически полезен в деловых взаимодействиях (Таблица 4).

Таблица 4. Псевдонимы 16 типов личности по В.Гуленко

- | | |
|-----------------------|------------------------------|
| 1. ENTP, "Искатель". | 9. ESFP, "Политик". |
| 2. ISFP, "Посредник". | 10. INTP, "Критик". |
| 3. ESFJ, "Энтузиаст". | 11. ENTJ, "Предприниматель". |
| 4. INTJ, "Аналитик". | 12. ISFJ, "Хранитель". |
| 5. ENFJ, "Наставник". | 13. ESTJ, "Управитель". |
| 6. ISTJ, "Инспектор". | 14. INFJ, "Гуманист". |
| 7. ESTP, "Маршал". | 15. ENFP, "Советчик". |
| 8. INFP, "Лирик". | 16. ISTP, "Мастер". |

В том же источнике приводится классификация типов личности по наиболее предпочтительным видам профессиональной деятельности (Таблица 5).

Таблица 5. Классификация типов личности по видам профессиональной деятельности

	Логика (Т)	Этика (F)
Конкретное восприятие (S)	<i>Управленцы:</i> маршал, мастер, управитель, инспектор.	<i>Социалы:</i> политик, посредник, энтузиаст, хранитель.
Интуиция (N)	<i>Ученые:</i> предприниматель, аналитик, искатель, критик.	<i>Гуманитарии:</i> наставник, гуманист, советчик, лирик.

Как уже отмечалось, в мире существует большое число моделей типов личностей, но на мой взгляд, наиболее полезные и распространенные модели хорошо согласуются с типологией Майерс-Бриггс, и служат, как правило, ее упрощением в целях практического применения в тех или иных ситуациях.

По-видимому, одна из первых попыток классификации людей по их типу личности, которая берет начало с трудов Гиппократ (V век до н. э.), была сделана на основе деления по темпераментам. *Темперамент* (от лат. temperamentum - надлежащее соотношение частей), характеристика индивида со стороны динамических особенностей его психической деятельности, то есть темпа, ритма, интенсивности отдельных психических процессов и состояний. Темперамент определяется наследственностью и устойчиво проявляется в течение всей жизни человека. Описанные выше характеристики группируются в четыре темперамента:

- **Холерик.** Имеет самые скоростные темпоритмы. Он много и быстро говорит, без промедления отвечает собеседнику. Часто перебивает: когда собеседник только начал о чем-то говорить, холерик уже все понял и имеет готовый ответ.
- **Флегматик.** Спокойный, миролюбивый и сдержанный человек. У него мягкие и неторопливые движения, негромкий голос. Он никогда не перебивает собеседника, умеет внимательно выслушать и кивает в знак согласия.
- **Сангвиник.** Человек сильный, энергичный, с хорошим самоконтролем. Как правило, он деловитый, выносливый и работоспособный. Нередко — трудоголик, любит хорошо зарабатывать и делать карьеру.

- Меланхолик. Чувствительный, обидчивый и очень ранимый. Легко расстраивается даже при мелких неудачах. Любит жаловаться на судьбу. Искренне верит, что самая «тяжелая доля» и «самые тяжкие испытания» из всех возможных на Земле выпали именно ему.

Из любопытного, Александр Дюма, видимо, сам того не зная, интуитивно вывел в своих произведениях четырех мушкетеров, ярких представителей этих четырех указанных типов — д'Артаньяна (холерик). Артоса (флегматик), Портоса (сангвиник) и Арамиса (меланхолик).

В. Гуленко в своей работе [14] дает однозначную интерпретацию темпераментов в терминах соционических типов. Шкалы «рациональность — иррациональность» и «экстраверсия — интроверсия» образуют в результате пересечения четыре группы соционических темпераментов (Таблица 6)

Таблица 6. Темпераменты в терминах соционических типов

	Экстраверсия	Интроверсия
Рациональность	Холерики	Флегматики
Иррациональность	Сангвиники	Меланхолики

Следовательно, 16 социотипов распределяются по темпераментам следующим образом:

1. Холерики, или рациональные экстраверты: энтузиаст, наставник, предприниматель и управитель.
2. Флегматики, или рациональные интроверты: аналитик, инспектор, хранитель и гуманист.
3. Сангвиники, или иррациональные экстраверты: искатель, маршал, политик и советчик.
4. Меланхолики, или иррациональные интроверты: посредник, лирик, критик и мастер.

Специфика программистской работы обуславливает общность психологических черт большинства профессиональных разработчиков ПО. Достоверных данных статистических исследований найти не удалось, однако, по мнению С.Макконелла [17], которое очень хорошо подтверждается моим личным опытом общения с коллегами, большинство разработчиков ПО принадлежит к двум типам:

«Инспектор» - ISTJ = I (интроверт) + S (конкретное восприятие) + T(логик) + J (рациональный).

«Аналитик» - INTJ = I (интроверт) + N (опирающийся на интуицию) + T(логик) + J (рациональный).

Причем характеристики S(конкретное восприятие) и N(интуиция) среди программистов делятся примерно поровну.

К подтипу S (ощущения) относятся разработчики, живущие в фактическом мире достижимого сегодня. Они конкретны, точны и практичны. Стремятся к специализации. Из них получаются успешные управленцы и настойчивые в реализации программисты. Устанавливают порядок и работают в рамках системы, организуя выполнение задач и завершая их в срок и в рамках бюджета. Люди типа «Инспектор»

- Любят основательность и детальность, справедливость, практичность. Спокойны, серьезны, настойчивы.
- Характеризуются решительностью в критических ситуациях, являются хранителями традиций.
- Просты и чужды манерности. Они трудолюбивы и упорны в работе.

- Стремятся охватывать все подробности и оперировать точными фактами.
- Могут выполнять и разбираться в сложных и многогранных задачах.
- Не выносят небрежного отношения к ресурсам, все необходимое должно быть в нужное время в нужном месте.

К подтипу N (интуиция) относятся разработчики, рассматривающие широкий спектр возможностей, абстрагирующиеся от технических деталей, склонные обобщать и теоретизировать. Из таких разработчиков, на мой взгляд, получаются успешные системные аналитики и архитекторы. Люди типа «аналитик»:

- Наиболее самоуверенные люди, безгранично верящие в себя и свои силы. Живут своим внутренним миром, концентрируясь на возможностях.
- Интересуются будущим больше, чем прошлым. Это люди, умеющие применить на практике теоретические модели.
- Абсолютно не признают авторитетов, базирующихся на положении, звании или прошлых заслугах. В оценке достоинств какой-либо теории полагаются только на свое собственное мнение, невзирая на то, кто отстаивает сходную позицию.
- Действуют в жизни, как бы участвуя в игре на гигантской шахматной доске, изобретая все новые и новые стратегические и тактические ходы.
- Отстаивают право думать по-своему. Это может и помочь и повредить их карьере так же, как и их способность игнорировать чужие мнения и желания.
- Склонны хвалить за достигнутое и избегают обсуждать негативные черты чьего-либо характера. Они предпочитают движение вперед копанию в недостатках.

Согласно [14], большинство разработчиков ПО как рациональных интровертов, следует относить к флегматикам. Это означает, что программисты люди спокойные как внешне, так и внутренне. Они настойчивые и упорные труженики, но им требуется время для раскочки, для сосредоточения внимания, для переключения внимания на другой объект. Интровертная рациональность делает их сдержанными и закрытыми, не отвлекающимися на внешние раздражающие факторы. Это вовсе не означает, что программист не способен вспылить. Просто, чтобы довести его до такого состояния, надо потратить много сил. Это самый уравновешенный из всех темпераментов. Видимо, люди с другими темпераментами в условиях постоянной неопределенности и изменений, жесткого давления сроков и заказчиков, выживают не так успешно.

Типы «инспектор» и «аналитик», безусловно, не единственные, психотипы которые востребованы в разработке ПО. В каждом проекте разработки программного продукта много других работ: бизнес-анализ, проектирование эргономики, графический дизайн, разработка пользовательской документации, наконец, руководство командой. Эти работы с программированием не имеют ничего общего. Для них требуются совершенно другая квалификация и другой склад мышления. При кустарном производстве программ эти задачи, как правило, поручаются программистам, которые это делать не умеют и не любят. Получается обычно плохо, да еще и дорого. В силу своей интроверсии, граничащей с аутизмом, программист просто не в состоянии увидеть свою программу чужими глазами – глазами пользователей. Никто уже не хочет работать с программами с технологической парадигмой навороченного пользовательского интерфейса - кустарным творением программистов - когда для того чтобы работать с системой, надо обязательно знать, как она устроена. Это типичное творение аутиста, которому гораздо важнее видеть, как работает его программа, чем разбираться в том, что она делает для пользователя.

Программисты любят и умеют программировать. Пусть они этим и занимаются. Не загружайте программистов несвойственной для них работой. Включайте в проектную команду, бизнес-аналитиков, эргономистов, художников-дизайнеров, документалистов. Разделение труда - залог перехода от кустарного производства к более эффективному промышленному производству.

Ступени роста

Лидерство, в первую очередь, - это эффективно управлять своей собственной жизнью и только потом другими людьми.

Стивен Кови [3] выделяет три ступени развития личности:

1. *Зависимость*. Человек рождается не эффективным и полностью зависимым от окружения. На оси зрелости зависимость выражена *ты-парадигмой* – ты обо мне заботишься; ты справляешься с чем-то ради меня; ты не справился; в неудаче я обвиняю тебя.
2. *Независимость* выражена *я-парадигмой* – я могу это сделать; я несу ответственность; я полагаюсь на самого себя; я могу выбирать Личное лидерство, независимость – это важнейшие достижения человека в жизни. Некоторые философские учения видят в независимости высшую точку развития человека.
3. *Взаимозависимость* выражена *мы-парадигмой* – мы можем это сделать; мы можем взаимодействовать; мы можем, объединив наши способности и возможности, создать вместе что-то более значительное. Синергия.

Лидерские способности человека на 80% определяются коэффициентом эмоционального интеллекта EQ (Emotional Intelligence) [18]. Вот три его компонента:

- Самосознание. Понять свои собственные чувства.
- Самоконтроль. Научиться управлять своими чувствами.
- Эмпатия. Умение увидеть мир глазами другого человека. Способность к сопереживанию и взаимопомощи.

Эмоциональный интеллект начинается с понимания своих эмоций. Только осознав свои чувства и научившись управлять ими, мы можем научиться понимать чувства других людей и, что еще важнее, понимать причины, которые их вызвали. Люди с высоким значением коэффициента эмоционального интеллекта хорошо владеют своими чувствами, сохраняют самообладание в любой обстановке. Они менее подвержены стрессу. У них хорошо развито чувство юмора по отношению к себе и к ситуации. Владеть чувствами совсем не означает подавлять их, наоборот люди с высоким эмоциональным интеллектом честны и открыты. Развитый EQ позволяет им адекватно оценивать и контролировать ситуацию, быстро адаптироваться к изменениям. Без развитого эмоционального интеллекта невозможно понять других людей и, следовательно, понять, как их можно мотивировать на достижение успеха в общем деле.

Серьезное изучение EQ началось в 70-х годах прошлого века. Психологи пытались определить, что делает людей профессионально успешными. Они изучали работу различных организаций и их служащих, тестировали их навыки межличностного общения — то, как они относились к другим, учитывали их IQ и технические навыки. В исследованиях обнаружилось, что умение контактировать с людьми не менее важно для личного успеха и успеха организации, чем профессиональная компетентность. Если у вас недостаточный EQ, не отчаивайтесь. В отличие от IQ, который формируется в ранней молодости, а затем практически не меняется, EQ можно повышать на протяжении всей жизни.

Что же позволяет человеку эффективно управлять своей собственной жизнью? Выдающийся психолог Виктор Франкл [8] ввел в оборот термин «проактивное поведение» в противоположность «реактивному поведению». Реактивное поведение - это когда выбор определяется внешними обстоятельствами, окружающими условиями и стимуляцией. Например, собака Павлова: раздражитель – реакция. Реактивные люди движимы чувствами, событиями, условиями и своим окружением. Проактивное поведение означает, что человек осознал свои глубинные ценности и

цели и действует в соответствии со своими жизненными принципами, независимо от условий и обстоятельств. Человек сам определяет свою программу. Во времена СССР про таких людей писали в производственных характеристиках, что они «занимают активную жизненную позицию».

Проактивный человек принимает ответственность за себя и свою жизнь, а не ищет причины для обоснования происходящих с ним событий в окружающих людях и обстоятельствах. Он подчиняет свою импульсивную реакцию своим тщательно продуманным, отобранным и принятым ценностям, проявляет инициативу, поскольку сознает свою ответственность за то, чтобы события вокруг развивались определенным образом, берет на себя риск ошибиться. Проактивный подход к ошибке заключается в ее быстром признании, исправлении и извлечении необходимого урока. Однако если ошибку не признать, не исправить и не извлечь из нее урока, то она станет ошибкой уже второго порядка. В этом случае человек обычно встает на путь самообмана и самооправдания, логически обосновывая случившееся для самого себя и для других. Это укрывательство – вторая ошибка, которая, усиливая первую, придает ей непропорционально большой вес и наносит огромный ущерб личности.

Человек отличается от животных тем, что обладает только ему присущими качествами: самосознание, воображение, совесть и воля.

Самосознание означает осознание, оценка человеком своего знания, нравственного облика и интересов, идеалов и мотивов поведения, целостная оценка самого себя как деятеля, как чувствующего и мыслящего существа. В самосознании человек выделяет себя из всего окружающего мира, определяет своё место в круговороте природных и общественных событий. Самосознание оказывает влияние не только на наши установки и поведение, но также и на то, как мы видим других людей.

Воображение – это построение образа средств и конечного результата предметной деятельности субъекта; создание программы поведения, когда проблемная ситуация неопределенна. Важнейшее значение воображения в том, что оно позволяет представить результат труда до его начала, тем самым ориентируя человека в процессе деятельности. Все, что мы производим, создается дважды. Существует мысленное – или первое – творение и физическое – или второе творение.

Совесть – это способность личности осуществлять нравственный самоконтроль, самостоятельно формулировать для себя нравственные обязанности, требовать от себя их выполнения и производить самооценку совершаемых поступков, это одно из выражений нравственного самосознания личности, глубокое внутреннее осознание того, что правильно, а что нет, принципы поведения и ощущение того, насколько наши мысли и поступки им соответствуют. Совесть проявляется как в форме рационального осознания нравственного значения совершаемых действий, так и в форме эмоциональных переживаний (например, "угрызений совести"). Именно совесть дает нам критерий самооценки целей и путей их достижения.

Воля - свойство человеческой психики, проявляющееся в активной самодетерминации и саморегуляции человеком своей деятельности и поведения вопреки внешним и внутренним препятствиям, влияниям и воздействиям. Эффективным людям совсем не обязательно нравится делать то, что они делают, но воля позволяет им подчинять свои чувства силе своей цели.

Именно эти четыре качества дают человеку возможность самому определять программу своей жизни и настойчиво добиваться поставленных целей.

Управляем своей жизнью

Как и в любом другом процессе управления, в управление своей собственной жизни мы можем выделить четыре постоянно повторяющихся этапа:

1. Определяем цели – формулируем то, что мы хотим достичь.
2. Формируем стратегию – определяем, каким образом мы собираемся достичь цели.

3. Реализуем личные проекты (наши дела), которые преобразуют стратегии в действия, а цели в реальность.
4. Обратная связь. Жизнь и окружение постоянно навязывают нам собственные сценарии. Если мы способны распознать неэффективные сценарии и неточные или неполные парадигмы (карты) внутри себя, то мы можем проактивно приступить к переписыванию себя заново.

Правильно определять свои цели и уметь на них фокусироваться - одно из основных качеств эффективного лидера. «Когда человек не знает, к какой пристани он держит путь, для него никакой ветер не будет попутным», - сказал 2 тысяч лет назад Сенека Луций Аней. Виктор Франкл [8] утверждает, что мы *выявляем, а не придумываем* свои жизненные цели. Говоря словами Франкла «У каждого есть свое особое призвание, или миссия, в жизни... Следовательно, каждый человек незаменим, а жизнь его неповторима. И поэтому задача каждого человека настолько же уникальна, насколько уникальна и его возможность выполнить эту задачу». Важно, что, какие бы цели мы себе не назначали, они должны основываться на универсальных человеческих принципах. Принципы – это естественные законы человеческого бытия, которые столь же реальны, столь же неизменны и бесспорны, как, скажем, закон гравитации в физике. Почти все религии и все философские направления, которые получили мировое распространение, согласны в одном: единственно возможное на земле счастье человека состоит в том, чтобы следовать Божьим заповедям (принципам) основные из которых:

- Человек – высшая ценность.
- Совершенствуй себя.
- Совершенствуй мир.

Принципы – это сама территория, объективная реальность, в которой мы существуем и действуем.

История 1. «Мы изменили курс...»

Стивен Кови цитирует рассказ Фрэнка Коха, опубликованный в "Записках" – журнале Военно-морского института.

«Двум военным кораблям учебной эскадры пришлось провести несколько дней на маневрах в бушующем море. Я служил на головном корабле и, когда стало смеркаться, заступил на вахту. Из-за тумана видимость была плохой, поэтому капитан остался на мостике следить за действиями команды.

Вскоре после того, как стемнело, впередсмотрящий доложил:

— Огни прямо по курсу!

— На месте или движутся? – выкрикнул капитан.

— На месте, капитан! – ответил впередсмотрящий, и это означало, что мы следуем курсом, грозящим столкновением с этим судном.

Тут капитан приказал сигнальщику:

– Передайте на судно: "Движемся курсом на столкновение, советую вам изменить курс на 20 градусов!"

Получили ответный сигнал:

— Желательно, чтобы вы изменили курс на 20 градусов!

Капитан говорит:

— Передай: "Я – капитан, измените курс на 20 градусов!"

— Я – моряк второго класса, – приходит ответный сигнал. – Советую изменить курс на 20 градусов!

К тому времени капитан распалился не на шутку.

– Сигнальте! – рявкнул он. – "Я – военный корабль! Измените курс на 20 градусов!"

Последовал ответ световым морзе:

– Я – маяк!

Мы изменили курс...»

Принципы подобны маякам. Принципы представляют собой естественные законы, управляющие цивилизованным обществом на всем протяжении истории. Они фундаментальны. Они – естественные законы, которые невозможно нарушить или обойти.

Наши ценности – это карты, которые помогают нам ориентироваться в окружающем нас мире. Наши цели в каждый момент жизни – это конкретные пункты на карте наших ценностей, которые мы для себя выбираем. Нашими ценностями могут быть семья, деньги, работа, удовольствия, «я сам» и много что еще. Однако концентрация наших усилий в направлении только на одну из этих ценностей само по себе не может принести счастья (Таблица 7).

Таблица 7. Жизненные ценности

Ценность	За	Против
Семья	Вы можете фокусировать свои усилия на семье, направлять туда серьезные инвестиции. Любовь и взаимопонимание делают жизнь более полноценной	Вы подчиняете свою жизнь желаниям и потребностям членов вашей семьи. Ваше влияние ограничено семейным кругом.
Деньги	Экономическая безопасность – это то, что дает человеку возможность действовать во всех других сферах жизни. В иерархии потребностей человека физическое выживание и финансовая безопасность идут первыми. Пока они не будут удовлетворены, хотя бы в минимальной степени, прочие потребности даже не заявят о себе.	Материальное благополучие не может быть целью, это лишь средство достижения других целей.
Работа	. Восприятие себя и чувство самооценки тесно связаны с работой: "я программист", "я бизнесмен", "я художник".	Ваша жизнь ограничена рамками профессиональной роли. Вам хорошо только тогда, когда вы работаете. Вы отождествляете работу с жизнью.
Удовольствия	Развлечения в разумных дозах позволяют отдохнуть и телу, и душе и могут идти на пользу семейным отношениям и отношениям с другими людьми.	Вы видите жизнь только с точки зрения удовольствий, которые она может вам дать (вспомним Б.Гребенщикова: «они сходят с ума от того, что им нечего больше хотеть»).
«Я сам»	Сосредоточенность на самом себе. «Я ни от кого не завишу, я живу и развиваюсь для себя».	Эгоизм. Ваша способность действовать ограничена вашими ресурсами. Преимущества взаимозависимости вам недоступны.

Какие бы ценности не определял для себя человек в каждый конкретный жизненный момент, какие бы он цели не достигал, только реальное развитие и движение человека в пространстве, заданном базовыми принципами, способно сделать его жизнь подлинно счастливой.

Для выявления своих целей и формирования стратегии их достижения мы используем свои исключительные человеческие качества: самосознание, воображение, совесть. Это первый этап творения – мысленное созидание. Для реализации наших целей – физического творения – нам необходимо такое качество, как независимая воля.

Мало определить самые лучшие цели и составить самый точный план их достижения, надо еще быть эффективным при движении по выбранному пути. Стивен Кови предлагает разделять все наши текущие дела на четыре квадранта (Таблица 8).

Таблица 8. Четыре приоритета наших дел

	Срочные	Не срочные
Важные	I Критические ситуации Неотложные проблемы Задачи с «горящим сроком»	II Восстановление Развитие Планирование Поиск возможностей Взаимоотношения
Не важные	III Совещания Телефон Переписка	IV Пустая трата времени

К важным делам следует относить только то, что вносит вклад в нашу программу, наши ценности и способствует продвижению к нашим наиважнейшим целям. Суть управления на основе приоритетов проста и состоит в следующем:

- Важные и срочные – делаем в первую очередь.
- Важные и не срочные - делаем в первую очередь.
- Срочные но не важные – не делаем.
- Не срочные и не важные – не делаем.

Нужна воля, чтобы сказать «нет» срочным, но не важным. Гете говорил: «Наиболее важные дела никогда не должны подчиняться наименее важным». Нужна воля, чтобы всегда находить время на важные, но не срочные дела. Сосредотачивая внимание на квадранте два, мы работаем на опережение и сокращаем дела из квадранта один в будущем («точим пилу»). Признанный авторитет в области менеджмента Питер Друкер утверждает, что «Эффективный человек мыслит не проблемами, он мыслит возможностями. Эффективные люди питают возможности и морят голодом проблемы. Они мыслят упреждающе». Управление на основе приоритетов основа эффективного управления личным временем.

Путь повышения эффективности человека - переход от зависимости к независимости. Человек осознал свои глубинные ценности и цели, действует в соответствии со своими жизненными принципами, независимо от условий и обстоятельств. Эффективный человек сам себе лидер, сам выявляет свои цели, сам пишет программу своей жизни. Наше самосознание, воображение и совесть позволяют сделать нам правильный выбор. Наша воля делает эффективное самоуправление возможным.

Эффективный программист

Не может быть эффективной команды, если участники не знают и не умеют делать свое дело. Р.М.Белбин в своей книге [19] приводит список факторов незрелости и неэффективности сотрудника:

- Не соблюдает инструкции
- Плохо контролирует время
- Не любит, когда контролируют его работу
- Не обращает внимание на качество работы
- Не может сконцентрироваться на работе
- Имеет личные проблемы
- Преувеличивает свои способности
- Не исполняет свою долю работы
- Не любит изменений в работе
- Не сообщает об ошибках
- Не лоялен по отношению к своей компании

В большинстве своем эти отрицательные качества носят временный характер и происходят от отсутствия опыта и недостаточной самостоятельности специалиста. Дружеская поддержка и помощь, как правило, позволяют справиться с большинством перечисленных проблем.

История 2. «Программист Ашманова» [20]

Программист:

«Чего там планировать, я быстрее сделаю и всё уже будет работать».

«Планировать разработку бессмысленно, жизнь всё равно богаче».

«Программные проекты всегда срывают сроки потому, что это сложное и творческое дело, вроде научных исследований».

«К пятнице готово не будет, но в понедельник - точно. Или во вторник».

«Я уже неделю ночами работаю, а вы меня укоряете за срыв срока».

Диагноз. Не достаточно самостоятельный программист. Не овладел навыками персонального лидерства. Не умеет четко ставить цели, проводить декомпозицию сложности, анализировать риски и планировать решение задачи. Нельзя доверять полностью самостоятельную работу.

Рекомендации. Помощь и поддержка более опытного коллеги. Декомпозиция сложности задачи на более простые и обозримые работы. Директивное управление, короткие задания на 4-8 рабочих часов, постоянный контроль.

Уоттс Хэмфри и Институт технологий разработки программного обеспечения Software Engineering Institute (SEI) создали индивидуальный процесс разработки ПО Personal Software Process (PSP) [21], который предоставляет детальные описания методов планирования и оценки, показывает разработчикам, как измерять собственную продуктивность и соотносить ее с существующим планом, объясняет, почему описанные методы могут помочь им в их работе.

Авторы данного подхода уверены, что прежде чем стать эффективным командным игроком, программист должен сначала достичь личной независимости и индивидуальной эффективности. Ниже приведены профессиональные компетенции, которыми должен обладать, согласно PSP, эффективный программист:

- учитывать время, затраченное на работу над проектом;

- учитывать найденные дефекты;
- классифицировать типы дефектов;
- оценивать размер задачи;
- освоить систематический подход к описанию результатов тестирования;
- планировать программные задачи;
- распределять их по времени и составлять график работы;
- индивидуальную проверку проекта и архитектуры;
- осуществлять индивидуальную проверку кода;
- выполнять регрессионное тестирование.

PSP призван помочь программисту полнее понять отношение между размером программы и временем, которое он тратит на ее разработку, должен сформировать упорядоченную систему, в рамках которой он может производить оценки, выполнять задачи, проверять статус работы и записывать результаты. Всем разработчикам ПО постоянно задается один и тот же вопрос: «Сколько времени у вас это займет?». Освоение индивидуального процесса помогает разработчикам научиться реалистично отвечать на этот вопрос.

Есть данные [22], что владение участниками команды PSP обеспечивает снижение числа ошибок в программах в среднем на 73%, рост точности оценки объема предстоящих работ на 26%, а точность оценки времени работы над проектом - на 40%. При этом производительность труда программистов (число строк кода в час) увеличивается на 21%.

Эффективный программист помимо технических знаний и умений должен обладать еще и личными компетенциями, необходимыми для командной работы:

- Занимает активную позицию, стремится расширить свою ответственность и увеличить личный вклад в общее дело.
- Постоянно приобретает новые профессиональные знания и опыт, выдвигает новые идеи, направленные на повышение эффективности достижения общих целей, добивается распространения своих знаний, опыта и идей среди коллег.
- Получает удовольствие от своей работы, гордится ее результатами и стремится, чтобы эти же чувства испытывали все коллеги.
- Четко осознает свои личные и общие цели, понимает их взаимообусловленность, настойчиво стремится к их достижению.
- Уверен в себе и в своих коллегах, объективно оценивает их достижения и успехи, внимательно относится к их интересам и мнениям, активно ищет взаимовыгодное решение в конфликтах.
- Является оптимистом, при этом твердо знает, что окружающий мир несовершенен; воспринимает каждую новую проблему, как дополнительную возможность подтвердить собственный профессионализм в своих глазах и во мнении коллег.

Бывает так, что, вроде бы, сильный специалист приносит команде больше вреда, чем пользы. Порой встречаются «клинические случаи», которые, на мой взгляд, наиболее эффективно лечить «хирургическим» путем – избавлять команду от проблемных людей. К патологиям поведения, которые неприемлемы в команде, следует отнести следующие:

- *Непорядочность*. Лживость, отсутствие совести и чувства справедливости, способность на низкие поступки.
- *Синдром острого дефицита эмпатии*. Эгоцентризм. Неуважение и невнимание к партнерам. Склонность к отрицательным оценкам других. Грубость. «Каждый сам за себя! – никто тебе не поможет!» «Человек человеку волк!»
- *«Звезданутость»*. Завышенная самооценка. Ощущение собственного превосходства. Умничанье. Человек сильно переоценивает свой личный вклад в общее дело и поэтому считает, что он должен работать меньше, чем его «менее способные» коллеги.
- *Вульгарный анархизм*. Я много говорю о свободе, как неотъемлемом атрибуте творчества и это действительно так. Но в российской традиции свобода часто подменяется понятием «вольница», а это совсем ни одно и то же. Свобода индивида заканчивается там, где начинается свобода другого. Свобода это диалектическое единство права выбора и принятия на себя ответственности за его последствия. Вольница – это полная безответственность, свобода от каких либо обязательств перед другими, ничем не сдерживаемые проявления чувств, действия или поступки. «Произвольничать, поступать самовольно, в обиду другим, нагло, дерзко» (с) В.Даль.
- *«Социальный паразитизм»*. Стремление прожить вольготно за чужой счет там, где ответственность размыта, а личный вклад трудно четко выделить.

История 3. «Суперспециалист»

Характеризуется следующими особенностями поведения:

Активен, самостоятелен, напорист.

По любому вопросу имеет свое особое мнение.

Всегда стремится быть победителем в спорах.

Любит оценивать других и указывать им на их недостатки.

Использует любой повод, чтобы продемонстрировать свое превосходство.

Диагноз. Человек - эгоист. Человек достиг личной независимости, но не осознал взаимозависимости. Неспособен к конструктивному взаимодействию. Самооценка, скорее всего, неадекватно завышенная.

Рекомендации. Избавиться. Если нет такой возможности, попробуйте найти для него четко специфицированную, достаточно сложную изолированную задачу, которая не находится на критическом пути проекта. Например, спроектировать и написать адаптер к SAP/R3 по заданному интерфейсу. Четко определите контрольные точки и критерии оценки работы в них. Будьте готовы неделями слушать о чрезвычайной сложности интегрируемой системы и трудности решаемой задачи, вместо представления полученных результатов. Имейте под рукой другого специалиста, который сможет решить эту задачу, когда потребуется.

Люди, склонные к принятию решений на основе «этики», могут обвинить меня здесь в бездушии и нежелании заниматься воспитательной работой. Мое мнение, что людей воспитывают в детском саду, ну еще немного в начальной школе. Дальше люди воспитываются только самостоятельно, а окружающие могут лишь помогать или не мешать в этом процессе. Убежден, что каждый взрослый человек имеет то, к чему он осознанно или неосознанно стремится. Нянчиться и воспитывать человека - это значит ограждать его от проблем, закрывать ему путь к переосмыслению своего опыта и развитию, «загонять болезнь внутрь» при помощи «социального аспирина».

Думать и действовать в духе «выиграл/выиграл»

Одно из коренных изменений, которое пришло к нам вместе с новой общественно-экономической формацией, - это изобилие товаров и услуг. Сегодня в мире всего хватит на всех. Сегодня, в отличие от индустриальной эпохи следует рассматривать жизнь как игру с *ненулевой* суммой. Это значит, что в любой конфликтной ситуации, как правило, можно думать и действовать в духе «выиграл/выиграл» - искать и находить *третью альтернативу*, которая способна обеспечить выигрыш обеим сторонам.

Мы все родом из предыдущих эпох, когда выигрыш в конфликте одной стороны, неизбежно должен был приводить к равноценному проигрышу другой. Поэтому мы часто пытаемся применять именно подход игр с нулевой суммой, не осознавая, что в новых условиях существует более эффективный подход.

Для иллюстрации рассмотрим условную кризисную ситуацию в ядерном противостоянии двух сверхдержав: стороны А и Б. Кризис заключается в том, что в заранее определенный момент времени «Ч» каждая из двух сторон может нажать или не нажать на «ядерную кнопку». В случае, если обе стороны одновременно нажимают кнопку, итоговый проигрыш каждой из сторон составляет по 30% промышленного потенциала. Если сторона А наносит удар в момент «Ч», а сторона Б нет, то итоговый проигрыш стороны А составит лишь 10%. Если, наоборот, сторона Б наносит удар в момент «Ч», а сторона А нет, то итоговый проигрыш стороны А увеличится до 80%. Рассуждая в соответствии со стратегией игры с нулевой суммой за сторону А мы должны составить для себя матрицу наших потерь:

		Сторона Б	
		Наносит удар	Не наносит удар
Сторона А	Наносит удар	30%	10%
	Не наносит удар	80%	0%

Следующим нашим шагом должно быть решение задачи минимизации математического ожидания потерь при неизвестной стратегии стороны Б. Поскольку стратегия нам неизвестна, мы будем полагать что события сторона Б «наносит удар» и «не наносит удар» равновероятны и равны 0.5.

Тогда математическое ожидание потерь игрока А при обоих возможных стратегиях будет:

$$A \text{ (наносит удар)} = 30 * 0.5 + 10 * 0.5 = 20 (\%)$$

$$A \text{ (не наносит удар)} = 80 * 0.5 + 0 * 0.5 = 40 (\%)$$

Исходя из нашего стремления уменьшить возможные потери мы со всей очевидностью должны избрать для себя стратегию «сторона А наносит удар».

Однако, адекватное понимание обеими сторонами того, что данная ситуация не описывается теорией игр с нулевой суммой приводит к единственно правильному решению – не наносить удар, которое гарантирует взаимовыгодное отсутствие проигрыша как для стороны А, так и для стороны Б.

Думать и действовать в духе "выиграл/выиграл" – это особый настрой, направленный на постоянный поиск взаимной выгоды при всех взаимодействиях людей друг с другом. «Выиграл/выиграл» означает, что все договоренности и решения обоюдно выгодны и удовлетворяют обе стороны. В условиях взаимозависимой реальности любой выбор, меньший чем «выиграл/выиграл», будет слабым и бледным и окажет негативное влияние на долгосрочные отношения.

Разрабатывали площадку Интернет-торговли для компании с мировым именем. Система состояла из трех компонентов: подсистема on-line заказов, подсистема обработки заказов службой поддержки клиентов и подсистема подготовки и сопровождения каталога продукции. Были определены этапы промежуточных поставок и срок ввода системы в опытную эксплуатацию.

Проект продвигался успешно. Подсистема on-line заказов была поставлена и протестирована клиентом в срок. Подсистема обработки заказов – тоже. А на третьем этапе, как это часто бывает, столкнулись с ошибкой промежуточного ПО, для которой, сколько не старались, не смогли найти обхода. Если коротко, то использованная система, просто не масштабировалась на такой объем данных, с которым собирался работать Заказчик. Применение данного ПО было одним из технических условий контракта, на котором настоял Заказчик. Ничего не смог предложить и уважаемый разработчик этого программного продукта. Да, он признал, что ошибка в его программе есть, да, он подтвердил, что обходное решение этой проблемы ему неизвестно, да, он включил исправление этой ошибки в план очередного релиза, который будет поставлен приблизительно через год.

Стало ясно, что третий компонент системы не будет поставлен в срок. Два очевидных альтернативные решения данной проблемы были рассмотрены в первую очередь.

Первое. Вариант «проиграл/проиграл». Заказчик аннулирует контракт (де-юре он, скорее всего, смог бы это сделать), мы возвращаем, ранее полученные по контракту деньги. Результат: само собой разумеется, убытки несет наша компания; убытки в виде упущенной прибыли из-за отсутствия автоматизированной системы несет заказчик.

Второе. Вариант «выиграл/проиграл». Мы включаем в контракт дополнительный пункт по разработке той функциональности, которая не доступна в базовом продукте, за дополнительную (и не малую, поскольку функциональность сложная) плату. Результат: мы выигрываем - получаем дополнительную прибыль по контракту; Заказчик несет дополнительные расходы.

Описываемая история примечательна тем, что на основе накопленного капитала взаимного доверия, которое установилось между исполнителями и Заказчиком в ходе успешной совместной работы, удалось найти еще одно решение, третью альтернативу, которая обеспечила выигрыш обеим сторонам.

Мы пересмотрели контракт. Суть изменения состояла в следующем. Для того чтобы ввести систему в эксплуатацию Заказчику требовалось, используя нашу подсистему подготовки и сопровождения каталога (как раз тот компонент, который не был готов), конвертировать его текстовую версию, с которой они работали прежде, в структуру данных новой системы. В силу большого объема информации это была хоть и разовая, но достаточно трудоемкая и, следовательно, дорогая работа. Мы предложили, за те деньги, которые оставались по контракту на разработку третьего компонента написать утилиту, которая хоть и не полностью автоматизирует импорт, но позволит существенно сократить объем рутинной ручной работы при этой операции. А разработку системы подготовки и сопровождения каталога было принято решение перенести на следующий год, когда поставщик промежуточного ПО исправит ошибку, которая стала препятствием в нашей работе.

В результате. Мы «выиграли», потому что получили дополнительную работу. Заказчик тоже «выиграл», потому что смог серьезно сэкономить на ручной конвертации каталога и смог ввести автоматизированную систему в эксплуатацию раньше срока. А поскольку бизнес Заказчика был устроен так, что каталог продукции обновлялся только раз в год, то внедрение третьего компонента нашей системы можно было безболезненно отложить на год.

И еще один интересный факт из этой истории. Описанная история происходила на фоне кризиса, в котором оказался Заказчик, после теракта 11 сентября 2001 года. В компании было объявлено тотальное 30-типроцентное сокращение. Но в результате внедрения новой

системы, количество заказов возросло настолько, что службу поддержки клиентов не только не сократили, но и расширили ее штат.

Руководителю дана власть, у него всегда есть возможность использовать ее и действовать в конфликте в духе «выиграл/проиграл». «Выиграл/проиграл» соответствует авторитарному стилю руководства: «Будет по-моему, а не по-твоему!». Люди с установкой «выиграл/проиграл» склонны использовать свое положение или личные качества, чтобы всегда добиваться своего. Но попробуйте применить принцип «выиграл/проиграл» к клиентам своей компании, много их у вас после этого останется?

Часто приходилось сталкиваться с ситуацией, когда коллега настойчиво отстаивает решение, которое на моей практике уже показало свою неэффективность. Исходя из соображений сиюминутной выгоды, очень хочется настоять на своем подходе, но в творческой деятельности обязательным элементом ответственности является свобода выбора пути решения стоящей проблемы. Даже если вы уверены в существовании более правильного решения, вы не вправе настаивать на нем, вы можете высказать свое мнение только в виде рекомендации. Если вы, все-таки, окажитесь правы, это обеспечит вам дополнительный прирост результата в формуле эффективности – увеличит капитал доверия. «Продавливание» своего решения приведет к невосполнимым потерям в доверии между вами и другими участниками проекта. Впрочем, вы ведь тоже можете ошибаться.

Сегодня в нашей отрасли мало что можно сделать в одиночку. Одна из особенностей современного производства ПО состоит в том, что это вид *коллективного* творчества. При этом от 30 до 50% рабочего времени участников программного проекта тратится на коммуникации и взаимодействия, эффективность которых определяется их способностью к сопереживанию, взаимопомощи и синергии. Только человек осознавший взаимосвязанность и взаимозависимость, научившийся думать и действовать в духе "выиграл/выиграл" может стать по настоящему эффективным.

Коммуникации

Очень важно, чтобы каждый член проектной команды знал, или хотя бы имел при желании возможность узнать о проекте не меньше, чем его менеджер. И не стоит ничего скрывать, ни проблем с заказчиком, ни разногласий с руководством. Трудно ожидать самостоятельные, нестандартные и эффективные решения от человека, который видит проблему только с одной стороны.

Важнейшим неформальным макро-показателем состояния проекта являются коммуникации, их качество и количество. Только благодаря эффективным коммуникациям можно достигать синергетического эффекта, который отличает команду от просто группы. Мне неоднократно приходилось убеждаться, что освоение новой технологии парой программистов, которые осуществляют интенсивный обмен знаниями, происходит минимум в 3 раза быстрее, чем в случае, когда ту же работу выполняет один программист.

Недостаточное количество коммуникаций свидетельствует, как правило, об отсутствии команды, каждый углублен в свою задачу и не интересуется, что делают его коллеги. В результате будет сделано не то, что нужно, а то, что будет сделано, вряд ли удастся интегрировать в единую систему. Если через день после получения недельного задания у программиста не возникло уточняющих вопросов, жди беды. Скорее всего, он с головой ушел в «проектирование дома, забыв уточнить, для чего он предназначен» [23]. Учитывая интроверсию большинства программистов и несклонность их к общению, менеджеру требуется прилагать значительные усилия для того, чтобы мотивировать необходимый уровень коммуникаций в проекте.

Каждая коммуникация имеет определенную цель, которой может быть одна из следующих:

- *Получения информации.* Например, вопросы или наведение справок о состоянии работы по проекту. Если вопросы сформулированы неаккуратно, может восприняться как сомнение в компетентности. «Ты до сих пор не сделал это?»

- *Высказывание мнения.* Передача своего видения ситуации, вопроса. Надо делать это сообщение аккуратно. Члены команды могут воспринять это как обучение или руководство, в котором они не нуждаются.
- *Обучение, инструктирование или руководство.* Передача информации и побуждение к действию. Имеет смысл, если собеседник сам осознает необходимость такого подхода.
- *Подтверждение, поддержка, поощрение.* Мы сообщаем собеседнику, что слышим и понимаем его. Мы информируем собеседника о том, что разделяем и поддерживаем его мнение, действия.
- *Распоряжение или приказ.* Силовой тип коммуникации в духе «выиграл/проиграл». Разрушительно действует на доверие. Эффективность будущего взаимодействия сильно снижается. Допустимо применение только в критических ситуациях.

Четкое понимание цели обеими сторонами – необходимое условие достижения эффективности.

Эффективность коммуникации сильно зависит от способа, которым мы ее осуществляем. Алистер Коуберн [2] приводит график (Рисунок 1), который демонстрирует резкое снижение эффективности коммуникации при переходе от личного общения «глаза в глаза» к другим способам общения.

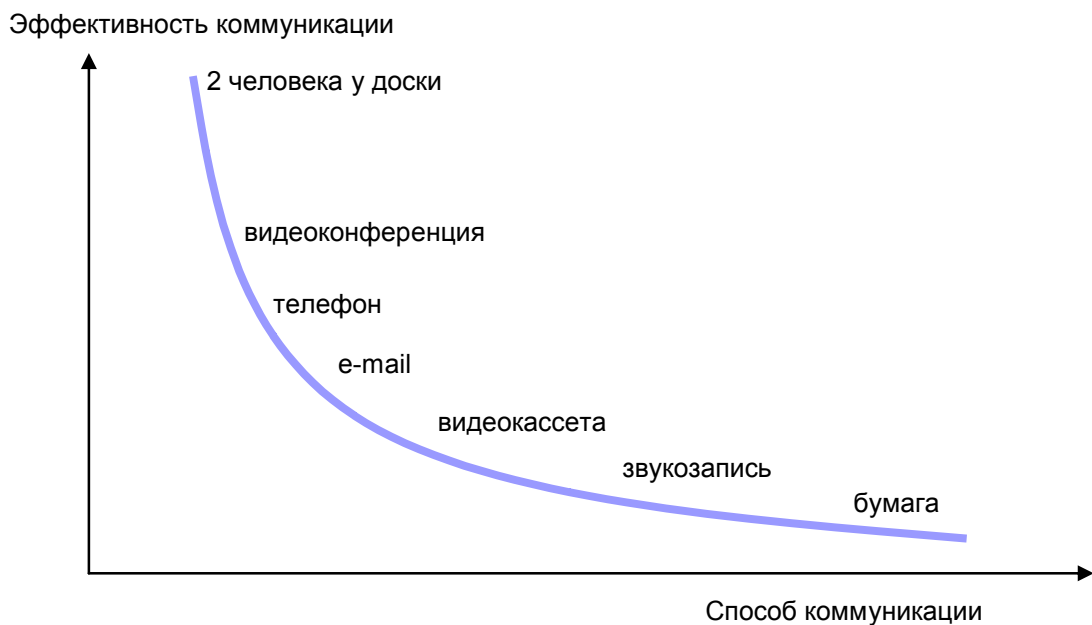


Рисунок 1. Зависимость эффективности коммуникации от способа ее осуществления

Наиболее эффективно происходят коммуникации в случае, когда люди находятся вместе в одной небольшой комнате. Поэтому, на мой взгляд, 5-7 человек оптимальный размер для эффективной команды.

По оценкам экспертов в области общения:

- только 10% информации передается через слова;
- 30% передается через интонацию;
- 60% – через язык мимики и жестов (а может быть еще через что-то, что, например, не передает телевизионное изображение).

Умение распознавать и правильно интерпретировать невербальные знаки, с помощью которых передается большая часть информации, один из обязательных навыков эффективного общения. Настоятельно рекомендую книгу Алана Пиза [24] для более глубокого изучения данного вопроса. К наиболее распространенным невербальным сигналам, которые вольно или невольно может подавать ваш собеседник, относятся следующие (Таблица 9).

Таблица 9. Интерпретация невербальных сигналов

Сигнал	Интерпретация
Размещение во главе стола, высокое кресло.	Стремление к доминированию.
Переговоры, сидя напротив по разную сторону стола.	Противостояние.
Скрещение рук, ног.	Оборона.
Сжатые кулаки. Наклон головы вперед. Взгляд исподлобья. Взгляд прямо в глаза. Учащенное дыхание. Губы напряжены, часто сжимаются.	Агрессия.
Стремление прикрыть рот (или, возможно, замаскированный жест: почесать нос, разгладить усы, коснуться кончика губ). Взгляд в сторону. Покраснение лица.	Не уверенность в своих словах. Ложь.
Прикосновение к глазу, уху.	Недовольство тем, что видим или слышим.
Наклон корпуса вперед. Взгляд в переносицу.	Заинтересованность.

Для построения эффективной коммуникации необходимо обязательно учитывать индивидуальные особенности людей. Для каждого типа личности существует свой наиболее эффективный способ общения. Если говорящий не принимает во внимание индивидуальные особенности слушающего, коммуникации, как правило, заходят в тупик. При этом будет отсутствовать (полностью или частично) передача информации. Например, если сообщение по форме не соответствует личности участника коммуникации, то получатель может почувствовать, что к нему отнеслись с пренебрежением, и не воспринять переданную информацию. «Вы так громко говорите, что я вас не слышу!» Или, говорящий при передаче информации подразумевает нечто, что неизвестно получателю. Следует отметить особенности в стиле общения, которые проявляются в зависимости от характеристики типа личности (Таблица 10).

Таблица 10. Зависимость особенностей в стиле общения от характеристик типа личности

E – Extroverted (экстраверт)	I – Introverted (интроверт)
<p>Предпочитают быть в курсе всех дел и участвовать в процессе коммуникации.</p> <p>Склонны приходить к собственным умозаключениям, обсуждая свои идеи с другими.</p> <p>Обычно много и быстро говорят, полагаются на высказывания других, а иногда, будучи возбужденными, могут и прервать других.</p> <p>Обдумывают свои идеи вслух.</p> <p><i>Интровертам такой стиль может показаться напористым и чересчур прямолинейным.</i></p>	<p>Отдают предпочтение письменной форме общения.</p> <p>Им необходимо сначала обдумать свои взгляды и суждения и только потом представить их на суд общественности.</p> <p>В процессе коммуникации могут ограничиваться всего лишь несколькими намеками (далеко не всегда очевидными для постороннего наблюдателя), не выражая при этом своих мыслей.</p> <p><i>Экстраверты могут интерпретировать такой стиль как незаинтересованность или невовлеченность интроверта в процесс коммуникации. У них даже может сложиться</i></p>

	<i>впечатление, что интроверт относится к чему-либо неодобрительно, с осуждением.</i>
S – Sensing (конкретное восприятие)	N – iNtuitive (интуиция)
<p>Обычно делает акцент на том, что было и есть сейчас реально и актуально: кто? что? где? когда?</p> <p>Предпочитают примеры из жизни и ждут того же от других.</p> <p>В процессе коммуникации задают много конкретных вопросов, чтобы выяснить детали, и затем предоставляют исчерпывающую информацию или примеры.</p> <p><i>Интуитивный тип считает такие вопросы и реакции придирчивыми, замедляющими процесс, пессимистичными, препятствующими изменениям, а также скучными и прозаичными.</i></p>	<p>Предпочитают, чтобы результатом общения были какие-то новые связи и более глубокие цели.</p> <p>В процессе коммуникации их внезапно осеняет, они перескакивают с одной темы на другую и предлагают различные точки зрения.</p> <p><i>Сенсорному типу вопросы и реакции, требующие проявления интуиции кажутся несущественными и не относящимися к делу, затягивающими процесс, а также нереалистичными и бесполезными.</i></p>
T – Thinking (логика)	F – Feeling (этика)
<p>Важны логика и ясность в процессе коммуникации, а также те сообщения, на которых необходимо делать акцент в работе.</p> <p>Сразу же выключится из беседы и перестанет принимать участие, как только поймет, что слишком много времени тратится на несущественные (с его точки зрения) моменты, например, выражение личных эмоций или невнятные высказывания.</p> <p><i>Его суждения могут быть интерпретированы этическим типом как критические, негативные и холодные. Представителей мыслительного типа могут считать людьми, которым нет дела до других.</i></p>	<p>Предпочитают, чтобы высказываемые сообщения не шли вразрез с интересами и системой ценностей как их собственных, так и других людей.</p> <p>Сразу же перестанет принимать участие в беседе, если в процессе коммуникации чувствуется некоторая разобщенность, начинаются какие-то межличностные трения, выступающий не вызывает симпатии или же личностный фактор не принимается в расчет.</p> <p><i>Логический тип может считать его легкомысленным, не в меру мягким и даже пристрастным.</i></p>
J – Judging (рациональность)	P – Perceiving (иррациональность)
<p>Предпочитают, чтобы любые коммуникации были исчерпывающими и ориентированными на конечную цель.</p> <p>Отдают предпочтение четкой структуре – коммуникативный процесс должен иметь начало, содержательную часть и заключение.</p> <p><i>Иррациональный тип чувствует в процессе такого взаимодействия давление, напряжение и контроль.</i></p>	<p>Предпочитают сначала настроиться на общение, а затем поддерживать его.</p> <p>Изменяют курс в интересующих их направлениях, предлагают дополнительную информацию, оставляют вопрос открытым.</p> <p>Информативная часть и проводимые исследования для них гораздо важнее, чем заключение.</p> <p><i>Рациональный тип может считать их неорганизованными, неспособными принимать</i></p>

В качестве иллюстрации тупиковой коммуникации, в которой не учитываются особенности типов личности собеседников можно привести пример общения экстраверта (И.Иванов) и интроверта (В.Пупкин).

История 5. Экстраверт и интроверт

Коммуникация

И.Иванов. Активно обсуждает вопрос, много говорит, перескакивает с одного на другое, рассуждает вслух, часто перебивает собеседника.

В.Пупкин. Предпочитает молчать, отвечает кратко, пытается сузить обсуждаемую тему.

Результат

И.Иванов: «Этот Пупкин просто не интересуется проектом! У него, видимо, есть дела поважней, о которых он все время думает!»

В.Пупкин: «Этот Иванов постоянно пытается на меня давить, навязывает свои решения! Не хочет меня слышать! Он, наверное, пытается пробиться в начальники!»

Другой пример коммуникационных проблем, которые могут возникнуть при общении между интуитом (И.Иванов) и человеком с конкретным восприятием (В.Пупкин).

История 6. Интуиция и конкретное восприятие

Коммуникация

И.Иванов. Пытается решать проблему для самого общего случая, повторяется, рассматривая вопрос с разных сторон, пытается связать обсуждаемую проблему с другими.

В.Пупкин. Постоянно задает вопросы: «А кто? А где? А когда? А ты это пробовал? А сколько раз? А это нам сейчас надо?»

Результат

И.Иванов: «Этот Пупкин просто тянет время своими глупыми вопросами! Он не хочет ничего менять! Лишь бы нечего не делать!»

В.Пупкин: «Этот Иванов опять рассуждает о «сферических конях в вакууме»! Конкретные вопросы его не интересуют! Будет и дальше постоянно генерировать свои новые идеи! Лишь бы нечего не делать!»

И еще одна иллюстрация неэффективной коммуникации, в которой один из партнеров принимает решения на основе логики (В.Пупкин), а другой - на основе этики (И.Иванов).

История 7. Этика и логика

Коммуникация

И.Иванов. «Я переживаю за качество нашего кода, ведь кому-то придется его сопровождать! Пользователи нашей системы люди, а не машины, они должны получать удовольствие от работы с нашим интерфейсом!»

В.Пупкин. «Если следовать логике, Заказчик нам платит за функциональность, а не за качество кода или «бантики» на интерфейсе! А у нас сроки и бюджет! Сколько мы потратим времени на комментирование кода и переделку интерфейса? Ну и что, если пользователям необходимо одновременно работать в трех окнах? Потренируются и будут это делать быстро! Эргономика - выдумки специалистов по юзабилити!»

Результат

И.Иванов: «Это Пупкин думает только о сиюминутной прибыли! Что скажут люди в будущем о нашем продукте, его не беспокоит! Его не волнует, что подумают о нем те, кто будет сопровождать нашу систему! Доброе имя компании для него пустой звук!»

В.Пупкин: «Этот Иванов заботится только о том, чтобы все всегда были довольны! Прибыль и интересы компании для него ничего не значат!»

Если при коммуникациях получатель информации постоянно чувствует, что не принимаются во внимание его личностные особенности, «с ним не считаются», то это может привести его к фрустрации, которая будет проявляться в гнетущем напряжении, тревожности и чувстве безысходности.

Следующее необходимое условие эффективности коммуникации - умение активно слушать [3]. Люди разные, и понять, какие они на самом деле, понять их чувства и намерения, можно, только слыша, что они говорят. Мало просто слушать собеседника важно *слышать*, что он говорит. Необходимо сначала стремиться понять, а потом – быть понятым, сначала ставить диагноз, а потом выписывать рецепт, – это трудный, но абсолютно верный принцип для самых разных сфер жизни. Прежде чем поднимать проблему, прежде чем оценивать и предписывать, прежде чем излагать свои идеи – стремитесь понять. Такой подход отличает всех истинных профессионалов. Программист-непрофессионал просто пишет код. Профессионал решает проблемы и удовлетворяет потребности пользователей.

Люди противятся всякой попытке манипулировать ими. Никакие приемы или техники, которые мы умеем применять, не будут эффективными, если они не будут основаны на искреннем желании понять. Люди стремятся быть понятыми. И сколько бы времени ни пришлось этому посвятить, отдача будет гораздо большей, поскольку ваши действия будут опираться на точное понимание проблем и обстоятельств.

Когда мы по-настоящему глубоко понимаем друг друга, мы открываемся для творческих решений и третьих альтернатив. Отличия между нами перестают быть непреодолимыми препятствиями для общения и развития. Наоборот, они становятся ступенями, ведущими к синергии.

Специалисты по эффективности коммуникаций выделяют четырех уровнях, на которых мы можем слушать:

1. *Игнорировать* говорящего, не слушать его вообще или делать вид, что слушаем: "Угу! Да-да! Так-так!" Так, обычно мы разговариваем с детьми, которые отвлекают нас от «важных» дел.
2. *Слушать избирательно*, выхватывая из разговора только отдельные, интересные нам части.
3. *Внимательно слушать*, концентрируя внимание, сосредоточиваясь на произносимых словах.
4. *Эмпатическое слушание* позволяет посмотреть на вещи с позиции другого человека, проникнуть в его систему представлений. Благодаря этому вы можете представить себе мир таким, каким видит его другой человек, вы понимаете его парадигму (внутреннюю карту его мира), вы понимаете, что он чувствует.

Высший четвертый уровень - эмпатическое слушание – это не один из хитрых психологических «приемчиков», а *цель*, к которой нужно стремиться, чтобы добиться эффективности при взаимодействии. Его суть не в овладении каким-то техническим навыками или множеством правил, а во внутреннем стремлении (установке) как можно глубже и полнее понимать своего собеседника. Эмпатическое слушание означает понимание чувств, переживаемых другим человеком, и ответное выражение своего понимания этих чувств. Суть эмпатического слушания заключается не в том, что вы с кем-то соглашаетесь. Его суть в том, что вы

полностью, глубоко понимаете этого человека – как умом, так и сердцем. Большинство людей слушает не с намерением понять, а с намерением ответить. Эти люди постоянно проецируют свое собственное видение на поведение других, не ставя диагноз, прописывают свои собственные очки, которые им лично помогают уже 10 лет каждому, с кем взаимодействуют.

История 8. Эмпатическое слушание

Беседа менеджера (М) с программистом (П).

Сценарий 1. Ничто так не мешает эффективности коммуникации, как оценки и советы

П: «Я не понимаю, почему я должен следовать этой архитектуре? Кому она нужна!»

М: «Ты просто недооцениваешь важность концептуальной целостности системы». *(Оценка!)*

П: «Без всякой архитектуры я бы мог сделать это гораздо быстрее!»

М: «Ты ошибаешься. Если ты не будешь следовать архитектуре, то пожалеешь об этом, когда придется ее сопровождать. Следует больше доверять опыту нашего архитектора». *(Оценка, совет).*

Эта же беседа глазами программиста будет выглядеть следующим образом.

П: Я не понимаю, почему я должен следовать этой архитектуре? Кому она нужна! *(У меня проблемы и мне от этого плохо!)*

М: Ты недооцениваешь важность концептуальной целостности системы. *(Мне еще раз указали на то, что я плохой программист!)*

П: Без всякой архитектуры я бы мог сделать это гораздо быстрее. *(Иначе я не смогу справиться с заданием в срок!)*

М: Ты ошибаешься. Если ты не будешь следовать архитектуре, то пожалеешь об этом, когда придется ее сопровождать. Следует больше доверять опыту нашего архитектора. *(Со мной не считаются и не хотят понять!)*

На этом месте, как правило, собеседник «закрывается» и разговор заканчивается. Дальнейшее развитие событий может пойти по одному из трех одинаково неэффективных сценариев: 1) программист не уложится в срок и впадет в еще большее уныние; 2) станет работать сверхурочно – утомление, снижение эффективности; 3) отступит от архитектуры и сделает по своему – проблемы в будущем.

При эмпатическом слушании выделяются четыре стадии:

1-я стадия. Повторение содержания. Вы просто подтверждаете, что слушаете собеседника.

П: «Я не понимаю, почему я должен следовать этой архитектуре? Кому она нужна!»

М: «Ты не понимаешь, почему ты должен следовать архитектуре? Ты считаешь, что она никому не нужна?»

2-я стадия. Перефразирование содержания. Вы подтверждаете, что слушаете собеседника. Вы думаете о том, что он сказал и понимаете смысл того, что он говорит.

М: «Ты считаешь, что тебе не следует придерживаться архитектуры?»

3-я стадия. Вы отражаете чувства собеседника. Стараетесь понять, чувства, которыми руководствуется ваш партнер.

М: «Ты разочаровался в архитектуре нашей системы?»

4-я стадия. Совмещает вторую и третью.

М: «Ты разочаровался в архитектуре нашей системы? Ты находишь, что тебе следует отступить от нее?»

Если бы менеджер не просто слушал своего собеседника, а стремился как можно глубже и полнее понять его, та же самая беседа могла выглядеть по-другому.

Сценарий 2. Умение увидеть мир глазами другого человека, способность к сопереживанию

П: Я не понимаю, почему я должен следовать этой архитектуре? Кому она нужна!

М: «Ты разочаровался в архитектуре нашей системы? Ты Ты находишь, что тебе следует отступить от нее?» (П, мысленно: «Да это так, если не следовать архитектуре, то я смог бы справиться с задачей в срок!»)

П: «Без всякой архитектуры я бы мог сделать это гораздо быстрее!»

М: «Ты обеспокоен тем, что на решение задачи потратишь слишком много времени?» («Обеспокоен» – отражает чувства собеседника, «много времени» - перефразированное содержание полученного сообщения)

П: «Да. Я беспокоюсь о том, что я не смогу решить задачу в срок, поскольку она оказалась сложнее, чем мы предполагали при планировании. Я не знаю, как мне быть?» (Есть, «контакт» установлен, происходит «раскрытие» собеседника, вы получили право на совет.)

М: «Это не страшно. Так бывает. Хорошо, что ты вовремя поднял этот вопрос. У Васи Пупкина задание оказалось намного проще, чем мы оценивали ранее, и сейчас он освободился. Ты можешь привлечь его к решению своей задачи, и вам удастся уложиться в срок». (Все. Диагноз поставлен верно. Лечение назначено адекватное.)

Помимо учета индивидуальных особенностей людей, умения активно слушать, для достижения эффективности при коммуникациях так же необходимо:

- Учитывать историю взаимоотношений: накопленный уровень доверия, искренности, общности взглядов. С человеком, с которым вы работали не в одном проекте, достаточно, порой, обменяться взглядами, чтобы понять то, что каждый из вас хотел сказать в текущей ситуации. С новыми людьми необходимы разъяснения и терпение.
- Учитывать текущую ситуацию: обстановка в компании, состояние проекта, настроение и эмоции, как свои так и собеседника. Если собеседник озабочен важными личными проблемами, вряд ли с вашей стороны будет уместен вольный шуточный тон и приставания с незначительными вопросами.
- Учитывать степень формальности обстановки. Разговор «с глазу на глаз». Совещание с заказчиком, рабочее совещание, кафе.
- Всегда проявлять доброжелательность и уважительность межличностных отношений. Общение должно быть всегда на равных уровнях.
- Избегать модальных глаголов и повелительного наклонения. В переписке слова в верхнем регистре воспринимаются как повышенный тон.

Еще несколько рекомендаций, следование которым может повысить вашу эффективность при коммуникациях.

«Я-сообщения». Это стандартный инструмент эффективного общения, когда утверждение делается от собственного имени. Например: «Я считаю, что в нашем проекте проблемы, которые

необходимо срочно обсудить». Говоря от первого лица, вы берете на себя ответственность за свое мнение. Вы предоставляете собеседнику возможность самому решить, разделяет он это мнение или нет.

Используйте открытые вопросы. Это вопросы, которые дают собеседнику возможность подробно изложить свое мнение, а не просто подразумевают краткий ответ «да/нет». Пример: «Вася, расскажи, пожалуйста, что тебе удалось сделать по нашему проекту на этой неделе и с какими проблемами ты столкнулся?» Такой вопрос предоставляет Васе свободу в выборе формы изложения и позволяет высказать все, что он считает нужным. Если Вася пытается уклониться от обсуждения «неудобного» для него, но важного для вас аспекта своей работы, то вы всегда можете задать дополнительный вопрос: «Звучит отлично, но будь добр, расскажи мне более подробно как ты решал задачу интеграции систем А и Б?»

Следите за отклонениями от темы. Часто обсуждение может незаметно переходить с одной темы на другую или сразу на несколько, особенно, если ни одна из сторон коммуникации не следит за содержанием или целью дискуссии. Если мы хотим достичь цели, мы должны следить за рамками дискуссии и уметь вернуть ее в нужное русло. «Все это очень интересно, но, похоже, мы отклонились от темы. Давайте вернемся к ней, пока мы окончательно не забыли, зачем здесь собрались». Стоит, однако, заметить, что злоупотребление этим подходом может принести вред, поскольку вас могут посчитать педантом, который подавляет свободу, инициативу и творчество.

Если обсуждение начинает «буксовать». Цель не достигнута. Высказываемые суждения начинают повторяться. Многие откровенно заскучали. Следует попытаться взглянуть на проблему по-другому. Вовремя взглянуть на проблему с другой стороны - это всегда акт творчества. Демарко и Листер описали подобную ситуацию, которая возникла у них при работе над книгой [25]. При написании очередной главы, они предполагали дать 5-6 рецептов создания эффективных команд. Сколько они не старались, это у них никак не получалось. Ни одного! Тогда они взглянули на данную проблему с противоположной стороны: а как можно помешать формированию команды? И тут у них сразу нашлось множество рекомендаций. Правда, название главы пришлось изменить, глава стала называться «Травля команд».

Пресекать доминирование. Достаточно типична ситуация, когда один человек все время пытается доминировать на совещаниях. Активно навязывает свое мнение. Остальным членам команды это явно не нравится. Установите регламент. «Сегодня мы торопимся. Каждый говорит не более 3 минут. Высказываться будем все, по кругу».

Избегайте проекции. Проекция – это приписывание другим своих собственных мыслей и чувств. Никто не обязан разделять ваше личное мнение по какому-либо вопросу, думать как вы. Следует формулировать собственное мнение при помощи «я-сообщения», а не в виде констатации «объективной истины». И всегда надо искренне интересоваться мнением других.

Остерегайтесь смещения. Смещение происходит, когда сильная негативная эмоция или чувство, возникшие в одной ситуации, смещается на сотрудника, который ничем не заслужил такого обращения. «Попал под горячую руку». Всегда старайтесь контролировать свои чувства и эмоции. Прежде чем начать следующую коммуникацию, убедитесь, что вы владеете своими эмоциями. Либо, по крайней мере, информируйте собеседника о своих негативных чувствах, которые к нему не имеют никакого отношения. «Я разозлен после совещания с боссом, не принимай, пожалуйста, мою взвинченность на свой счет».

Бойтесь стереотипов. Стереотипы – это тенденция представления различных типов отношений в упрощенном виде. Категории: «Начальник», «подчиненный», «коллега», «заказчик», «большой босс». Без учета индивидуальных особенностей каждой конкретной личности эффективность коммуникаций сильно снижается. У вас создается впечатление, что вы все знаете наперед. Это ограничивает вашу способность слышать и сопереживать.

Проводите совещания эффективно. Совещания бывают разных типов. Статус митинги, «мозговые штурмы», информационные, агитационные и проч. Но если результатом собрания должны стать разработка и принятие решения по конкретным проблемам, то оно должно быть спланировано, подготовлено и подобающим образом организовано. (Источник – форум RSDN. Спасибо автору под ником bka1, более четкой, лаконичной и полной формулировки я не встречал.) Цитирую:

- Митинг (оно же собрание) надо планировать.

- Должна быть четкая повестка собрания (цель собрания, что обсуждается и в какое время).
- К собранию все должны готовиться заранее.
- У каждого участника должна быть своя роль, которой надо придерживаться. Роли должны быть известны заранее из повестки. Обязательные роли: модератор (ведет собрание) и секретарь (пишет протокол).
- Модератор ведет собрание по повестке и следит за тем, чтобы не уходило от темы.
- Если к собранию народ не подготовился, то такое собрание надо переносить.
- Все участники собрания обязаны подчиняться модератору. Особенно это касается ВИПов и прочих, которые норовят перехватить внимание на себя и пытаются по привычке рулить сами.
- В результате собрания должны быть приняты решения, которые естественно должны попасть в протокол.

Это список стоит дополнить рекомендациями от А. Орлова, данными им при обсуждении чернового варианта рукописи книги.

«Из моей практики в Интеле. Если какой-то вопрос не может быть выяснен прямо на митинге, то человеку дается AR («Action Required»), который фиксируется в протоколе. AR состоит из трех частей: Who, What, When. Следующий митинг (если речь идет о, скажем, project status митингах) начинается с проверки AR'ов предыдущего митинга.

На совещаниях есть еще одна распространенная проблема, это когда менеджер спрашивает «Все согласны?», то те, кто не согласны, часто молчат. А потом не то, что саботируют, но работают без особой поддержки идеи.

Хорошая штука, которую я услышал на тренинге Джима Маккарти (автор книги «Программируем командный дух») – это правило большого пальца. То есть, человек, предлагающий идею, спрашивает «Как вам идея? Раз-два-три». На счет «три» каждый должен:

- либо показать большой палец вверх («поддерживаю»);
- либо большой палец вниз («идея не нравится») – тогда он должен объяснить почему;
- либо большой палец параллельно земле («не против, но есть сомнения») – тогда он также должен объясниться, что его смущает.

Я это, естественно, пробовал на практике. Работает».

Конфликты

История 9. «Группомыслие»

Совместная работа группы характеризуется следующими особенностями:

Девиз группы: «Давайте работать, а не конфликтовать!»

Все участники стараются избегать конфликтов и поддерживать согласие.

Как правило, никто не спорит, все соглашаются с мнением руководителя и следуют его указаниям.

При возникновении трудных ситуаций, все ждут решения от руководителя.

Редкие противоречия разрешаются общим голосованием.

Диагноз. В психологии подобная ситуация называется «Парадокс Абилина» [26]. Люди принимают решения, основанные не на том, что они сами хотят, но на том, что они думают, что другие хотят. В результате получается, что каждый делает что-то, что никому на самом деле не нужно. Подобное избегание производственных конфликтов снижает здоровую интеллектуальную конкуренцию, ведет к шаблонности и застою.

Рекомендации. В этом случае следует целенаправленно мотивировать инакомыслие и интеллектуальную конкуренцию в команде. Например, назначать рецензентов – критиков, которые должны найти слабые стороны в решении, предлагаемом их коллегой для общего обсуждения. Или поручить двум разработчикам решить одну и ту же критичную для проекта задачу, используя разные подходы или технологии, а затем поручить сравнить и оценить полученные результаты третьему коллеге.

Производственные конфликты – это источник развития. Интеллектуальная конкуренция заставляет членов команды работать, максимально напрягая свои способности. Конфликты – это профилактика шаблонов, возможность нестандартных эффективных решений. Конфликты позволяют производить более «креативный» продукт. И, наконец, без конфликтов не бывает настоящих команд. Именно конфликты обеспечивают большую сплоченность команды, и ее эффективность в достижении конечной цели.

Оговорюсь сразу, речь в данном разделе идет исключительно о *производственных* конфликтах, а не о личностных. Личным конфликтам не может быть места в эффективных командах. Люди, которые служат источниками межличностных проблем, должны быть удалены или, хотя бы, изолированы от других участников команды. Мало, что действует так разрушительно на эффективность программистских команд, как «одно гнилое яблоко в корзине».

Конфликт – столкновение противоречащих интересов, целей, желаний людей в ходе их взаимодействия. Необходимо понимать и всегда помнить, что у каждого конфликта есть три составляющих компонента: 1) объект - предмет конфликта; повод - внешняя декларируемая позиция каждой из сторон и 3) причина – внутренняя позиция каждой из сторон.

Для эффективного разрешения конфликта на основе сотрудничества, в духе «выиграл/выиграл» рекомендуется применять следующую последовательность шагов:

- Признать, что конфликт есть.
- Отделить проблему от людей: конкурируют идеи, а не люди.
- Договориться об общем, что принимается обеими сторонами, - формулировка проблемы, разделяемые цели.
- Сформулировать видение проблемы каждой из сторон.
- Собрать как можно больше объективных данных о ситуации.
- Выдвинуть и рассмотреть максимум альтернативных решений.
- Выбрать оптимальное решение, взаимовыгодное для всех сторон.
- Проинформировать о решении всех участников проекта, которых оно касается.

Важно постоянно помнить, что вы по одну сторону баррикады, а проблема – с другой.

Однако, далеко не всегда ситуация складывается таким образом, что возможно разрешение конфликта в наиболее эффективной манере. Например, если сторонам не удастся найти взаимовыгодное решение, то возможно пойти по пути взаимных уступок, чтобы получить компромиссное решение в стиле «проиграл/проиграл». Это решение безусловно менее эффективно, чем решение в духе «выиграл/выиграл», однако в жизни встречается часто, когда время на разрешение конфликта ограничено.

Возможен и другой подход в духе «выиграл/проиграл», если ситуация требует незамедлительного разрешения проблемы. Например, в случае возникновения критической ошибки в системе, установленной у клиента, из-за которой он каждую минуту несет существенные материальные потери. В этой ситуации необходимо, чтобы лидер взял решение конфликта и ответственность на себя и действовал в директивном стиле «выиграл/проиграл». Еще раз подчеркну, что

авторитарный стиль разрешения конфликтов применим только в виде исключения в критических ситуациях, поскольку может разрушительно воздействовать на доверие команды к лидеру.

Следующий возможный подход в разрешение конфликта – это приспособление, действие в духе «проиграл/выиграл». Например, пожертвовать своими интересами, ради интересов оппонента или согласиться с начальником, когда нет возможности склонить его на свою сторону. Следует, однако, помнить, что частое употребление этого подхода может ослабить ваш авторитет.

И последняя рекомендация по способу разрешению конфликтов – это уклонение. Можно просто отложить разрешение проблемы. Если позволяет время, можно подождать в надежде на то, что со временем конфликт станет неактуальным или позиции сторон сблизятся, или станет доступна дополнительная информация, которая позволит решить проблему более эффективно. Противопоказания - со временем конфликт может углубиться и обостриться.

Можно еще долго рассказывать о примерах эффективного и неэффективного взаимодействия, но это все равно, что теоретически обучать езде на велосипеде. Вы никогда не научитесь, пока не сядете в седло и не начнете крутить педали и руль. Еще раз напомню, образно говоря, моя цель показать, что «велосипед существует и на нем можно быстро ездить». И еще одно замечание. Каждый имеет то, к чему осознанно или неосознанно стремится. Изменить что-то в своем окружении можно, лишь изменив себя. Бессмысленно овладевать навыками «эмпатического слушания», не развив в себе умение увидеть мир глазами другого, способность к сопереживанию и взаимопомощи.

Глава 4. Руководство командами

Группа и команда

В одной из своих последних книг [27] Том Демарко пишет о том, что есть только четыре основных правила менеджмента программных проектов:

1. Найти нужных людей.
2. Дать им ту работу, для которой они лучше всего подходят.
3. Не забывать о мотивации.
4. Помогать им сплотиться в одну команду и работать так дальше.

Все остальное — административная ерундистика.

Идеи командного менеджмента на Западе зародилась в начале 80-х годов. Эффективность команд в новых экономических условиях одними из первых оценили такие гиганты, как Procter & Gamble и Boeing [28]. Энтузиазм некоторых молодых коллег, по поводу приходящих к нам с Запада разнообразных гибких методологий в разработке ПО, скорее всего от недостатка информации и опыта. Суть всех этих подходов: Scrum, xProgramming, Crystal, ASD, FDD и прочих, - состоит в том, что они декларируют своей высшей ценностью *командную работу*, ориентированность на людей и их взаимодействие, а не на процессы и средства. На мой взгляд, эти методологии похожи на попытки «переоткрытия» представителями индивидуалистской западной культуры принципов коллективизма, хорошо зарекомендовавших себя в России с прежних времен. Один из основателей западной гуманистической психологии, идеи которой положены в основу гибких методологий, В. Франкл писал, что на него огромное влияние оказали сочинения именно русских писателей таких, как Ф. Достоевский и Л. Толстой, которых он постоянно цитировал в своих трудах.

И совсем нелепо, на мой взгляд, выглядят на российской земле разнообразные тренинги по формированию команд, импортированные из-за рубежа. «Ropes Courses» (веревочные курсы) [29], своим названием обязаны альпинистскому снаряжению, которое использовалось в этих тренингах. Проводились они, как правило, на свежем воздухе, начинались с предтренинговой экспертизы, «разогрева», затем следовал ряд несложных упражнений, выполняемых на земле, а после -

кульминация: «экзерсисы» на высоте и свободное падение на руки сотоварищей и коллег и т. д. А как еще можно научиться доверию?!

Если вспомнить историю, то для российского менталитета командная работа достаточно привычный вид коллективной деятельности. С XIII века в России существовали производственные артели - различные формы добровольных объединений людей с целью осуществления общей хозяйственной деятельности. Например, артели плотников, лесорубов, рыбаков, земледельцев и др. Русская рабочая артель являлась одним из устоев народной жизни [30]. Она была добровольным товариществом совершенно равноправных работников, призванных на основе взаимопомощи и взаимовыручки решать практически любые хозяйственные и производственные задачи. Равноправие артели резко отличались от капиталистических предприятий. Равноправие, конечно, не означало уравниловки - распределение дохода осуществлялось по труду.

Истинный российский коллективизм¹ не имеет ничего общего с вульгарным коллективизмом: со стадностью, подавлением личности («я нет, есть только мы!», «я – последняя буква алфавита!»), уступчивостью, групповой психологией и слепым подчинением меньшинства большинству. *Суть российского коллективизма в том, что человек ощущает себя элементом органичной системы, где он выполняет свою функцию, свою задачу, совершенно особенную, которую не выполняет никто другой и выполнить не может. Эту задачу он выполняет совершенно сознательно, стремясь к максимальной реализации своих целей.* Объединение людей в артель не только не ограничивало дух самостоятельности и предприимчивости каждого артельщика, а, напротив, поощряло его. Артель позволяла сочетать склонность русского человека к самостоятельному и даже обособленному труду с коллективными усилиями.

Конформизм и соглашательство, скорее присущ западному менталитету [31]. На пути к индивидуальному успеху западный человек должен научиться делать многое из того, что противоречит его взглядам и убеждениям, должен уметь приспосабливаться к мнению людей, от которых зависит его продвижение по социальной лестнице. Часто западный человек учитывает мнение других на всякий случай - никогда нельзя сказать заранее, окажется ли впервые увиденный человек членом какого-либо полезного сообщества. Отсюда и склонность подчиняться авторитету группы и менять свое мнение в зависимости от мнения других.

Исследователь Соломон Эш поставил такой эксперимент. Испытуемым предлагали сравнить длину двух линий. Перед этим несколько "подсадных" давали в их присутствии неправильные ответы. Выяснилось, что американца довольно легко заставить не поверить своим глазам. Если четыре-пять человек уверенно делали заведомо неверное утверждение, 68% американцев с ним соглашались. Среди европейцев процент конформистов оказался чуть ниже, но тем не менее был достаточно высок -- от 40 до 60% в разных странах. В культурах, которые традиционно считаются более коллективистскими, например, русские, вдвое реже, чем американцам, линии разной длины кажутся одинаковыми, если так считает коллектив. В Японии конформистов было всего 20%, а среди эскимосов Канады таких не нашлось вообще.

Известный российский публицист и политик, А.И. Герцен видел в коллективизме, в уникальности существующей сельской общины, в городской артели и в самобытной воинской организации казачества характерную черту жизни и психологии русского народа. Когда наши предки говорили, что «на миру и смерть красна», то имелась в виду именно крестьянская община села, та ограниченная группа, к которой они принадлежали, а не «вся связанная совокупность множественного бытия».

Позднее во времена СССР широкое распространение на производстве получили рабочие бригады, а в научно-исследовательских институтах - временные научные творческие коллективы, которые объединялись на основе общности цели, взаимопомощи и коллективной ответственности за результат. В чем их преимущество перед строгими многоступенчатыми иерархическими организационными структурами я имел возможность изучить на собственном опыте. Эффективная команда предполагает самоорганизацию и самоуправление совместной деятельностью, взаимный контроль, взаимопомощь и взаимозаменяемость, ясность общих ценностей и целей, коллективную

¹ Если следовать русской философской традиции, например, А.С.Хомяков, С.Н. Булгаков, П.А. Флоренский, то более точно было бы употребить здесь не «истинный коллективизм», а непереводаемый на иностранные языки термин «соборность».

ответственность за результаты труда, всемерное развитие и использование индивидуального и группового потенциалов.

О признании на западе важности коллективизма и командообразования в разработке ПО так же свидетельствуют и последние труды Института технологий разработки программного обеспечения SEI. В частности, «Командный процесс разработки программного обеспечения (Team Software Process, TSP)» [32] определяет, что «основная задача руководителя группы состоит в том, чтобы поддержать мотивацию и целеустремленность команды, направленные на достижение наивысшей эффективности выполнения работ».

Следует различать группу людей, формально объединенных для выполнения связанных работ, и команду, способную к самоорганизации, самоуправлению и развитию (Таблица 11).

Таблица 11. Группа и команда

	Группа	Команда
Цель	Ставится узкая задача, общие цели не проясняются.	Ясные, приняты всеми цели и стратегия их достижения.
Участие в работе	Выполнение должностных инструкций и распоряжений.	Активная позиция, нацеленность на результат, личная ответственность.
Ролевая структура	Строгое распределение ролей, должностей, обязанностей.	Разделение компетенций. Гибкая структура. Ротация ролей.
Руководство	Администрирование, наличие формального лидера-начальника.	Лидерство на основе компетентности и доверия, наставничество, помощь и поддержка.
Принятие решений	В основном приказы, и решения принятые большинством.	Эффективные процедуры принятия решений на основе доверия и взаимовыгоды.
Конфликты	Замалчивание, сокрытие, игнорирование.	Признание, интеллектуальная конкуренция, эффективное разрешение: «мы по одну сторону баррикады, а проблема – с другой».
Взаимодействие	Закрытость, избегание критики, принцип «не высываться».	Доверие, свобода, проявление инициативы.
Коммуникация	Через формального лидера служебная переписка.	Открытость. Уверенность друг в друге и взаимное уважение.
Творчество	Стереотипность, работа по правилам.	Гибкость и адаптивность. Непрерывное совершенствование и рост компетенций. Раскрытие творческого потенциала каждого.

Разные работы требуют разную форму организации трудовых коллективов. Совсем различные процессы, способы управления и организация требуется при соревнованиях команд на байдарках в спокойном гребном канале и при сплаве на рафте по неизведанным горным порожистым рекам.

Когда внешние условия хорошо известны и стабильны, когда производственные операции хорошо изучены и неоднократно испытаны, функции исполнителей определены и постоянны, нет места самоорганизующимся командам, они могут принести только сбой в хорошо отлаженный механизм. В этом случае основой эффективности служат узкая специализация и повышение компетенции. Если водитель трамвая начнет искать новые пути, жди беды.

Самоорганизующиеся и саморазвивающиеся команды требуются там, где разрабатывается новый продукт, внешние условия и требования к которому постоянно меняются, где применяемые производственные технологии используются впервые, где постоянно требуются поиск новых возможностей, интеллектуальные усилия и творчество.

Командные роли

Особенности поведения людей бесконечны, однако, диапазон полезных моделей поведения в работе команды достаточно ограничен. Специалисты в области оценки и развития команд [19] утверждают, что существует лишь девять командных ролей (Таблица 12), баланс которых является решающим фактором успеха или неудачи в командной работе.

Таблица 12. Типы командных ролей

Тип	Достоинства	Допустимые недостатки
Генератор идей	Креативен, решает сложные проблемы.	Отстаивает право собственности на идею, пренебрегает сотрудничеством.
Исследователь ресурсов	Он легко общается с людьми. Если нужно провести переговоры, лучшего кандидата не найти.	Быстро «угасает», теряя интерес к начатому делу.
Координатор	Умеет задействовать ресурсы группы, обобщает. Ставит цели. Руководит.	Может присваивать достижения команды.
Мотиватор (шейпер)	Бросает вызов, подталкивает к действиям. Обладает напором и мужеством преодоления препятствий.	Не сдержан в выражениях и не способен исправить ситуацию с помощью юмора или извинения.
Аналитик	Дает взвешенную оценку идей.	Внушает пессимизм, циничен.
Вдохновитель команды	Устраняет трения в команде, сплочивает. Умеет слушать других.	Нерешительность. Избегает конфликтных ситуаций.
Реализатор	Эффективен, организован, практичен. Претворяет идеи в практические дела.	Оказывает сопротивление нововведениям.
Контролер (педант)	Добросовестно совершенствует. Ищет ошибки и упущения.	Навязчивое стремление к достижению совершенства во всем.
Специалист	Увлечен и высокопрофессионален.	Игнорирует факты вне области своей компетентности.

Следует отметить, что некоторые типы командных ролей совпадают по названию с псевдонимами личностных типов по В.Гуленко (Таблица 4), однако, это совсем не одно и то же. Тип личности определяет преимущественный способ взаимодействия с окружающим миром, направленность психологической энергии, предпочтения при сборе информации и принятии решений. Роли по Белбину описывают типовые модели конструктивного поведения людей в командах. Психологи не усматривают прямой корреляции между предпочтением человека исполнять определенную командную роль и его типом личности [33]. По моим личным наблюдениям, роль, которую человек принимает на себя, часто зависит от состава и состояния команды. В одной команде человек может выступать в роли генератора идей, а в другой - в качестве аналитика или реализатора.

Тем более следует различать «командные роли» и «функциональные роли». Люди, выполняющие одни и те же должностные функции в проекте, например, тестирование, могут исполнять разные командные роли. Ниже приведено более подробное описание каждой из командных ролей (Таблица 13).

Таблица 13. Описание типов командных ролей

Описание	Кредо
<i>Генератор идей</i>	
<p>Оригинальный мыслитель, который дает жизнь новым идеям. Независимый сотрудник с развитым воображением, но подобно остальным людям имеет негативные черты характера – может быть чрезмерно чувствителен к критике. Для успеха генератору идей необходимы конструктивные отношения с руководителем или координатором группы.</p>	<p>Когда проблема ставит вас в тупик, попробуйте ее проанализировать со всех сторон. Там, где есть проблема, найдется и решение. Чем, больше проблема, тем больше вызов. Не беспокоить, гений работает. Хорошие идеи всегда сначала кажутся странными. Идеи рождаются в мечтах. Без постоянного новаторства невозможно движение вперед.</p>
<i>Исследователь ресурсов</i>	
<p>Так же, как и генератор идей, в состоянии привнести новые идеи в группу, но эти идеи будут заимствованы извне, благодаря широким контактам. Несколько бесцеремонный, гибкий и ищет благоприятные возможности. Обычно разговаривает по телефону или находится где-нибудь на встрече. Не дает развиваться групповой инертности. К отрицательным качествам характера относятся лень, самодовольство и, иногда, требуется кризис или давление обстоятельств, чтобы мотивировать его.</p>	<p>Мы могли бы сделать на этом состоянии. Чужие идеи необходимо воровать с гордостью. Никогда не изобретай заново колесо. Возможности появляются в результате ошибок других людей. Вы уверены, что мы не можем воспользоваться этим? Всегда можно позвонить, чтобы выяснить необходимый вопрос. Потраченное на разведку время редко потрачено зря.</p>

Координатор	
<p>Обычно формальный лидер группы. Руководит и направляет группу в сторону достижения целей. Может заранее определить, кто из работников хорош для выполнения необходимых задач. Обычно спокойный, уверенный и распорядительный. Однако иногда склонен к излишнему доминированию, и группа становится продолжением его сильного «Я».</p>	<p>Давайте всегда помнить об основной цели.</p> <p>Если у кого-то есть что-нибудь, что можно добавить к этому?</p> <p>Нам необходимо достигнуть консенсуса прежде, чем мы пройдем дальше.</p> <p>Никогда не думайте, что молчание означает согласие.</p> <p>Я думаю, что мы можем дать шанс кому-то еще.</p> <p>Умение правильно делегировать полномочия - это искусство.</p> <p>Менеджмент - это умение организовать людей для выполнения любой работы.</p>
Мотиватор	
<p>Энергичный и в состоянии внедрять идеи. Видит мир как проект, который требует внедрения. Обычно уверенный, динамичный, эмоциональный и импульсивный. Мотор группы, но может быть раздражительным, несдержанным, нелюбезным.</p>	<p>Просто сделай это!</p> <p>Сначала скажи «нет», а потом вступай в переговоры.</p> <p>Если вы говорите «да, Я это сделаю», я считаю, что так оно и будет.</p> <p>Меня не удовлетворяет то, что мы добиваемся всего, что можем.</p> <p>Я могу быть резким, но по крайней мере я прав.</p> <p>Я заставлю всех работать.</p> <p>Когда продвижение к цели становится все труднее, за дела берутся крутые.</p>
Аналитик	
<p>В предыдущем издании у Белбина, на мой взгляд, было более точное название для этой роли - Критик</p>	
<p>Оценивает предложения и занимает позицию наблюдателя за продвижением. Не дает группе двигаться неправильным путем. Осмотрительный, бесстрастный, имеет аналитический склад ума. Может казаться равнодушным, незаинтересованным, иногда становится чрезмерно критичным.</p>	<p>Все ли возможности мы использовали?</p> <p>Если это не логично, то этим и не стоит заниматься!</p> <p>Лучше долго принимать правильное решение, чем быстро принять неправильное.</p> <p>Кажется, это лучший вариант при данном</p>

	<p>соотношении сил.</p> <p>Давайте взвесим все альтернативы.</p> <p>Решения не должны основываться на энтузиазме.</p>
<i>Вдохновитель команды</i>	
<p>Стремится объединять и вносить гармонию в отношения между членами группы. Занимает позицию понимающего чужие проблемы, стремится помочь и сглаживает конфликты. По натуре человек добрый, стремится налаживать неформальные отношения. Однако бывает нерешительным в сложных или кризисных ситуациях.</p>	<p>Вежливость мне ничего не стоит.</p> <p>Меня очень заинтересовала ваша точка зрения.</p> <p>Если вас это устраивает, то устраивает и меня.</p> <p>У каждого человека есть хорошие качества, к которым можно взывать.</p> <p>Чем люди больше слушают, тем меньше говорят.</p> <p>Вы всегда можете почувствовать хорошую рабочую атмосферу.</p> <p>Я стараюсь быть гибким.</p>
<i>Реализатор</i>	
<p>Может преобразовать стратегический план в конкретные управленческие задачи, которые доступны для решения. Хороший организатор, методичный и прагматичный. Идентифицируется с группой, лояльный и честный сотрудник. Однако может быть негибким, непреклонным.</p>	<p>Однако может быть негибким, непреклонным.</p> <p>Если это возможно, мы сделаем это.</p> <p>День практики стоит года теории.</p> <p>От напряженной работы еще никто не умер.</p> <p>Если это трудно сделать, мы сделаем это немедленно. Если это невозможно сделать, это займет чуть больше времени.</p> <p>Человеку свойственно ошибаться, но компании не свойственно прощать ошибки.</p> <p>Давайте возьмемся за ближайшую задачу.</p> <p>Я полностью поддерживаю политику компании.</p>
<i>Контролер</i>	
<p>Отлично умеет создавать отчеты о работе группы. Озабочен точным выполнением взятых обязательств и старается не упускать из виду даже мелких деталей. Заставляет придерживаться точного расписания дел, но может становиться излишне тревожным.</p>	<p>Это дело требует нашего пристального внимания.</p> <p>Надписи, сделанные мелким шрифтом, всегда нужно читать.</p> <p>«Если чему-то суждено не сбыться, то пусть так»</p>

	<p>оно и будет», и как О’Тул сказал о законе Мэрфи, «Мэрфи был оптимистом».</p> <p>Нет оправдания тому, что вы не можете стать самым лучшим.</p> <p>Совершенство это то, что сделано достаточно хорошо.</p> <p>Один стежок, но вовремя, стоит девяти.</p> <p>Это кто-нибудь проверял?</p>
Специалист	
<p>Профессионал, самостоятелен стремится стать экспертом в своей области. Обладает высокой профессиональной/технической экспертизой и знаниями, гордится своей работой. Приносит вклад только в узкой сфере своей профессиональной экспертизы.</p>	<p>На этой работе вы никогда не перестанете учиться.</p> <p>Выберите работу себе по душе, и вам никогда не придется работать.</p> <p>Истинный профессионализм - это собственная награда.</p> <p>Моя работа приводит меня в восторг.</p> <p>Чем больше вы знаете, тем больше вам предстоит узнать.</p> <p>Лучше знать много об одном, чем мало обо всем.</p> <p>Комитет – это двенадцать человек, которые делают работу одного человека.</p>

Ошибка подбирать в команду подобных себе людей. Правильно подобранные «лебедь, рак и щука», как правило, оказываются гораздо успешнее «родственных душ». *Единообразие – это единомыслие и стагнация.*

В списке командных ролей отсутствует роль «лидер». В разное время руководитель команды может и должен брать на себя ту роль, которая позволит сделать команду более сбалансированной. Выступить генератором идей, когда команда стала буксовать в решении проблемы, стать придирчивым критиком-аналитиком, если ощущается переизбыток идей, энтузиазма и оптимизма, или принять на себя роль вдохновителя, когда участники команды приуныли. Способность выполнять разные командные роли, помогать другим людям выполнить их задачи как можно лучше - это одно из обязательных качеств эффективного лидера.

Этапы формирования команды

Группа людей, объединенных для выполнения работы, сама по себе не становится самоуправляемой командой. Становление команды это процесс, включающий в себя последовательное прохождение четырех четко выраженных этапов [34].

1. Объединение.
2. Разногласия и конфликты.
3. Становление.

4. Отдача.

Замечание А. Орлова «В англоязычной литературе названия этих этапов звучат более выразительно: 1) Forming, 2) Storming, 3) Norming, 4) Performing. Кстати, иногда добавляют еще пятый этап Reforming, который возвращает команду из последней стадии в начало :»).

Объединение

Этот этап характеризуется, как правило, избытком энтузиазма, связанного с новизной задач. Новизна сама по себе может, как правило, выступать на первых порах значимым мотивирующим фактором. На участников действует так называемый эффект Готорна – прирост энергии и интерес, наполняющий людей, когда они осваивают что-то новое и непривычное [25].

Люди, объединенные в рабочую группу, имеют различные мотивы и ожидания. Важно понимать, в чем будет выигрыш каждого участника в случае общего успеха проекта, и использовать это знание для сплочения сотрудников. Более подробно эта тема будет рассмотрена в главе 6. Мотивация.

Кроме того, в новый коллектив каждый привносит свою «социальную схему», которая представляет собой личные взгляды на то, как должна функционировать команда. Участники группы должны преодолеть внутренние противоречия, пройти через конфликты прежде, чем сформируется действительно спаянный коллектив. На этом этапе многое зависит от лидера: он должен сформировать общеконандное видение проекта. Все участники группы должны четко понимать не только *что* именно они будут делать, но и *почему* они будут это делать.

Разногласия и конфликты

Самый сложный и опасный, однако, неизбежный период в становлении команды. Мотивация новизны уже ослабла, а сильные и глубокие внутренние стимулы у команды еще не появились. Каждый участник пытается установить и отстоять свою роль в проекте. На этом этапе возможны соперничество, споры, оборонительная позиция. Неизбежные сложности или неудачи порождают конфликты, «поиск виновных».

Лидеру на этом этапе важно обеспечить открытую коммуникацию в команде – конфликты не следует прятать или разрубать. Споры необходимо разруливать – спокойно, терпеливо и тщательно. Не стоит навязывать свои решения группе, даже если они кажутся очевидными, а следует помогать участникам команды самим приходиться к нужным решениям. Не все верные решения могут быть эффективными. Решение будет работать только тогда, когда есть люди, которые хотят его выполнять. Каждый человек мотивирован участвовать в принятии решения по поводу собственных проблем и готов принимать на себя обязательства по выполнению совместно принятых решений.

После «закалки» команды на этапе разногласий и конфликтов в коллективе начинают вырабатываться наиболее оптимальные методы взаимодействия, общения и совместной работы.

Становление

На этапе «*Становление*» в команде укрепляется доверие, люди начинают замечать в коллегах не только проблемные, но и сильные стороны. На смену битве амбиций приходит продуктивное сотрудничество. Четче становится разделение труда, исчезает дублирование функций. Лидер перестает находиться в состоянии постоянного аврала, работа по построению команды на этом этапе – уже не тушение пожара, а *скрупулезный труд по отработке общих норм и правил*.

Поскольку, разработка ПО, как отмечалось выше, - корпоративная игра, то первое, что мы должны сделать, это договориться о *правилах игры* – нормах и регламентах, которые определяют права и ответственность участников команды в проекте. Повторим вслед за автором [35] каждому проекту свои процессы. И чем больше и ответственней проект, тем «тяжелей» должна быть технология. Обязательно должны быть сформулированы и внедрены механизмы эффективного пересмотра правил, которые утратили свою актуальность или эффективность. Технология должна изменяться вместе с развитием и ростом проекта, но на каждом этапе она должна быть определена и описана.

Опасность на этом этапе состоит в том, чтобы избежать чрезмерной бюрократии устанавливаемых правил и процедур. Стандарты должны разрабатываться только для решения важных и актуальных проблем, с которыми уже столкнулась команда, а не на все случаи жизни. Регламенты должны давать основные принципы, которыми должны руководствоваться участники проекта, а не подробные правила, которые требуется постоянно соблюдать. Стандарты и регламенты должны облегчать работу участников команды, а не затруднять ее, они – система маяков, которые предостерегают программный проект от того, чтобы он не оказался на мели.

Отдача

Когда наступает этот этап, то, наконец-то, можно приступить к получению дивидендов за потраченные усилия. Команда работает эффективно, высок командный дух, люди хорошо знают друг друга и умеют использовать сильные стороны коллег. Высок уровень доверия. Это лучший период для раскрытия индивидуальных талантов. Люди хотят и могут совершенствоваться, они более всего заинтересованы в профессиональном росте. Растет значение нематериальной мотивации сотрудников, а оценивать и поощрять материально лучше команду в целом.

На этом этапе лидер использует стратегию «делегирования». Руководитель поддерживает на необходимом уровне мотивацию участников команды, следит за качеством их работы. Основное внимание руководителя сосредотачивается на делах из второго квадранта – «точим пилу» (см. параграф «Управляем своей жизнью»). Настоящий лидер работает на опережение. Он внимательно следит за изменениями в команде, окружении, целях и задачах проекта – предвидит и избегает риски или снижает их возможные воздействия на проект. Постоянно наблюдает и оценивает эффективность всех процессов, применяемых в проекте. «Что лишнее мы делаем?» «Что можно делать проще?» «Что угрожает проекту?». Работает на сокращение ненужных усилий вместо того, чтобы «стремиться к новым героическим победам». Определяет узкие места и применяет корректирующие действия там, где процессы начинают буксовать или риски слишком велики. Важно: *не команда должна приспосабливаться к процессам, а процессы должны подстраиваться под команду по мере ее развития и становления.*

Расформирование

Мы рассмотрели этапы *формирования* команды, но следует также сказать несколько слов и о *расформировании* команды.

Если в ходе проекта команда так и не сложилась, то расформирование группы, в основном, управленческий аспект работы руководителя. Главная задача на этом этапе - проверить и передать Заказчику результаты проекта и аккумулировать полученный опыт. Для этого, как правило, достаточно выполнить следующие работы:

- Провести приемо-сдаточные испытания продукта.
- Закрывать контракты с заказчиком.
- Завершить все работы по проекту. Не оставить «зависших» работ.
- Освободить ресурсы и проинформировать об этом всех линейных руководителей.
- Отчитаться перед руководством своей компании и получить его оценку исполнения проекта, в том числе и представленную в материальном выражении.

Важно, не забыть об «обратной связи» без которой, не может устойчиво управляться ни одна система, а именно, необходимо документировать уроки проекта – возникшие проблемы и способы их решения. Том Демарко утверждает [36], что риски, которые мы должны в первую очередь анализировать в новом проекте, это проблемы проектов вчерашних. В итоговом отчете по проекту целесообразно помимо извлеченных уроков высказать рекомендации по совершенствованию производственных технологий, по применению в компании приобретенного в проекте опыта, а, возможно, и по повторному использованию программных компонентов наработанных в проекте.

Несколько по-другому обстоит дело, если при работе в проекте удалось сформироваться эффективной команде, способной к новым достижениям. Оптимальное решение - заранее назначить команду в полном, а лучше расширенном, составе на новый более сложный и интересный проект. Это позволит компании аккумулировать приобретенный человеческий капитал. Такая перспектива, кстати, послужит для участников команды отличным стимулом для скорейшего завершения текущего проекта. Другое, возможное решение, тоже, на мой взгляд, весьма эффективное и полезное, это разделить команду на два новых проекта. Это позволит в этих новых проектах создать основу, на базе которой с высокой степенью вероятности кристаллизуются две новые успешные команды.

Крайне нежелательно распыление команды по другим проектам. Это – растрата человеческого потенциала, который накопился в проекте. У членов сплоченной команды подобное решение может вызвать разочарование и фрустрацию. Значительные трудности, связанные с переменами, утратой комфорта межличностного общения, необходимость расставаться с товарищами, могут отрицательно сказаться на мотивации команды при завершении проекта. Более того, знаю случаи, когда команды практически в полном составе, сразу или постепенно, уходили из компании, предпочитая сохранение сплоченного и эффективного коллектива лояльности компании.

Лидерство и управление

Лидерство (leadership) и управление (management) не следует противопоставлять друг другу. Это две составляющих одного и того же процесса. Лидерство – это высшее проявление менеджмента. Современные менеджеры должны быть и управленцами, и лидерами. Лидерство и управление одинаково важны, они не могут существовать в отрыве друг от друга. Нельзя быть лидером запасов, денежных потоков и затрат. Ими необходимо управлять. Потому что у вещей нет права и свободы выбора, которые присущи только человеку. С людьми нужно быть эффективными, а производительными – с вещами. Нельзя ориентироваться на производительность в отношениях с людьми. Интеллектуальными людьми невозможно управлять. Творческие команды можно только направлять и вести.

Управление – это расчленение, анализ, определение последовательности действий, конкретная реализация. Управление фокусируется на нижнем уровне: как мне сделать это наилучшим образом? Эта компетенция руководителя определяет эффективность движения по выбранному пути. Исполняя эту роль руководитель:

- Устанавливает проектные процедуры.
- Определяет организационную структуру и отчетность.
- Готовит планы и расписания работ.
- Контролирует ход работ и корректирует недопустимые отклонения.
- Управляет рисками проекта.

Для понимания управленческой стороны в работе руководителя проекта могу рекомендовать американский стандарт [37], который описывает 44 процесса, распределенные по 9 областям знаний в управлении проектами. Для людей с техническим образованием выполнение роли управленца, как правило не вызывает особых затруднений.

Однако в инновационных проектах, с высокой степенью неопределенности и подверженных высоким рискам просто управления недостаточно. «Высокопроизводительное управление в отсутствие эффективного лидерства подобно упорядочению расстановки стульев на палубе тонущего «Титаника». Никакой успех в управлении не компенсирует провала в лидерстве. ...Лидерство – это высшее проявление менеджмента, это создание для людей перспективы и высвобождение их потенциала» [3]. Лидерство – это, прежде всего, энергичная деятельность правого полушария мозга. Оно сродни искусству и философии. Когда вы имеете дело с лидерством, вы должны задавать себе основные, главные вопросы бытия.

Тема лидерства необычайно актуальна и поэтому сопровождается большим числом спекуляций. Например [18], в англоязычной литературе за последнее время издано более 9 тыс. книг по вопросам лидерства. Вот названия некоторых из них:

- «Иисус – гендиректор».
- «Уроки лидерства от Иисуса».
- «Если бы Аристотель управлял General Motors».
- «Менеджер мафии».
- «Дао личного лидерства».
- «Быстрый путь к лидерству».
- «10-минутное руководство по эффективному лидерству».
- «Винни Пух о менеджменте».
- и т.д.

Манфред Кетс де Врис утверждает, что «больше всех знают о лидерстве парикмахеры и таксисты».

Роли и стратегии лидера

Лидер – это, прежде всего состоявшаяся личность. Слабый, зависимый, не самостоятельный человек, который не управляет собой, не стал лидером для самого себя, не может быть лидером для команды. Просто самостоятельные и независимые люди, не достаточно зрелы для того, чтобы думать и действовать взаимозависимо. Поэтому они могут хорошо работать индивидуально, но не могут быть эффективными лидерами или командными игроками. Суть лидерства - взаимодействие и синергия. Думать и действовать в духе «выиграл/выиграл» - это настоящее искусство, позволяющее лидерам открывать новые возможности, то, что до них не существовало. Эффективное лидерство это не столько работа с людьми, сколько работа ради людей.

Не бывает лидеров без приверженцев и единомышленников. Для того чтобы стать лидером, необходимы:

- признание командой профессиональной компетентности и превосходства лидера;
- полное доверие команды к действиям и решениям лидера, признание его человеческих качеств, убежденность в его честности, порядочности, вера в его искренность и добросовестность.

На разных этапах формирования команды руководитель обязан применять разную управленческую стратегию. Выбор стратегии руководства зависит от степени признания лидерства и квалификации исполнителей (Таблица 14).

Таблица 14. Стратегии руководства и их применение

Стратегия	Действия	Рекомендации по применению
«Инструкции»	Директивное управление, жесткое назначение работ, строгий контроль сроков и результатов.	Руководитель еще не признан командой как профессионал, не имеет кредита доверия.
«Объяснения»	Сочетание директивного и коллективного управления. Объяснение своих решений.	Доверие есть, а уверенности в правильности и профессионализме решений нет – надо объяснять.

«Участие»	Приоритетное коллективное управление, обмен идеями, поддержка инициативы подчиненных.	Руководитель признан как профессионал. Но не имеет доверия со стороны команды. Чтобы его завоевать, необходимо активно привлекать участников коллектива к выработке и принятию решений.
«Делегирование»	Пассивное управление сформировавшегося руководителя. Делегирование полномочий и наблюдение за работой команды.	«Высший пилотаж». Коллективное лидерство. Команда становится самоуправляемой. Руководитель отслеживает макропроцессы. Доверие и признание должно быть взаимным. Следует помнить, что руководитель делегирует свои полномочия, но не свою ответственность за результаты проекта.

Когда я только начинал руководить людьми, основной моей проблемой было неумение и нежелание делегировать полномочия. Я отказывался делегировать что-либо другим, так как считал, что для этого требуется слишком много времени и усилий, и что я сам с этим справлюсь быстрее и лучше.

Правильное делегирование другим является, возможно, самым мощным и наиболее эффективным из всех имеющихся инструментов руководителя, лучшим индикатором эффективного управления. Делегирование невозможно без взаимного доверия. «Доверие – это высшая форма мотивации человека. Оно вызывает к жизни лучшее, что в человеке есть» [3]. Тот, кому делегировали руководство, становится сам себе боссом, ведомым собственной совестью в направлении достижения оговоренных желаемых результатов. Доверие требует времени и терпения, необходимости обучать и развивать людей, так чтобы их способности смогли соответствовать этому доверию. Но сделав эти разовые инвестиции в человеческий капитал, мы потом долго сможем получать дивиденды.

Стивен Кови предлагает пять пунктов «договора о делегировании»:

1. Ожидаемый результат (что надо сделать, но, ни в коем случае, как). У одного из моих коллег над рабочим столом висел листок со следующим предупреждением: «Приходите ко мне с проблемой, и вы получите решение. Если придете ко мне с решением – получите проблему».
2. Правила и стандарты (минимум ограничений, например, интерфейс или что точно не надо делать).
3. Выделяемые ресурсы (рабочее время, возможно с помощниками, аппаратно-программные средства).
4. Критерии оценки («что такое хорошо и что такое плохо?»).
5. Последствия (мотивы и стимулы).

Чтобы стать лидером, руководитель в процессе своей повседневной деятельности в зависимости от ситуации должен уметь исполнять в команде разные роли:

1. *Штурман* - формирует общее видение целей и систему ценностей, определяет курс, учитывая постоянные изменения, которые происходят вокруг и находя новые возможности.
2. *Образец* для подражания с точки зрения человеческих качеств. Личность, которая заслуживает полное доверие. «Учитель не тот, кто учит, а тот, у которого учатся».

3. *Помощник* – создает и, когда необходимо, меняет структуры, процессы, условия, которые обеспечивают эффективность работы каждого. Лидеры следуют правилам до того момента, пока они не увидят, что правила перестают действовать.
4. *Вдохновитель* – выявляет и направляет способности каждого на достижение результатов, а не на процессы и методы. Поощряет свободу, ответственность, инициативу и творчество, признает право на ошибку.

Для того чтобы эффективно исполнять эти роли, руководитель обязан обладать следующими профессиональными компетенциями:

- Видение целей и стратегии их достижения.
- Глубокий анализ проблем и поиск новых возможностей
- Нацеленность на успех, стремление получить наилучшие результаты.
- Способность сочувствия, понимания состояния участников команды.
- Искренность и открытость в общении.
- Навыки в разрешении конфликтов.
- Умение создавать творческую атмосферу и положительный микроклимат.
- Терпимость, умение принимать людей какими они есть, принятие их права на собственное мнение и на ошибку.
- Умение мотивировать правильное профессиональное поведение членов команды.
- Стремление выявлять и реализовывать индивидуальные возможности для профессионального роста каждого.
- Способность активно "обеспечивать", "доставать", "выбивать" и т.д.

Важное качество лидера это умение принимать людей такими, какие они есть. Принимать их недостатки, уметь разглядеть их достоинства и использовать эти достоинства для достижения общей цели.

История 10. «Старший программист»

Имеет глубокие знания и развитый интеллект, быстро осваивает все новое, нацелен на решение трудных задач. Пользуется заслуженным авторитетом среди коллег.

В начале проекта активно выдвигал новые идеи, убедительно их обосновывал, добивался их признания всеми. Находил неизвестные возможности, существенно сократившие трудоемкость работ по проекту.

В середине проекта потерял интерес. Стал «витать в облаках» и отвлекаться на изучение каких-то новых технологий. Постоянно заваливает сроки, делает глупые ошибки, непростительные для его опыта. Расхолаживающе воздействует на команду.

Диагноз. Программист, скорее всего, склонен исполнять командную роль «генератор идей» или «аналитик».

Рекомендации. Лучше всего перевести на новый сложный проект, который станет для него «вызовом», и в котором он будет более эффективен, чем в несвойственной для него роли «реализатор».

Применение той или иной стратегии руководства, принятие лидером на себя правильной роли зависит от стадии, на которой находится формирование команды.

История 11. «Шумиха»

Симптомы:

Частые смены приоритетов задач.

Споры о том, что делать.

Много вопросов по каждой задаче.

Сомнения в реальности сроков.

Недовольство отсутствием прогресса.

Диагноз. Команда находится на первом этапе образования – «Объединение». Цели и стратегия их достижения не ясны и не приняты всеми участниками команды.

Рекомендации. Стратегия – «инструкции», директивное управление. Лидер выполняет роли «штурман» и «вдохновитель».

История 12. «Конфликты»

Симптомы:

На совещаниях бесконечные неконструктивные споры и дискуссии.

Постоянно доминируют одни и те же лица. Другие предпочитают отмалчиваться.

Мнения высказываются как объективные факты. «На самом деле эта задача решается так...!»

Постоянно даются оценки. «Это все неправильно!» «Это все не важно!»

Присутствует агрессия «Ты, просто, ничего не понимаешь!». «А ты разве не знаешь, что...!»

Диагноз. Команда находится на стадии «Разногласия и конфликты». Отсутствие культуры эффективных коммуникаций. Неумение слушать и слышать.

Роль лидера – «помощник», образец.

История 13. «Неразбериха»

Симптомы:

Никто не знает, где искать необходимую для решения задачи информацию.

Не понятно, кто и за что отвечает.

Постоянные обращения к руководителю за уточняющими указаниями.

Постоянная суета. Задачи начинают решаться, но не завершаются.

Много делается того, что на самом деле не нужно. Дублирование работ.

На совещаниях бесконечные неконструктивные споры и дискуссии.

Постоянно доминируют одни и те же лица. Другие предпочитают отмалчиваться.

Мнения высказываются как объективные факты. «На самом деле эта задача решается так...!»

Постоянно даются оценки. «Это все неправильно!» «Это все не важно!»

Присутствует агрессия «Ты, просто, ничего не понимаешь!». «А ты разве не знаешь, что...!»

Диагноз. Команда не прошла стадию «Становление». Роли не разделены. Отсутствуют или неэффективны рабочие процессы.

Рекомендации. Стратегия – «объяснения». Роль лидера – «помощник», «образец».

История 14. «Оборона»

Никто не стремится проявлять инициативу.

На совещаниях предпочитают отмалчиваться.

Все ждут принятия решений от руководителя.

Много жалуются на проблемы и трудности в работе.

Требуют письменных уточнений и разъяснений.

Стараются не информировать о своих ошибках. Если ошибка всплыла, то стремятся свалить ответственность на других.

Все пытаются избегать конфликтов или уклоняться от их разрешения.

Диагноз. Недостаток доверия. Неверие в свои силы. Стремление избежать неудач.

Рекомендации. Стратегия - «участие», роль лидера - «вдохновитель».

История 15. «Менеджер должен занимать очередь, чтобы покритиковать сотрудника...»

Симптомы:

Ни одно предлагаемое решение не принимается на веру. Все требуют факты для его обоснования.

Активно анализируются возможные негативные последствия или упущенные возможности при принятии решения.

Конфликты носят исключительно производственный характер. При решении конфликтов активно ищутся взаимовыгодные возможности.

«Менеджер проекта должен занимать очередь, чтобы покритиковать сотрудника, не выполняющего свои обещания» [25].

Диагноз. Команда, наконец-то, достигла плато наивысшей эффективности и находится на стадии «Отдача».

Рекомендации. Стратегия – «делегирование». Роли лидера – «штурман» и «вдохновитель». Задачи – наблюдать за изменениями, «точить пилу», поддерживать высокий уровень мотивации в команде.

Проблемы неисполнения

Для того, чтобы ваш сотрудник мог эффективно решить поставленную вами задачу, *необходимо и достаточно* выполнение четырех условий:

1. *Понимание целей работы.* Роль лидера «штурман». Задача лидера обеспечить общее видение целей и стратегии их достижения.

2. *Умение ее делать.* Роль лидера «образец». Задача лидера – научить, быть наставником и образцом для подражания
3. *Возможность ее сделать.* Роль лидера «помощник». Задача лидера обеспечить для этого все необходимые условия.
4. *Желание ее сделать.* Роль лидера «вдохновитель». Обеспечить адекватную мотивацию участника на протяжении всего проекта, на мой взгляд, одна из самых сложных задач лидера.

История 16. «Делаем все по правилам!»

Симптомы:

Программист стремится сделать наиболее общее решение задачи, учесть все возможные последующие изменения и расширения.

Старается разработать самый быстрый алгоритм, требующий минимальных ресурсов.

Использует в решении все лучшие практики, паттерны проектирования, самые новые инструменты.

Диагноз. «Зло преждевременной оптимизации». Причина программист не научился управлять приоритетами. Не различает важные и не важные дела. Низкая эффективность. Для решения задач в срок постоянно не хватает времени. Сверхурочные. Раздражительность. В результате появляется большой и проблемный код.

Рекомендации. Более четко формулировать цели и расставлять приоритеты. Определить конкретные критерии оценки результата, директивное управление. Нацеливайте программистов на правильное решение задачи максимально быстрым способом. Опыт свидетельствует, что более чем в 90% случаев ее не придется переделывать. Если работа вашей подпрограммы занимает только 1% общего времени выполнения, то, затратив сколько угодно времени на оптимизацию ее алгоритма, вы не добьетесь улучшения системных характеристик более чем на 1%. Такой выигрыш вряд ли стоит того, чтобы тратить время на оптимизацию. Со всеми остальными «программистскими наворотами» дело обычно обстоит так же.

Неадекватное понимание целей конкретного проекта может привести к проблемам. Большинству опытных программистов свойственно стремление решать свою задачу не просто хорошо, а как можно лучше.. Доводя свою программу до совершенства, программист способен затратить гораздо больше времени, чем реально требуется на задачу. Это стремление всегда вступает в конфликт с коммерческой стороной профессионального программирования, как способа получения прибыли. С точки зрения заказчика, качественная программа это та, которая удовлетворяет заявленным требованиям. Не более того.

Второе условие эффективной работы программиста – умение делать порученную работу. Как правило, начинающие программисты мало что умеют. Их главный метод программирования – копирование чужих образцов (Copy/Paste). Это естественный путь обучения ремеслу. Вспомним художников, которые учатся, копируя полотна великих мастеров. Важно, чтобы образцы для подражания были достойными. Поэтому целесообразно поручать задачу паре программистов, в которой один из них выступает наставником, а другой подмастерьем, перенимающим опыт.

У меня нет личного опыта парного программирования, которое рекомендует xProgramming, когда одну программу пишут по очереди. Но есть накопленная с годами уверенность, что ревизия кода более опытным коллегой на предмет поиска «изобретения велосипеда» просто необходима. Кстати «изобретение велосипеда» любимое занятие не только среди начинающих, но и среди уже достаточно опытных программистов, у которых всегда возникает потребность переписать все по-своему. Этому, как правило, есть две причины. Первая - недооценка сложности поставленной задачи. Вторая - недостаток времени для изучения достижений и возможностей новых технологий, используемых в проекте. Дополнительно, парная ответственность за исходный код страхует ваш проект от негативных последствий неожиданного ухода одного из специалистов

Третье условие эффективной работы – программисту должна быть предоставлена *возможность* решить поставленную задачу. Здесь речь идет не о тривиальном наличии компьютера и инструментов разработки. И не о наличие отдельного кабинета, о котором пишет Де Марко. В творческой деятельности обязательным элементом ответственности является свобода выбора пути решения стоящей проблемы. Свобода не только необходимое условие творчества, но и важный мотивирующий фактор. Предоставьте членам проектной команды право на ошибку. Это нормальный атрибут творческого поиска. На ошибках учатся. Умный не тот, кто не делает ошибок, а тот, кто их не повторяет. Чем чаще мы ошибаемся, тем быстрее учимся и быстрее добиваемся успеха. Ошибки - это просто часть культуры инновации.

Одним из элементов свободы является отсутствие жестких сроков на выполнение задачи. Для профессиональных управленцев отсутствие жестких сроков может звучать как нонсенс, но в творческой деятельности это один из обязательных элементов эффективной работы. Бессмысленно заставлять программистов работать больше, устраивать сверхурочные авралы и субботники. Работать больше, это совсем не значит - работать продуктивнее. Наоборот. Излишнее давление и суета приводят к непродуманным решениям и многочисленным последующим переделкам. «Хорошо управляемое предприятие - это спокойное место. Зато «фабрика, отличающаяся "кипучей" деятельностью и "трудовым героизмом" работников, который бросается в глаза любому посетителю, является на самом деле плохо управляемой». Это не я сказал, это говорит управленец «в законе» [38].

И наконец, четвертое, задача не будет решена за любое отведенное на нее время, если у программиста нет желания это сделать. Или сумейте мотивировать программиста на выполнение задания, или передайте его задачу другому участнику проекта. Иного пути нет. Часто бывает ситуация, что программист не знает и не умеет, например, если технология совершенно новая, но хочет. И этого стремления, как правило, бывает достаточно, чтобы освоить новую технологию и успешно решить поставленную задачу.

Глава 5. Практики демотивации

Классификация антипаттернов руководства

Бесполезно пытаться мотивировать участников команды на успех проекта, не исключив из своего руководящего арсенала практики демотивации, применение которых в управлении творческими коллективами не приносит ничего, кроме вреда. Почему эти методы управления людьми не работают в проектах разработки ПО, мы обсудили во Введении.

Как отмечалось выше, эффективный менеджер – это в первую очередь лидер, который признан командой. Во-первых, он должен получить признание командой своей профессиональной компетентности. Во-вторых, между ним и командой должно быть установлено взаимное доверие. Не выполнение хотя бы одного из этих условий служит причиной воспроизводства руководителем неадекватных методов управления, которые приводят к катастрофическим результатам.

Некоторые руководители программных проектов, в первую очередь начинающие, в работе с людьми постоянно наступают на одни и те же грабли. В компьютерной науке для подобных частонаступаемых граблей придумано название «антипаттерны» - это повторно используемые практики, которые могут давать видимость эффекта и даже временный эффект, однако, их применение наносит несоизмеримый ущерб конечному результату.

Антипаттерны руководства командами разработки ПО можно разделить на два класса, исходя из «диагноза заболевания» - источника их устойчивого повторения.

Некомпетентность. Руководитель не самостоятелен. Не имеет необходимого опыта. Не является специалистом в своей области. Сильно зависим от окружения и обстоятельств. Не готов брать на себя ответственность. «Армия львов под управлением барана всегда проиграет армии баранов под управлением льва», (с) Наполеон Бонапарт.

Мнительность. Отсутствие доверия между руководителем и участниками проекта. Чрезмерная настороженность. Скрытность. Неспособность делегировать полномочия. Руководитель не

осознал синергию взаимозависимости и взаимопомощи. Все взаимодействия рассматривает только в духе «выиграл/проиграл». Исходит из предпосылок индустриальной эпохи Генри Форда: «Работники ленивы, поэтому им необходимы внешние стимулы для работы. У людей нет честолюбия, и они стараются избавиться от ответственности. Чтобы заставить людей трудиться, необходимо использовать принуждение, контроль и угрозу наказания». Манфред Кетс де Врис называет данное отклонение «параноидальным управлением».

Каждый руководитель, который применяет перечисленные ниже практики в руководстве разработчиками ПО, должен помнить, что тем самым он расписывается в своей некомпетентности и параноидальной подозрительности. Ниже приведена лишь небольшая часть из существующих антипаттернов управления командами. Однако, перечисленные здесь грабли, на мой взгляд, являются наиболее распространенными и тяжелыми по последствиям наступания на них. Следует заметить, что на практике описанные антипаттерны чаще всего применяются не изолированно, а в той или иной композиции.

Антипаттерны некомпетентности

«Я сделал все, что мог!»

Руководитель не способен оценить сложность решаемых задач, эффективность способов их решения и реальное состояние проекта. Постоянные негативные оценки прогресса проекта: «Все плохо! Надо делать быстрее! Надо делать лучше!», не вдаваясь в подробности «как это надо делать». Раздувание недостатков до глобальных размеров. Постоянная публичная критика участников команды и жалобы вышестоящему руководству на некомпетентность исполнителей. Надежда на то, что если проект провалится, то удастся свалить вину на команду: «Я сделал все, что мог!». Если же проект окажется успешным – присвоить успех себе: «Если бы я не «строил» команду, ничего бы не получилось!»

«Yes-man!»

Руководитель, который полностью зависим от Босса. Всегда согласен с его мнением, как бы оно не расходилось с его собственными взглядами и мнением его команды. Руководствуется принципом: «Возлюби Босса своего и никогда не спорь с ним!». Интересы фирмы, команды, продукта и его пользователей не учитываются вовсе. «Босс хочет, что бы мы закончили проект в два раза быстрее. Как? Это ваша забота, вам за это деньги платят!». Не принимает ни одного решения без согласования с Боссом. Испытывает постоянную потребность «угадать и угодить» Культивирует любимчиков в команде, которые никогда с ним не спорят.

«Охота на ведьм»

Руководитель сознает свою некомпетентность и опасается, что если он не возглавит борьбу с «врагами» проекта, то «врагом» проекта может оказаться он сам. Постоянный поиск виновных и «козлов отпущения». Наказания. Разжигание конкуренции в команде: «После проекта останутся только лучшие!». Действия по принципу «разделяй и властвуй». Исполнители проекта разделяются на несколько противоборствующих клик. Пресечение любой критики и опасного для себя поведения подчиненных: «Ваше поведение непрофессионально / некорпоративно!» (Вы ни разу не слышали этой фразы – завидую, вам везло с начальниками).

«Нет времени точить пилу!»

Руководитель не умеет управлять приоритетами, постоянно занимается пожаротушением, полностью погружен в решение неотложных вопросов. Времени на уточнение целей, разработку стратегии, адекватное планирование, оценку и повышение эффективности применяемых процессов не остается. Героический труд («у плохих генералов всегда много героев»). Темп поступления проблем превышает скорость их решения. Большинство задач, которые получают участники проекта, имеют наивысший приоритет и срочность. «Это надо было сделать еще вчера!» Работа постоянно выполняется под давлением нереальных сроков. Обучение, анализ, проектирование и рефакторинг – «это все потом!». Постоянные сверхурочные и авралы. Большой

проблемный код. В англоязычной википедии у этого антипаттерна другое, более жесткое название – «Headless chicken».

Антипаттерны мнительности

«Агрессия»

«На то и волк в лесу, чтоб пастух не дремал!» Руководитель стремится удерживать подчиненных вне «зоны комфорта». «Настоящие лидеры задают трудные вопросы и выбивают людей из зоны их комфорта. Затем они управляют возникающим стрессом!», (с) Р.А. Хейфетц, Д.Л. Лори, «Работа лидера», 1997. Постоянное давление. Нереальные сроки. Сверхурочные. Авралы. Непрерывная критика и понижение самооценки. «Заодно можно сэкономить на зарплате!». Грубость. Запугивание. «У нас незаменимых нет!» «Поощрение непричастных и наказание невиновных». Назначение в несколько проектов одновременно. «Чем больше, тем лучше - обязательно что-нибудь провалит! Сотрудник должен все время ощущать свою вину!» Постоянные перемещения людей по горизонтали и вверх-вниз. Обязанности определены самым общим образом: «Ты в ответе за все!» Поручения не зависят от обязанностей. Правила меняются по ходу игры.

«Управление грибами»

Удержание работников в неинформированном и постоянно занятом состоянии. Замыкание всех внешних и внутренних потоков информации на себя. Фильтрация и искажение информации в личных интересах. Удержание исполнителей в постоянной зависимости от более информированного начальника. Строгое разграничение прав доступа к проектной документации и исходным кодам. Ограничение доступа в Интернет («а вдруг узнают среднюю зарплату на рынке?»), запрет ICQ, препятствование общению с коллегами из других компаний. Загрузить работой так, чтобы времени на обдумывание чего-либо не оставалось. Постоянно находить какие-нибудь «важные и неотложные» дела. Почему управление грибами? Потому, что грибы держат в темноте и кормят навозом.

«Микроменеджмент»

Отсутствие делегирования в любом виде. Строгий контроль всех работ через активное личное наблюдение. Руководитель замыкает на себя принятие решений по всем вопросам. Влезает во все самые мелкие работы, от структуры таблиц в БД до цвета шрифтов в интерфейсе. Вместо объяснения того, что надо делать, постоянные указания на то, как надо делать. Считает себя самым компетентным. Навязывает собственные идеи. «Думать – это моя работа! Просто, делай, как я сказал!» Жестко контролирует, каждый шаг исполнителей.

«Методологическое безумие»

Безграничная вера руководителя в методологию с большой буквы «М» - всеобъемлющую теорию того, как следует решать весь класс задач, возникающих в процессе производства. «Методологию создавали умные люди, а исполнители некомпетентны!» Методология принимает все решения, люди не принимают решения вовсе. Многоступенчатая бюрократия. Все процессы регламентированы. Все делается по инструкции. Эксперименты запрещены. Применяемые методы ограничены. Установлен тотальный контроль соблюдения регламентов. Инструкции постоянно разрастаются вследствие попыток учесть все возникающие новые ситуации. Внедрены всеобъемлющие нормы и метрики. Большая часть времени исполнителей тратится на соблюдение правил и писанину, которую никто никогда не читает. Большая доля «сизифова труда».

Последствия применения антипаттернов

Поскольку все рассматриваемые антипаттерны связаны с человеческим фактором, то негативные последствия их применения схожи. Это:

- Фатальная демотивация исполнителей. Вместо мотивирования сотрудников на успех, мотивирование их на избегание риска и негативных для себя последствий.

- Подавление свободы, самостоятельности, творчества и инициативы.
- Деструктивное подчинение. Это когда все работают строго по инструкции и только в соответствии с указаниями руководства. Отсутствие ответственности исполнителей. «А какие ко мне претензии? Как сказали, так я и сделал!»
- Низкая эффективность и качество работы.
- Ухудшение морального климата. Вместо доверия и сотрудничества подозрительность и формальное взаимодействие. А вы никогда не видели, как тимлид беседует с программистом только «под протокол» и с подписями на каждом листе?
- Стрессы. Усталость участников. Личные проблемы.
- Увольнение наиболее профессиональных сотрудников.
- Провал проекта.

Глава 6. Мотивация

Гуманистическая теория мотивации

Управлять людьми – значит побуждать их к правильным действиям. А побуждать их можно, только управляя мотивами. Согласно теории мотивации для того, чтобы человек совершил какое либо действие он должен испытывать некую потребность и предполагать, что, выполнив это действие, он в той или иной степени эту потребность удовлетворит. Поэтому управление мотивами осуществляется, как правило, в двух направлениях: 1) формирование правильных потребностей; 2) формирование правильной оценки степени их удовлетворения.

Команда успешного проекта это команда *победителей*. Успешный проект должен сделать победителем всех его участников [39]. Рассматривайте работу команды над проектом как кооперативную игру с ненулевой суммой, в которой каждый участник, стремится к достижению своих личных целей, продвигая проект к успеху.

Существуют десятки различных теорий мотивации. Наилучшим образом накопленный опыт организации креативных людей «лег» на постулаты экзистенциальной и гуманистической психологии (А. Маслоу, К. Роджерс, В. Франкл, и др.). На мой взгляд, эффективный программист – это самоактуализирующаяся личность, для которой необходима *метамотивация* [7]. Об этом же пишет Э. Йордан [40]: «Деньги, выгода, комфорт и тому подобное являются факторами «гигиены» - их отсутствие вызывает неудовлетворенность, однако они не могут заставить людей полюбить свою работу и дать им необходимые внутренние стимулы. Что действительно может дать такие стимулы, так это ощущение значительности достигнутых результатов, гордость за хорошо выполненную работу, более высокая ответственность, продвижение по службе и профессиональный рост - все то, что обогащает работу». О том же 35 лет назад писал Вейнберг [41]: «Творчески работающему программисту присуща глубокая внутренняя мотивация».

Работа А. Маслоу «Мотивация и личность» [42] была впервые опубликована в 1954 году и долгое время не переводилась на русский язык. Именно в ней была сформулирована иерархическая теория потребностей, выстраивающая пирамиду с основанием из базовых нужд и с потребностью в самоактуализации на вершине (Таблица 15).

Таблица 15. Пирамида потребностей А. Маслоу

<p><i>Потребность в самоактуализации, заключающаяся в самореализации и росте как личности.</i></p>
--

Потребности в уважении, включающие внутренние (самоуважение, личные достижения) и внешние (статус, признание) факторы уважения.

Социальные потребности, заключающиеся в привязанности, принадлежности к какой-либо общности, одобрении со стороны других, дружбе.

Потребности в безопасности, включающие потребности в защите от физических и психологических опасностей и уверенность в выполнении в будущем физиологических потребностей.

Физиологические потребности, являющиеся необходимыми для выживания (еда, жилище, секс).

С точки зрения А. Маслоу, каждый человек обладает врожденным стремлением к самоактуализации, причем это стремление к максимальному раскрытию своих способностей и задатков выступает наивысшей человеческой потребностью. Правда, для того чтобы эта потребность проявилась, человек должен сначала удовлетворить всю иерархию нижележащих потребностей. Высшая природа человека опирается на его низшую природу, нуждаясь в ней как в основании, и рушится без этого основания. Таким образом, большая часть человечества не может проявить свою высшую природу без удовлетворения базовой низшей природы.

Какие именно потребности надо удовлетворять, чтобы мотивировать участника команды, зависит от характеристик конкретного человека:

- Особенности личности.
- Профессионального опыта.
- Внешних обстоятельств, жизненной ситуации (семейные проблемы, экономический кризис, сокращения в компании).
- «Багажа» - нечто, что мешает активному позитивному отношению члена команды к поставленной задаче. Например, у человека личные проблемы, или он рассчитывал стать менеджером вместо вас.

Необходимо помнить, что мотивы могут изменяться в течение проекта. Еще одно важное замечание. Следует избегать упрощенных подходов и распространенных ошибок мотивации, к которым следует отнести следующие.

- *«То, что является мотивацией для меня, будет являться мотивацией и для других!»* Необходимо помнить, что все люди разные и находятся в разных жизненных условиях.
- *«Мотивацию для человека, прежде всего, составляют деньги».* Это утверждение верно в случае, когда денег нет. Если материальные потребности удовлетворены процентов на семьдесят, то дальнейшее повышение доходов не заставит сотрудника работать более эффективно. Проверено многократно. Более того, слишком высокие, относительно рыночных, зарплаты в компании могут породить эффект «сверхлояльности», когда сотрудник из боязни потерять свое рабочее место будет соглашаться со всем «бредом», который порой поступает от начальства.
- *«Самый лучший лидер проекта это «умелый вдохновитель».* Трудовой энтузиазм, основанный на «идеологической обработке», быстро угасает и перестает мотивировать. Приверженность миссии компании – это не для российского менталитета. Лидер просто обязан быть наставником, который помогает участникам команды выявлять свои *личные* цели и способствует их достижению.

- «Эти люди – профессионалы. Им не нужна никакая мотивация». Попробуйте уговорить суперпрограммиста перейти на техническое сопровождение системы, даже если вы пообещаете ему в два раза большую зарплату.
- «Я ко всем отношусь одинаково. Это станет основной мотивацией для людей». Еще раз напомню, все люди различаются по типу личности, возрасту, опыту, квалификации и много еще чем.
- «Если правильно мотивировать, все в людях можно изменить». Это заблуждение. Взрослые люди меняются очень медленно. Порой на это требуется годы, порой – десятилетия. У вас в распоряжении просто нет этого времени.

В своих поздних работах, в частности, [7], опубликованной посмертно в 1971 году, А. Маслоу, пошел дальше и существенно пересмотрел свою концепцию мотивации и личности, фактически отказавшись от многоступенчатой пирамиды потребностей, которую продолжают усердно цитировать многочисленные источники. Все человеческие потребности он подразделил на низшие, «дефицитарные», продиктованные недостатком чего-либо и потому насыщаемые, и высшие, «бытийные», ориентированные на развитие и рост, а, следовательно, — ненасыщаемые. Более того, он пришел к выводу, что только *самоактуализирующиеся* люди могут считать себя полностью психически здоровыми.

Кстати, говоря о термине «самоактуализация», нужно отметить, что он использовался еще К. Юнгом, хотя это редко отмечается психологами-гуманистами [43]. По Юнгу, самоактуализация означала конечную цель развития личности, достижение ею единства на базе наиболее полной дифференциации и интеграции различных ее сторон. Весьма близки по своему содержанию к идее самоактуализации также концепции «стремления к превосходству» и «творческого Я» в «индивидуальной психологии» А. Адлера.

Мотивация и тип личности

Особенности типа личности [14] определяют мотивацию участников команды на решение задач определенной специфики (Таблица 16).

Таблица 16. Зависимости мотивации от характеристик типа личности

E – Extroverted (экстраверт)	I – Introverted (интроверт)
<p>Лучше справятся с:</p> <ul style="list-style-type: none"> • нечеткими задачами, предполагающими широкое поле деятельности; • задачами, требующими повышенной активности и интенсивных энергозатрат; • первоочередными задачами, требующими не столько качественного, сколько быстрого решения. 	<p>Лучше справятся с:</p> <ul style="list-style-type: none"> • ясными, конкретно поставленными задачами; • не первоочередными, но требующими глубокой и качественной проработки задачами; • задачами, допускающими неспешность и постепенность в своем решении.
S – Sensing (конкретное восприятие)	N – iNtuitive (интуиция)
<p>Привлекайте для:</p> <ul style="list-style-type: none"> • текущих, повседневных, технологически 	<p>Привлекайте для:</p> <ul style="list-style-type: none"> • нестандартных, неясных по

<p>отработанных заданий;</p> <ul style="list-style-type: none"> практически ориентированных задач с гарантированной материальной отдачей; простых по структуре, четко сформулированных задач. 	<p>содержанию, запутанных задач;</p> <ul style="list-style-type: none"> решения задач перспективы, предполагающих отдаленный по времени результат; экспериментальных, исследовательских задач.
T – Thinking (логика)	F – Feeling (этика)
<p>Поручайте:</p> <ul style="list-style-type: none"> управление, организацию и контроль производственной деятельности; составление и анализ планов, графиков, программ и других логических структур; механические, технические, не требующие частых и глубоких контактов с клиентами задания. 	<p>Поручайте:</p> <ul style="list-style-type: none"> работу, связанную с постоянными контактами с клиентами и удовлетворением их эмоционально-этических запросов; рекламу, разведку ресурсов, установление неформальных связей с окружающим миром; вопросы отдыха, праздничных мероприятий, налаживания благоприятной атмосферы в коллективе.
J – Judging (рациональность)	P – Perceiving (иррациональность)
<p>Мотивированы на:</p> <ul style="list-style-type: none"> планомерные, регулярные, предсказуемые по срокам задачи; задачи, требующие систематичности и последовательности действий; задачи, которые предполагают или допускают упорядочение. 	<p>Мотивированы на:</p> <ul style="list-style-type: none"> разнообразные по подходам, малопредсказуемые по срокам задачи, содержащие элемент стихийности; задачи, требующие гибкости, смекалки, наблюдательности; задачи, возникающие в экстремальных и кризисных ситуациях.

Мотивация и опыт

Потребности каждого участника команды так же зависят от его возраста, опыта и квалификации. Предлагаю свою экспертную оценку распределения мотивирующих потребностей для профессиональных разработчиков ПО (Таблица 17).

Таблица 17. Зависимость мотивации участника команды от опыта и возраста

Потребности	Профессионализм		
	Начинающий	Опытный	Мастер
Материальные (зарплата, условия труда, социальный пакет и проч.)	50%	20%	
Безопасности (стабильность компании, востребованность технологии проекта на		20%	

рынке труда, возможность повысить свою квалификацию)			
Принадлежности (возможность учиться у более опытных коллег, опыт участия в успешном проекте, признание в коллективе)	40%	20%	10%
Самоуважения (развиваться, делать что-либо лучше других, повышение в должности, самостоятельность и ответственность в работе)	10%	30%	40%
Самоактуализации (амбициозность целей проекта – сделать то, что никто не делал или не смог сделать)		10%	50%

Пропуск в той или иной графе свидетельствует не об отсутствии соответствующей потребности, а о том, что при помощи дополнительного удовлетворения этой потребности не получится мотивировать данного специалиста. Например, для профессионала материальные потребности и потребности безопасности не играют существенной роли, поскольку, если они в достаточной степени не удовлетворены, он просто меняет работу и больше не думает о них.

Для начинающих программистов хорошим стимулом является само участие в успешном проекте (может быть в первом в их жизни), возможность учиться ремеслу у более опытных и искушенных коллег.

Для опытных программистов хорошим стимулом может служить новизна и востребованность на рынке труда технологий, используемых в проекте (потребность безопасности). Для них также существенны сложность и самостоятельность (потребность самоуважения) в решении поставленных задач. Как правило, я стремлюсь ставить задачи примерно в 1,5 раза сложнее, чем те которые данный программист решал ранее. Для опытного программиста каждая новая задача должна предоставлять дополнительную возможность доказать свой профессионализм.

История 17. «Хороший парень»

Программист:

Общительный.

Стремится всем нравиться.

Оптимист, особенно в оценках проекта.

Готов выполнять любое поручение начальника.

Никому не говорит «нет».

Старательно избегает конфликты.

Никогда не настаивает на своем мнении.

Озабочен тем, «чтобы угадать и угодить».

Диагноз. Сильно зависим от мнения окружающих. Не имеет четких личных целей. Мотивирован на комфортные взаимоотношения, а не на успех.

Рекомендации. Роль лидера – «наставник». Требуется помочь человеку выявить личные ценности и выработать стратегию их достижения.

История 18. «Тихоня»

Программист:

Ведет себя сдержанно, стремится не высовываться.

Ожидает подробных инструкций. Старается действовать строго в пределах своих функциональных обязанностей.

Старается задавать как можно меньше вопросов.

Редко высказывает свое мнение, никогда не настаивает на нем.

Избегает любые ситуации, связанные с возможными конфликтами.

Замалчивает «неприятную» информацию.

Диагноз. Не самостоятелен. Пассивен. Отсутствие доверия к руководителю. Мотивирован не на успех, а на избежание неудачи. Как правило, такое поведение обусловлено прошлым отрицательным опытом работы с агрессивным руководством.

Рекомендации. Роль лидера – «помощник». Поддержка, поощрение веры в себя, инициативы и самостоятельности.

История 19. «Ведущий программист»

Программист:

Имеет хорошую квалификацию и большой опыт работы, признанные командой.

В целом, неплохо справляется с порученными задачами, но делает «от и до», нельзя сказать, что по настоящему хорошо.

Инициативу не проявляет. Стремится получать задачи попроще.

Менеджер:

Попробовал повысить оклад, но это дало только кратковременный результат.

Обошел очередной премией. Вызвало только раздражение.

Диагноз. Отсутствует мотивация к эффективной работе. Возможно, это связано с тем, что человек не видит перспектив личного роста.

Рекомендации. Роль лидера - «наставник». Необходимо помочь выявить личные цели и спланировать карьерное развитие сотрудника.

Сложнее дело обстоит с суперпрограммистами. Их главным мотивом, как правило, служит самоактуализация, поэтому они стремятся решать задачи, которые до них еще никто не делал. Оптимальное их место в проекте – системная архитектура и реализация архитектурно значимых компонентов - скелета системы. При правильной мотивации оставшаяся часть их потребностей принадлежности и самоуважения реализуется через обучение коллег и передачу им своего опыта. На эту деятельность следует планировать до 50% времени суперпрограммиста. Суперпрограммист в проекте должен играть роль технического лидера, который ведет за собой остальных участников под лозунгом: «Делай как я!». Он всегда должен быть готов продемонстрировать, как можно решить эффективно любую задачу в проекте. Вывод, который отсюда следует это то, что на серьезный проект, надо набирать программистов разной квалификации. И начинающих, и звезд. Если вы берете в проект суперпрограммиста, то должны быть уверенным, что вы сможете эффективно использовать его опыт и найдете достойную задачу, которая его заинтересует.

Набор сотрудников

Девиз компании SouthWest Airlines: «Мы нанимаем жизненные позиции». Необходимо помнить, что мотивация разработчиков ПО начинается с приема на работу. В старой экономике людей нанимали за умения и обучали нужному отношению к делу. В новой экономике необходимо поступать с точностью до наоборот: *нанимать за нужное отношение к делу и учить необходимым умениям*. Не следует более нанимать людей, которые знают и умеют, а потом заниматься промыванием их мозгов и пытаться мотивировать их на эффективную работу. Нанимаем людей с правильным отношением к жизни, ищем тех, кто хочет работать, а затем, если необходимо, помогаем им получить требуемые технические навыки. Настоящий лидер предлагает не работу, а возможности.

В идеале, конечно, следует стараться привлечь и знающих, и умеющих, и подходящих по своим жизненным позициям. Но если приходится выбирать, то правильнее выбрать жизненную позицию. Рональд Рейган говорил: «Окружите себя самыми лучшими людьми, которых вы только сможете найти, передайте им в руки власть и не мешайте им». Этот подход не столько близок мне по характеру, сколько проверен опытом.

В настоящее время нет каких-либо достоверных формальных методик определения квалификации программиста. Мой опыт показывает достаточно значимую корреляцию хороших способностей к программированию и сверхвысокого значения IQ. Высокий балл диплома и престижное математическое или техническое образование свидетельствует об общих способностях кандидата успешно осваивать новый материал. Диплом о престижном высшем образовании немаловажен для хорошего программиста, хотя и не является необходимым условием. Я знаю программиста, который, не имея высшего образования, успешно руководит командой разработчиков в представительстве известной иностранной фирмы.

Отдельно остановлюсь на тестах и практических заданиях по специальности. Относительно их необходимости у меня имеются большие сомнения. Как правило, технические тесты ориентированы на проверку узкоспециальных знаний: структур данных, алгоритмов, конкретных стандартов и API, и т.д. Правильный/неправильный ответ на вопрос: « $i=1$; $i = i++ + ++i$; чему равно i ?», - мало о чем свидетельствует. Попытки давать кандидатам олимпиадные задачи по программированию, или задачи из книжки «Математические головоломки», у меня, как правило, ассоциируются со стремлением, возможно, не очень успешных вчерашних студентов к самоутверждению или с желанием продемонстрировать кандидату его несостоятельность, чтобы затем снизить предложение по зарплате. Я применял тесты по специальности - несколько десятков простых вопросов на 15-20 минут по базовым понятиям стека технологий J2EE таких, которые просто невозможно не знать, если ты имеешь практический опыт в данной области. Результаты этого теста использовались только для того, чтобы выяснить, в какой области кандидат более уверенно ориентируется и адекватнее позиционировать его среди имеющихся направлений разработки.

Технические тесты могут продемонстрировать уровень IQ и эффективность левого полушария, возможности которого подтверждены, как правило, дипломом о высшем образовании и всем успешным опытом предыдущей работы. Часто при обсуждении данного тезиса меня спрашивали: «А вдруг кандидат наврал о прошлом опыте?» На мой взгляд, одного двух вопросов с просьбой уточнить некоторые конкретные технические детали достаточно, чтобы рассеять или подтвердить подобные сомнения. И еще, если вы не в состоянии распознать то, что человек выдает себя не за того, кто он есть на самом деле, возможно, вам еще рано руководить людьми.

Креативность и эффективность человека в командной разработке зависят в первую очередь от уровня его эмоционального интеллекта, EQ и деятельности правого полушария: целеустремленность, воля, эмпатия, интуиция, способность к эвристическим суждениям. Я, к сожалению, пока не встретил достоверных тестов правого полушария. Поэтому слушаем, слушаем и еще раз слушаем. А главное *слышим!* При отборе кандидатов только очное интервью «глаза в глаза» и личный опыт работы с людьми позволят вам создать эффективную программистскую

команду. О практике проведения интервью могу рекомендовать работы [23, 44, 45], в которых изложены подходы, во многом совпадающие с моим опытом.

При проведении технического интервью следует придерживаться следующих основополагающих принципов:

- Помните, вы ведете переговоры с *потенциальным партнером* по бизнесу, а не пытаетесь «купить на рынке товар подешевле».
- Каждый человек уникален и ему присуща своя ценность. Если вам нечего предложить кандидату, то это не значит, что он «лузер».
- Имейте в виду, что несправедливо обиженный кандидат – пятно на бренде компании. Сегодня это значит не меньше, чем несправедливо обиженный клиент.
- Люди не рождаются победителями, они ими становятся. Кандидата стоит нанимать только в случае, если вы сумеете предоставить ему эту возможность.
- «Программист – это не профессия, это образ мышления», (с) программисты.

Проводить интервью, желательно, вдвоем с человеком, который будет непосредственно руководить или работать с кандидатом. Это позволит быть более объективным, посмотреть на кандидата с «разных сторон» и лучше его понять.

Ваша задача при проведении интервью – получить ответ на 3 вопроса:

1. Знает дело?
2. Умеет его делать?
3. Хочет его делать?

Обычно при проведении интервью я стараюсь придерживаться следующего сценария:

1. Вступление.
2. «Что Вы сделали?»
3. «Что вы хотите делать?»
4. «Какие у Вас есть вопросы?»
5. «Я могу предложить Вам ...»

Вступление

В этой части полностью солидарен с Джозел Сполски [23]: «Я трачу около 30 секунд, рассказывая кандидату, кто я такой и как буду проводить интервью. Я всегда заверяю кандидата, что нас интересует его подход к проблеме, а не его конкретные ответы. Кстати, в процессе интервью не стоит сидеть через стол от кандидата, поскольку это создает формальную преграду и мешает ему расслабиться. Лучше поставить стол к стене или обойти стол и сесть рядом с кандидатом».

Вспоминаем невербальные знаки: переговоры через стол – противостояние. Смена работы это всегда «стресс», который, по мнению психологов, занимает по значению 3-е место после развода семьи и пожара. Помогите кандидату прийти в себя. В противном случае вы будете беседовать с другим человеком.

«Что вы сделали?»

Вопросы:

- Что вы можете рассказать о вашем самом успешном проекте (если студент, то тема диплома или самый интересный учебный предмет)?
- Что вы считаете своей наибольшей профессиональной удачей?
- Что вам меньше всего нравилось делать на прежней работе?
- Почему хотите поменять место работы?

Что оцениваем:

- Эмоции и заинтересованность кандидата при ответах. Нет «горения» - «не зачет». «Вполнакала проекты не делаются» - очень точная формулировка И. Ставинского (цитата по [46]). Вряд ли, вам удастся его «зажечь». К вашей работе он будет относиться так же.
- Логическая последовательность, лаконичность и ясность повествования. Разработка программ это наполовину общение. Умение четко формулировать и объяснять так, чтобы это было понятно 10-летнему ребенку – необходимое качество. Уточняйте детали, даже если вы знаете, что такое «комфортное отображение», попросите кандидата объяснить, что он имел в виду, в двух словах.
- Зрелость личности. Если кандидат винит в своих неудачах коллег, окружение, начальство – несамостоятельная личность. Если не вспоминает коллег при рассказе о своих достижениях - возможно, недооценивает важность эффективного взаимодействия.

«Что вы хотите делать?»

Если кандидат не знает, что он хочет, его не стоит брать. Скорее всего, вы имеете дело с недостаточно зрелой личностью с человеком, который не стал лидером в своей личной жизни. Однако обязательно следует попытаться уточнить:

- Чем бы вам определенно не хотелось заниматься?
- Если бы работу выбирали вы, что бы это было?
- Какие качества вы цените/порицаете в коллегах?
- Как вы видите развитие своей карьеры?

Если кандидат объясняет к чему он стремиться, следует уточнить, адекватно ли он оценивает свои возможности. Предложите ситуации связанные с будущей работой:

- Предложите некорректную задачу (которая при заданных условиях имеет множество решений). Например, построить дом, (с) Джоэл Сполски, или «Определить количество бензоколонок в Москве» (с), rsdn.ru. Если кандидат спешит выдать первое пришедшее в голову решение – «не зачет».
- Предложите ситуацию на умение взаимодействовать, попробуйте отрицать что-то бесспорное для кандидата и поспорить с ним.

Что оцениваем:

- Целеустремленность. Если у человека есть ясное видение своих профессиональных целей и стратегии их достижения, а у вас есть проект, на котором он может свои цели реализовать, скорее всего, это «ваш человек». Даже если знаний и умений пока недостаточно. Желания «просто работать» мало. Скорее всего, человек мотивирован на избежание неудач, а не на достижение успеха. От такого трудно ожидать инициативы, поиска новых возможностей и творчества.
- Умение анализировать проблему, проводить ее декомпозицию на более простые компоненты, определять последовательность действий, синтезировать и обосновывать предлагаемое решение.
- Способность к диалогу и эффективному взаимодействию. Умение рассуждать, задавать вопросы, анализировать ответы, искать взаимовыгодное решение в конфликтных ситуациях.

«Какие у Вас есть вопросы?»

Собеседование должно быть двухсторонним. Вы пытаетесь понять, подходит ли кандидат вам. Кандидат – подходит ли ему работа, которую вы предлагаете. Если кандидат задаёт вопросы - это хорошо. Вопросы, которые обычно ожидаются:

- Что за проект, в котором предстоит работать: Какие цели? Когда начат? Каково состояние? Сколько человек постоянно на проекте? Опытность команды? Распределение ролей? Какова текучесть?

- Как организован процесс разработки ПО: Используемые технологии и подходы? Средства автоматизации? Техническое обеспечение? Принятие решений? Постановка задач? Отчетность и оценка результатов? Авралы? Сверхурочные?
- Система мотивации?

Что оцениваем:

- Факт наличия вопросов говорит об активной позиции и заинтересованности кандидата в работе.
- «Правильные вопросы» свидетельствуют об опытности специалиста.

«Я могу предложить Вам...»

Принимать решение следует сразу. Возможно, понадобится 5 мин. посоветоваться с коллегой. Не стоит откладывать решение. «Я могу предложить Вам...» Далее должны следовать описание позиции, должностных обязанностей и размера компенсации. Возможно, эта будет другая позиция, а не та, на которую, кандидат изначально позиционировал себя, если, на ваш взгляд в этой роли он будет более эффективен. Обсудите с кандидатом ваше видение его места в проекте. Не стоит предлагать кандидату другую позицию, если она с более низкой компенсацией. Будет похоже торговлю и попытку действия в духе «Выиграл/проиграл».

Если вы не смогли в ходе интервью снять все сомнения, не берите. Доверяйте своей интуиции. Лучше ошибиться и не взять достойного кандидата, чем рискнуть и получить «проблему». Будьте честны. Имейте волю отказывать: «Я могу предложить Вам...» Далее перечень конкретных отсутствующих знаний и навыков, приобретение которых позволит кандидату, занять данную позицию в будущем. Кандидаты возвращаются! Не стоит отказывать в стиле «Если будет принято положительное решение, мы Вам обязательно позвоним». Это, скорее всего, ложь. Не стоит подрывать доверие к своей компании.

Несколько советов кандидатам или Что еще оценивается в ходе интервью

Совет 1. *Никогда не опаздывайте на интервью.* Если вы все-таки опаздываете, предупредите заранее. Возможные негативные оценки опоздания:

- Вы распорядились чужим временем. Время одна из приоритетных ценностей делового человека, следовательно, вы проявили неуважения к его ценностям – это признак не умения эффективно взаимодействовать.
- Вы не сдержали свое слово. Возможно вы не обязательный человек.
- Вы не умеете прогнозировать риски и планировать свои действия.

Совет 2. *Одежда должна быть естественной.* Вы должны быть одеты так, как предполагаете ходить на работу, одежда не должна стеснять и сковывать вас. Возможные негативные оценки:

- Кандидат одет в майку и кеды, мятую, несвежую одежду — скорее всего, он не серьезно относится к работе, если не потрудился привести себя в приличный вид, собираясь на деловую встречу.
- Кандидат одет слишком помпезно: новые ботинки, костюм, дорогой галстук – по-видимому, кандидат не уверен в себе, хочет произвести впечатление внешним видом.

Совет 3. *Ведите беседу с достоинством, свободно, на равных.* Вы с работодателем предполагаете в будущем стать *партнерами*. Старайтесь слышать собеседника. Самоуверенность, но не наглость, серьезность, но не чопорность, внимательность, но не подобострастность, вдумчивость, но не умничанье – вот, на мой взгляд, отличительные характеристики зрелой личности. Возможные негативные оценки при общении:

- Кандидат смотрит на работодателя как солдат на генерала «лихо и слегка придурковато». От готовности всегда соглашаться с начальником – жди беды.
- Кандидат рассматривает собеседование как экзамен – стремиться рассказать все, что знает по заданному вопросу и вокруг него. Старается угадать ответ. Пытается скрыть, завуалировать свое незнание чего-либо, уходя от вопроса и переводя разговор на другую

тему. Кандидат, скорее всего, хочет представить себя тем, кем он на самом деле не является.

- Кандидат «растопыривает пальцы» и пытается доказать, что он «круче всех». Как правило, это не так. Возможно, это свидетельство завышенной самооценки.
- Кандидат скован. Отвечает односложно. Опасается проявлять эмоции. Скорее всего, он не уверен в себе, несамостоятелен, зависим.
- Кандидат не следит за обратной связью. Пытается продолжить повествование, даже если следуют вербальные или невербальные указания на то, что партнеру данная тема уже не интересна. Возможно, недостаточные навыки эффективных коммуникаций.

Оценка и развитие

Не думайте, что, наняв сотрудника, вы заключили с ним договор раз и не всегда. Вы должны нанимать сотрудника постоянно. Успешно работая в своей области, сотрудник приобретает новые знания и опыт, следовательно, стоимость его услуг растет. Поэтому его необходимо перекупать, пока это не сделали конкуренты, если, конечно, вы не стремитесь стать для них «кузницей кадров». С другой стороны, сотрудник сам может переоценить стоимость своего труда, исходя из своих реальных физических и психологических затрат, которые определяются качеством обеспеченности рабочего места, атмосферой в коллективе, отношением руководства и проч.

Пересмотр договора о найме, «по договоренности», без прозрачной процедуры дезориентирует и демотивирует сотрудников. В силу своей интроверсии, программист, как правило, предпочтет найти повышение зарплаты в другой компании вместо того, чтобы просить об этом свое начальство. Поэтому, настоятельно рекомендую, проводите периодический пересмотр условий договора - аттестацию.

В ходе регулярных аттестаций необходимо давать экспертную оценку следующих характеристик сотрудника:

- Квалификация - знания и опыт.
- Самостоятельность и лидерство: 1 – кодирование готовых алгоритмов; 10 – разработка стратегических решений.
- Инициативность и творчество: 1- требует инструкции на все свои действия; 10 – ищет и предлагает новые возможности, политики, процедуры.
- Эффективность взаимодействия: способность четко формулировать и объяснять, умение конструктивно разрешать конфликты, сопереживание, взаимопомощь.
- Результативность работы: процент рабочего времени занятого в успешных проектах, количество и сложность реализованных требований к продукту, число отлаженных строк кода в час (SLOC/h), плотность ошибок на 1000 строк кода (bugs/KSLOC).²
- Достижение целей, поставленных на предыдущей аттестации.

² Следует, однако, помнить, что формальные метрики лишь дополнительное средство при проведении оценок. Их стоит учитывать при сравнении сотрудников, которые достаточно длительное время занимаются сходной работой, например разработкой экранных форм пользовательского интерфейса. Если у одного разработчика при прочих равных условиях, производительность в два раза ниже, а плотность ошибок в два раза больше, то это, по крайней мере, повод для того, чтобы проанализировать ситуацию и разобраться в проблеме.

Принятие формальных метрик в качестве одного из основных показателей эффективности творческой деятельности – зло. Вспоминается эпизод, когда в коллектив из кандидатов и докторов наук за год работы создал систему формальных показателей, которая, по их мнению, позволяла оценивать эффективность научной деятельности подразделений научно-исследовательского института. Назывался труд «Методика подведения итогов социалистического соревнования» - около 100 показателей и более 100 страниц методических указаний по их расчету. Не сработало! Уже через три месяца самые «продвинутые» руководители научились работать на методик, а не на науку, и стали получать самые большие премии.

Кто оценивает? В первую очередь, руководство: линейный руководитель и менеджеры проектов, в которых участвовал данный сотрудник. Однако, не менее важны экспертные оценки, которые ставят люди, взаимодействующие с этим сотрудником на разных уровнях, коллеги, смежники, подчиненные, клиенты. В западной литературе этот подход называется оценка по методу «360 градусов».

По результатам аттестации должен быть сделан один из следующих выводов:

- соответствует/не соответствует должности;
- рекомендуется в резерв на должность (это значит, что данный сотрудник займет эту должность без конкурса, как только такая вакансия возникнет);
- рекомендуется на повышение в должности;

Еще один важный и необходимый результат аттестации – план развития, - какую новую ступень карьерной лестницы будет достигать сотрудник и что ему необходимо сделать, чтобы на нее перейти. Что изучить и освоить: новые технологии (согласуем с личными целями и стратегическими потребностями компании), смежные специальности (снижаем риск в случае увольнения коллеги). Каких результатов добиться, например, успешно справиться с ролью руководителя группы разработчиков в проекте N.

Поставленные цели должны соответствовать широко известному в западном менеджменте критерию SMART [47], представляющему собой аббревиатуру из пяти английских слов: Specific (конкретный), Measurable (измеримый), Achievable (достижимый), Relevant (сопоставимый) и Time-bound (определенный во времени). Конкретность цели означает, что всем должно быть понятно, в чем именно состоит цель. Цель должна быть измеримой, чтобы можно было точно узнать, достигнута она или нет. Достижимой — это означает, что имеется достаточно знаний и ресурсов для достижения цели, но и не слишком простой — она должна бросать вызов (challenge) сотруднику. Сопоставимой — цель должна соотноситься с целями компании. Определенной во времени — для каждой цели должны быть четко обозначены временные рамки. Цели, которые одновременно и конкретны, и трудны, и важны, как для компании, так и для самого сотрудника, служат значимым мотивирующим фактором для самоактуализирующегося человека.

И последний вопрос, на который необходимо ответить по результатам аттестации, а надо ли повышать зарплату участнику команды? А если надо, то насколько?

Сколько надо платить программисту?

При решении этого вопроса необходимо помнить, что участник команды это не «винтик», как думают некоторые CEO, который можно легко заменить, купив на рынке аналогичный. Я не утверждаю, что программисты незаменимы. Заменяемы и еще как. На сегодняшний день средняя текучесть кадров в ИТ составляет по моим оценкам 20-30% в год, однако, отрасль не только не рухнула, но и успешно развивается.

Итак, сколько же надо платить программисту? Как правило рыночная «вилка» вознаграждения для данной квалификации известна, допустим от X до 1.5*X. Можно рискнуть и платить по нижней планке X. Однако, возможность получать в 1.5 раза больше за ту же работу, скорее всего перевесит все остальные стимулы, для удержания специалиста в данной компании. Эта ситуация усугубляется еще и тем, что «охотники за головами» делают разрыв в вилке еще больше, чтобы побыстрее перекупить квалифицированные кадры. Надо ли платить по верхней планке, тем более, если она завышена? А, может быть, следует платить еще больше?

Заранее, приношу свои извинения, за занудность и излишнюю подробность нижеследующего изложения в стиле «как для домохозяйек». Сколько раз я не пытался объяснить свое видение подхода к решению этого вопроса людям, которые должны были принимать решение о повышении оклада, они с трудом понимали меня. Может, просто, не хотели?

Программисты заменяемы, но у этой замены есть вполне определенная цена (Рисунок 2).

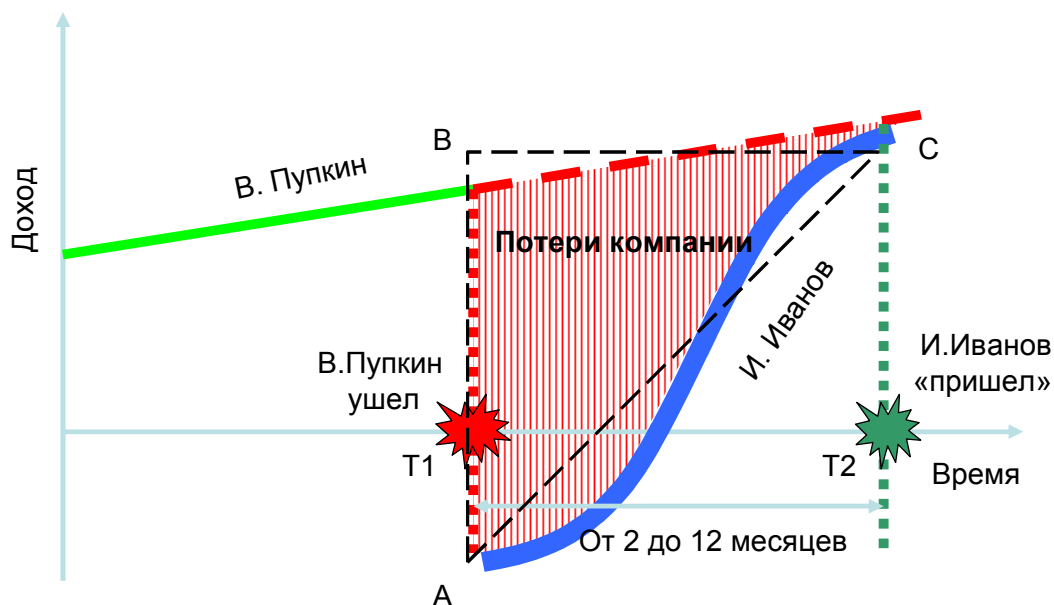


Рисунок 2. Цена замены программиста В.Пупкина на программиста И.Иванова

Определим цену замены программиста В.Пупкина на программиста И.Иванова. Предположим, что Василий успешно работает в вашей компании уже длительное время. Квалификация, опыт и продуктивность его работы постоянно растут, а зарплата, остается на прежнем уровне 2000 у.е. Наконец, Василий оставил вашу компанию в момент T1, но ваша доблестная кадровая служба в тот же момент (!) на ту же зарплату (!!) нашла ему замену в лице И.Иванова, равнозначную по опыту и квалификации (!!!). *Замечу, что это весьма и весьма оптимистичный сценарий.*

Начнет ли Иван приносить с первого дня ту же прибыль компании, что и В.Пупкин? Не начнет. Более того, прежде чем он начнет приносить *хоть какую-то* прибыль, от него будут одни убытки: затраты на внутренне обучение, затраты рабочего времени других участников проекта на консультации и введение его в контекст. И вот, наконец, в момент времени T2 И.Иванов начнет приносить ту же прибыль, что и В.Пупкин. В результате такой замены компания понесет убытки равные площади заштрихованного участка.

Выполним, приближенную оценку для величины этих потерь. Для этого заменим вычисление площади криволинейной фигуры на вычисление площади прямоугольного треугольника ABC. Найдем катет АВ. Сначала найдем величину отрезка между точками T1 и В. Учтем, что себестоимость единицы рабочего времени программиста складывается из его оклада и накладных расходов (зарплата руководства, уборка, аренда, плата за электроэнергию и проч.). Пусть накладные расходы составляют 200% от зарплаты В.Пупкина. Тогда себестоимость рабочего месяца В.Пупкина составит 6000 у.е. Поскольку, мы рассуждаем об успешном бизнесе, то вправе предположить, что его доходность составляет 25%. Следовательно, компания была должна получать ежемесячный доход от работы Василия, где-то, 7500 у.е. Это и есть искомая длина отрезка между точками T1 и В. Для того чтобы оценить отрезок между точками А и T1, предположим, что на начальном этапе адаптации нового сотрудника его коллега с равной зарплатой тратил на помощь ему 20% своего рабочего времени, следовательно компания дополнительно недополучила доход равный $0.2 * 7500$ у.е. Таким образом, длина искомого отрезка между точками А и T1 составляет 1500 у.е. Следовательно, суммарная длина отрезка АВ будет равна 9000 у.е.

Величина катета ВС = T2 – T1, время адаптации зависит от сложности работы, которую делал В. Пупкин и, той роли, которую он играл в проектной команде. Возьмем среднюю величину равную 6 месяцам. Таким образом, по нашей оценке, суммарные потери компании в результате замены

Василия составят величину $9000 * 6 / 2 = 27000$ у.е., равную площади нашего аппроксимирующего треугольника. *И это еще при реализации самого оптимистичного сценария.*

Так, сколько же платить? Пусть текущая стоимость квалификации и опыта В.Пупкина на рынке рабочей силы ИТ оценивается вознаграждением от 2000 до 3000 у.е. Очевидно, что при прочих равных условиях вероятность ухода В.Пупкина к конкуренту обратно пропорциональна его доходу, и изменяется от 1.0 до 0.0 в диапазоне рыночных предложений. Если мы не поднимаем Василию зарплату, то с вероятностью, равной 1, компания в ближайшие 6 месяцев недополучит доход на 27000 у.е. Если мы инвестируем в Васю - наш человеческий капитал, и поднимем ему месячную зарплату до 3000 у.е., то он гарантировано останется работать в нашей компании. При этом мы за ближайшие 6 месяцев потеряем только 6000 у.е.

А что дальше? А дальше очередная аттестация и новый пересмотр договора о найме.

Не следует описанное выше принимать как руководство к действию, как алгоритм расчета конкурентоспособной зарплаты ваших разработчиков ПО. Жизнь гораздо сложнее и многообразнее любой модели. Все сказанное выше, следует воспринимать лишь, как подход к тому, как надо думать над вопросом: «Повышать или не повышать зарплату В. Пупкину?»

Стандарт People CMM

Институт технологий разработки программного обеспечения, SEI долгое время работал над совершенствованием процессов разработки ПО. Результатом этой работы явилось создание модели зрелости процессов (Capability Maturity Model, CMM). В качестве дальнейшего развития этой модели институт предлагает модель оценки уровня персонала компании (People Capability Maturity Model, P-CMM), которая может служить основой стратегического совершенствования в управлении человеческим капиталом в организации.

Билл Кёртис один из авторов модели P-CMM в свое время проводил исследование среди разработчиков и выяснил, что соотношение в уровне квалификации между лучшими сотрудниками и специалистами среднего уровня составляет 20:1. Однако, не стоит рассчитывать в коммерческой разработке ПО на команды из суперзвезд. Приходится работать с коллективом людей средних способностей. Но выбор правильной технологии и развитие квалификаций сотрудников позволяет создать команду, которая оказывается очень производительной и добивается прекрасных результатов. Мы собираем людей с коэффициентом мастерства, равным единице, а не 20, но организуем их так, что они обеспечивают очень высокий уровень эффективности работы.

Базовый подход модели P-CMM - это *осознание ценности каждого работника как личности и необходимости его дальнейшего развития*, что полностью соответствует принципам, изложенным в данной книге.

Подобно CMM, модель P-CMM определяет 5 уровней зрелости компании в части организационного развития и управления персоналом [48].

1. *Начальный.* Отличается расплывчатостью ответственности, низкой вовлеченностью персонала, отсутствием задачи развития персонала у менеджеров. Работа с кадрами ведется, как правило, без осмысления того, зачем это делать.
2. *Повторяемый.* Проводится политика, направленная на развитие способностей персонала.
3. *Определенный.* Внедрение в компании стандартов управления, основанных на лучших практиках работы с персоналом.
4. *Управляемый.* Определяются и вводятся количественные показатели цели в управлении и развитии персонала.
5. *Оптимизируемый.* На основе количественных показателей в компании осуществляется непрерывное улучшение процессов работы с персоналом.

Стратегическими целями разработанной модели P-CMM являются:

1. Увеличение эффективности компании путем повышения квалификации персонала.
2. Обеспечение гарантии того, что высококачественная разработка ПО является отличительной особенностью не отдельных работников, а компании в целом.

3. Обеспечение совместимости между мотивацией отдельного сотрудника и мотивацией организации в целом.
4. Сохранение в организации наиболее ценных и эффективных сотрудников.

Внедрение данной модели в полном объеме, а, тем более, получение формального сертификата, на мой взгляд, относятся к области маркетинговой стратегии и потребуют достаточно больших временных и материальных затрат. Если отвлечься от маркетинга, то большинство организаций используют эту модель для грамотного руководства улучшениями в области развития кадров, а не для сертификации. Примером может служить корпорация Boeing [49], где благодаря внедрению практик People CMM уровня 2 удалось существенно снизить текучесть кадров, улучшить процессы управления производительностью персонала и повысить уровень удовлетворенности сотрудников своей работой.

Заключение или Что надо программисту для счастья?

Программист устроен просто. Он состоит из четырех компонентов: тело, сердце, разум и душа.

1. Телу необходимы деньги и уверенность в завтрашнем дне.
2. Сердцу - любовь и признание.
3. Разуму – развитие и самосовершенствование.
4. Душе – самореализация.

Предоставьте все это вашим сотрудникам, и эффективность их труда возрастет многократно. В противном случае профессионалы найдут все это в другой компании, а в вашей останутся только посредственности.

Работа менеджера проекта подобна труду садовника.

Подобно тому, как садовник любовно отбирает наиболее подходящие растения для своего будущего сада, менеджер набирает людей в команду, наиболее соответствующих целям проекта.

Подобно тому, как садовник ищет лучшую почву для каждого растения с учетом его особенностей, менеджер для каждого участника проектной команды ищет наиболее подходящую для него задачу.

Подобно тому, как садовник тщательно лелеет своих питомцев, оберегает их от вредных воздействий, следит за тем, чтобы ни одно растение не затеняло другое, а только дополняло его и способствовало его росту, менеджер проекта терпеливо работает с каждым участником проектной команды, способствуя его правильному развитию, охраняя от внешних и внутренних потрясений, для того, чтобы максимально раскрыть его индивидуальные способности и увеличить отдачу от них, с удовлетворением отмечает каждое новое достижение.

«Что посеешь, то и пожнешь» - этот закон одинаково применим как к труду садовника, так и к труду менеджера проекта. Пренебрежительное отношение к людям породит лишь ответное пренебрежение. Вложите в людей часть своей души, и вам воздастся сторицей.

PS. Я ничего не сказал о том, что для успеха проекта надо еще составлять календарный план, управлять рисками, требованиями, конфигурацией, качеством, проектировать архитектуру, соблюдать стандарты кодирования, тестировать и документировать систему и много чего еще. Но, во-первых, все это вам, скорее всего, уже давно известно. Во-вторых, наличие детальных диаграмм Ганта пока еще не спасло от катастрофы ни один проект. И, наконец, в-третьих, «разруха не в клозетах, разруха в головах», как говорил профессор Преображенский.

1. Jim Johnson. «Chaos: The Dollar Drain of IT Project Failures. Application Development Trends», January 1995. Standish Group.
2. Алистер Коуберн, Люди как нелинейные и наиболее важные компоненты в создании программного обеспечения, Humans and Technology, Октябрь, 1999 .
3. Стивен Р. Кови, «7 навыков высокоэффективных людей. Мощные инструменты развития личности», 2-е изд., М., Альпина Бизнес Букс, 2007
4. Брукс Фредерик, "Мифический человеко-месяц, или Как создаются программные комплексы", Пер. с англ., СПб., Символ-Плюс, 1999.
5. Кьелл А. Нордстрем, Йонас Риддерстрале, «Бизнес в стиле фанк. Капитал пляшет под дудку таланта», Стокгольмская школа экономики в Санкт-Петербурге, 2005.
6. П. Друкер, «Задачи менеджмента в XXI веке», М., «Вильямс», 2002.
7. А.Маслоу. «Новые рубежи человеческой природы», М., Смысл, 1999.
8. В. Франкл. «Человек в поисках смысла», М., Прогресс, 1990 г.
9. Кэтлин Мелимука, «Разговор с Томом де Марко», Computerworld, #05/1996.
10. Richie O'Bower, «Programming as the best creative specialty», 1997.
11. А. Зубарев «Мифотворцы», 2003 (<http://zay-note.blogspot.com/2007/05/blog-post.html>).
12. С.Рубинштейн, «Основы общей психологии», СПб, «Питер», 2000.
13. Богомаз С.А., «Психологические типы К.Г.Юнга, психофизиологические типы и интертипные отношения», Томский государственный университет, 2000
14. Гуленко В. В., «Менеджмент слаженной команды», 2003 г., Изд.: АСТ, Астрель
15. Б. Шнейдерман, «Психология программирования», М., Радио и связь, 1984.
16. Отто Крегер, Дженет Тьюсон, Хайл Ратледж Типы людей и бизнес. Как 16 типов личности определяют ваши успехи на работе, М., Персей Вече Аст, 1995 .
17. С.Макконнелл, «Профессиональная разработка программного обеспечения», М., «Символ», 2007.
18. Манфред Кетс де Врис, «Мистика лидерства. Развитие эмоционального интеллекта», М., Альпина Бизнес Букс, 2005
19. Р.М. Белбин, «Типы ролей в командах менеджеров». М., HIPPO, 2003.
20. Игорь Ашманов, «Словарь ненормативной лексики программиста», 2002, (<http://www.ashmanov.com/pap/obspro.phtml>)
21. M. Pomeroy-Huff, J. Mullaney, R. Cannon, M. Sebern, «The Personal Software Process (PSP) Body of Knowledge», version 1.0, SPECIAL REPORT CMU/SEI, 2005
22. С. Бобровский, «Ошибкам - бой!», PC Week 144 (20), 1998
23. Joel Spolsky, «The Guerrilla Guide to Interviewing», 2000 (<http://www.joelonsoftware.com/articles/fog0000000073.html>)
24. Аллан Пиз, Язык телодвижений. Как читать мысли окружающих по их жестам, М. Эксмо-Пресс, 2007.
25. Том Демарко, Тимоти Листер, «Человеческий фактор: успешные проекты и команды», Спб. Символ-Плюс, 2005.
26. Harvey, Jerry B. «The Abilene Paradox and other Meditations on Management», Jossey-Bass, 1996
27. Том Демарко, «Deadline. Роман об управлении проектами», М., Вершина, 2006.
28. Л. Томпсон, «Создание команды», М., Вершина, 2005

29. Е. Чинарова, «Team-building, или «Потому что мы - команда!», Журнал «Управление компанией», №3, 2005.
30. О.Платонов «Терновый венец России. История Русского народа в XX веке», Т. I, М., "Родник", 1997.
31. А.Фенько , «Миф о русском коллективизме», Журнал «Власть» №45, 2002 г.
32. Watts S. Humphrey, «The Team Software Process (TSP)», Technical Report CMU/SEI, 2000
33. Higgs, Malcom , «A comparison of the Myers-Briggs Type Indicator and Belbin Team Roles», Henley Management College, 1996
34. Стивен У. Фланнес, Джинджер Левин, «Навыки работы с людьми для менеджеров проектов», М., Технологии управления Спайдер, 2004 г.
35. А. Коуберн, «Каждому проекту своя методология», Humans and Technology Technical Report, TR 99.04, Oct. 1999 (русский перевод http://www.maxkir.com/sd/methyperproject_RUS.htm).
36. Том ДеМарко, Тимоти Листер, Вальсируя с Медведями. Управление рисками в проектах по разработке программного обеспечения, М., Компания р.m.Office, 2005.
37. «РМВОК. Руководство к Своду знаний по управлению проектами», 3-е изд., PMI, 2004.
38. Питер Ф. Друкер «Эффективный управляющий».
39. В. W. Boehm, "Theory-W Software Project Management: Principles and Examples", IEEE Transactions on Software Engineering, Vol. 15, No. 7, July 1989
40. Эдвард Йордан, «Путь камикадзе», М., Лори, 2005.
41. Gerald Weinberg, «Psychology of Computer Programming», Van Nostrand Reinhold, 1971.
42. Абрахам Гарольд Маслоу, Мотивация и личность, М., Евразия, 1999 г.
43. С. Степанов, «В поисках человечности. Абрахам Маслоу (1908—1970)», Школьный психолог, № 30, 2001
44. Эд Салливан, «Время - деньги. Создание команды разработчиков программного обеспечения», Русская редакция, Москва, 2002.
45. Фергус О'Коннэл, «Как успешно руководить проектами. Серебряная пуля», М., КУДИЦ-ОБРАЗ, 2003.
46. И. Ашманов, «Правила Ашманова». Часть 2: об управлении проектами, 2002 (<http://www.ashmanov.com/pap/ashrul2/>)
47. А. Иншакова, Д. Иншаков, ИТ-персонал: оценка, мотивация и развитие, 2007 (<http://www.iemag.ru>)
48. Bill Curtis, William E. Hefley, Sally A. Miller, «People Capability Maturity Model (P-CMM Version 2.0)», Software Engineering Institute, 2002.
49. Sally Miller, Bill Curtis, and William Hefley, Improving Workforce Capabilities with the People Capability Maturity Model, issue of CrossTalk, April 2003.