

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра информатики

Ю.А. Зиссер, А.А. Мелещенко

Управление разработкой и продвижением программного обеспечения

Учебное пособие

по курсу «Менеджмент производства программного
продукта» для студентов специальности 31 03 04
«Информатика»

В 2-х частях

Часть I
Менеджмент

Минск 2002

УДК 681.3.06(075)

ББК 32.973я73

3-64

Рецензент:

президент института приватизации и менеджмента,
кандидат технических наук Д. А. Дичковский

Зиссер Ю.А.

3-64

Управление разработкой и продвижением программного обеспечения: Учеб. пособие по курсу «Менеджмент производства программного продукта» для студентов специальности 31 03 04 «Информатика». В 2-х ч. Ч. I: Менеджмент / Ю.А. Зиссер, А.А. Мелешенко – Мн.: БГУИР, 2002. – 164 с.: ил.

ISBN 985-444-425-2 (ч.1)

Пособие является дополнительным учебным материалом для студентов, изучающих принципы управления разработкой и продвижением проектов программного обеспечения.

УДК 681.3.06(075)

ББК 32.973я73

ISBN 985-444-425-2 -ч.1
А.А., 2002

© Зиссер Ю.А., Мелешенко

ISBN 985-444-421-X

© БГУИР, 2002

Содержание

Введение.....	6
Общие принципы управления.....	8
Успех организации.....	8
Основные понятия.....	10
Кто такие менеджеры и что они делают?	12
Менеджер и предприниматель.....	15
Коммуникации.....	17
Преграды в межличностных коммуникациях.....	19
Стратегическое планирование.....	21
Планирование реализации стратегии.....	24
Принятие решений.....	27
Методы прогнозирования.....	28
Организация взаимодействия и полномочий.....	31
Мотивация.....	34
Теории мотивации.....	35
Контроль.....	41
Группы.....	42
Хоторнские эксперименты.....	42
Неформальные группы.....	44
Как повысить эффективность групп.....	45
Лидерство и власть.....	47
«Лучший стиль» руководства.....	50
Управление и люди.....	51
Управление конфликтами.....	52
Стресс.....	53
Эффективность организации.....	54
Понятие о системе управления производством.....	54
Классификация систем управления производством.....	56
Жизненный цикл изделия и процесса.....	57
Проектирование продуктов и процессов в сфере услуг.....	59

Управление проектами и качеством.....	60
Управление проектами.....	60
Системы качества и стандарты качества.....	65
Управление проектами разработки программного обеспечения.....	68
Законы Брукса.....	68
Мифический человеко-месяц.....	70
Концептуальное единство.....	72
Эффект второй системы.....	74
Упрощение концептуальной сложности.....	75
«Серебряной пули нет»	76
Стратегия открытых исходных текстов как альтернатива концепции Брукса.....	78
Классические ошибки управления проектами.....	79
Планирование быстрой разработки.....	87
Жизненный цикл разработки программного обеспечения.....	90
Оценка трудоемкости.....	95
Типичные причины оптимистических планов.....	97
Методики оценки трудоемкости выполнения проекта.....	97
Оценка объема работ по функциональным баллам.....	97
Оценка длительности проекта по трудоемкости.....	100
Оценка, основанная на рисках.....	100
Оценка по случаям.....	101
Оценка по вероятности.....	101
Общие рекомендации по оценке сроков.....	102
Как вести исторические записи.....	104
Разработка, ориентированная на заказчика.....	105
Мотивация труда программистов и их менеджеров.....	107
Гигиенические факторы для программистов.....	109
Управление функциональностью.....	112
Средства повышения производительности.....	114
Как «вытянуть» безнадежный проект.....	116

Испытанные приемы быстрой разработки.....	118
Ежедневный выпуск прототипа.....	118
Эволюционное прототипирование.....	119
Самопожертвование.....	119
Прототипирование с выбрасыванием.....	120
Разработка в срок.....	120
Внешняя закупка ПО.....	120
Договор подряда.....	123
Как сделать подряд успешным.....	126
Проблемы менталитета.....	128
Переписка.....	130
Приложение 1: Разработка резюме.....	136
Приложение 2: Собеседование при приеме на работу.....	147
Указатель имен.....	153
Литература.....	163

Введение

Зачем программистам нужно изучать менеджмент и маркетинг? Для ответа на этот вопрос задайте себе еще один: часто ли Вы сталкивались с 50-летними или даже 40-летними программистами? Вроде нет. Так куда же они подевались? Правильно, после не очень продолжительной карьеры программиста они в свои 35–40 лет стали начальниками разных рангов, и теперь на 100% времени заняты управлением. Следовательно, наука управления для программиста очень нужная, ибо большую часть производственной биографии программисту предстоит вовсе не программировать, а руководить!

Далее. По статистике 10% населения становятся предпринимателями, владельцами собственного бизнеса. В этом случае, кроме менеджмента, важнейшей составной частью является продвижение товаров и услуг на рынок. По той же статистике 80% предприятий гибнут в первые 4 года своего существования, и основная причина их гибели – неправильное продвижение товаров и услуг и ошибки в управлении.

Специалистов по маркетингу у нас выпускают пока чрезвычайно мало и без выраженной специализации. Научные методы пока используются недостаточно. Чаше предприятия, ссылаясь на якобы неприменимость испытанных методов маркетинга и менеджмента в отечественных условиях, действуют методом «тыка», наступая на одни и те же «грабли» и набивая шишки на классических ошибках, описанных во всех учебниках.

В наших рекламных агентствах работают хоть и талантливые, но все же бывшие инженеры, учителя и т.д. Там нет ни одного специалиста с университетским дипломом по маркетингу или менеджменту. Напомню, что в США большая часть руководителей всех рангов имеет степени MBA (Master of Business Administration).

Любительский подход к управлению и маркетингу не может не сказываться на состоянии отечественной экономики. Недостаточно выпускать лучшие в мире товары, полагая, что «хороший товар сам найдет своего покупателя» – это классическая ошибка. Ведь потенциальные покупатели часто даже не догадываются о его существовании и достоинствах.

Это пособие нужно, в первую очередь, следующим категориям читателей.

1. Руководителям небольших программистских предприятий или подразделений, которые вынуждены заниматься всем сами – организацией производства, маркетинга и т.д.
2. Менеджерам и специалистам отделов маркетинга программистских предприятий, которые хотят управлять профессионально.
3. Студентам по компьютерным наукам.
4. Всем, кто так или иначе связан с B2B-маркетингом, к которому применимо большинство описанных в пособии принципов.

Общие принципы управления

Успех организации

Сначала рассмотрим несколько классических историй успехов и неудач организаций. По мере чтения попытайтесь догадаться, что это за компании.

История первая. Студент 3-го курса университета в целях самообразования решил в одиночку переписать одну всемирно известную крупную программную систему, имеющую тысячи внедрений. Конечно, он не справился с работой самостоятельно, но сумел заинтересовать других программистов-любителей со схожими интересами, которые также стали работать над системой совершенно бесплатно в свободное от основной работы время. Может ли из такого самодеятельного, любительского проекта родиться серьезный бизнес и новая эпоха в софтверном бизнесе? Может, если речь идет о Linux и финне Линусе Торвальдсе.

История вторая. Француз, учившийся программированию в Швейцарии, самостоятельно написал компилятор примитивного учебного языка программирования. Язык был разработан профессором выпускавшей его кафедры для использования в учебном процессе – для реализации фрагментов компилятора студентами в качестве курсовых и дипломных проектов. В языке намеренно не было сложных конструкций и операций ввода-вывода, что делало почти невозможным промышленное использование языка. С этим компилятором молодой выпускник уехал в США и попытался организовать его промышленный выпуск по цене в 10 раз ниже среднего уровня цен на компиляторы с других языков программирования. Может ли быть успешной столь безумная идея? Имеет ли шансы новый язык программирования? Да, может – ведь речь идет о Филиппе Кане, языке программирования Pascal и фирме Borland International.

История третья. Молодой программист из Санкт-Петербурга на досуге написал простую почтовую программу, позволяющую работать с электронной почтой через Web-интерфейс. До него в мире были разработаны и широко применялись десятки, если не сотни аналогичных систем. Что

могло из этого получиться? Самая популярная в СНГ бесплатная почта Mail.Ru.

История четвертая. Одна компания является гигантом и имеет консервативный имидж в обществе. Еще совсем недавно предписывалось, чтобы все, включая ремонтных рабочих, появлялись на работе в белых рубашках и галстуках. За почти столетие своего существования компания никогда не считалась научно-техническим лидером. Между тем ее цены на 25% выше цен конкурентов. Низкие планы объемов реализации приводят к тому, что планы менеджеров по продажам почти всегда выполнены, премии выплачиваются регулярно. Есть ли шансы у такой компании? Правильный ответ – «да», ибо речь идет о компании IBM.

История пятая. Два человека, бросивших колледж и продававших устройства для незаконных бесплатных междугородних разговоров, разработали компьютер за 1300 долл. Штаб-квартира этой компании располагалась в спальне одного учредителя, а сборочный конвейер – в гараже другого. Как Вы думаете, что это за фирма? Представьте себе, Apple.

История шестая. Фирма Apple некогда добилась выдающихся успехов в выпуске первых промышленных персональных компьютеров с защищенной лицензиями архитектурой и собственным закрытым программным обеспечением, недоступным для лицензирования другими производителями. Однако некая фирма А разработала конкурирующую архитектуру и наняла фирму В разработать процессор, а фирму С – операционную систему к ней. После выпуска компьютера фирма А сразу разрешила всем желающим клонировать свои компьютеры, за что и поплатилась, ошибочно решив не взимать за это лицензионные отчисления. Фирма В успешна, но сталкивается с жесткой конкуренцией, а фирма С, воспроизведя часть возможностей Apple, лицензирует операционную систему и программные продукты на чрезвычайно льготных условиях всем производителям, пожелавшим сделать это. В результате сегодня фирма Apple занимает всего лишь около 1% рынка, а ее бизнес несоизмеримо меньше соответствующего бизнеса ее конкурентов – IBM, Intel и Microsoft.

История седьмая. В 2000-2001 гг. закрылись тысячи Интернет-компаний, имевших отличное финансирование, великолепный менеджмент и маркетинг, укомплектованный, хорошо подготовленный штат сотрудников, высокие оценки биржевых аналитиков.

Еще в 1994 г. двое никому не известных аспирантов создали каталог всех имеющихся на тот момент нескольких сотен ресурсов в видимой им части межуниверситетской компьютерной сети. И что же дальше? Может быть, кто-то уже догадался? Это компания Yahoo! Inc., владеющая самым крупным, самым посещаемым в мире порталом, на котором 35% американцев проводит более 2/3 своего времени в Интернете.

История восьмая. Мир меняется, менеджеры должны принимать решения быстро. Один человек не может руководить тысячей подчиненных. За одним исключением. Есть одна организация, пожалуй, самая старая на Земле. Она насчитывает всего четыре уровня иерархии и существует свыше полутора тысяч лет. В ней полностью отсутствует демократия, решения принимаются корректно с опозданием в десятки лет. Что это за организация? – Конечно же, Римская католическая церковь.

Можно в каждом из вышеприведенных случаев что-то объяснить, но на самом деле простых ответов нет. В бизнесе не существует путей и средств, пригодных во всех ситуациях.

Большинство организаций рассчитывает жить вечно, хотя ни одна организация не существовала всегда. Билл Гейтс как-то говорил о Microsoft, что понимает: гибель его компании – вопрос времени, но видит свою задачу в том, чтобы отдалить это печальное событие, насколько это возможно.

Основные понятия

Организация – это группа людей, деятельность которых сознательно координируется для достижения общей цели или целей. Иными словами, для существования организации необходимо наличие хотя бы двух человек, общая цель и совместная деятельность.

Цели – это конечные состояния организации, желаемый результат, которого стремится достичь группа, работая вместе. Цели подразделения должны вносить конкретный вклад в цели организации как целого и не вступать в противоречие с целями других подразделений. Поскольку жизнь меняется, любой организации время от времени приходится менять свои цели.

Организация считается успешной, если она достигла своей цели. Размер организации или ее финансовый оборот не являются критерием успешности. Например, в России есть два наиболее известных в СНГ разработчика бухгалтерского программного обеспечения – компании «1С» и «Парус». Обе компании выпускают примерно один и тот же ассортимент продукции, тираж программ и объем сбыта сопоставимы, оба предприятия прибыльны. Однако штат 1С на порядок меньше, соответственно эффективность этого предприятия на порядок больше.

Необходимо различать эффективность и результативность работы организации. П.Друкер говорит, что результативность – это когда делаются правильные вещи, *doing the right things*, а эффективность – когда вещи создаются правильно, *doing things right*. Успешным считается решение, которое реализуется результативно и эффективно.

Задание – это предписанная работа, серия работ или часть работы, которая должна быть выполнена заранее установленным способом в заранее оговоренные сроки. Задание предписывается работнику, а не его должности.

Горизонтальное разделение труда – разделение работы на составляющие компоненты. Классический пример – разбиение предприятия на производство, маркетинг и финансы. Разделения труда может и не быть вовсе – например, в небольшом кафе, где хозяин все делает сам. При горизонтальном разделении труда возникают подразделения, «организации в организациях», которые имеют свои собственные цели и задачи. Таким образом, большие и сложные организации состоят из нескольких специально созданных подразделений и многочисленных неформальных групп, возникающих спонтанно. Примером может служить БГУИР. Его подразделения – это факультеты – ФКСиС, ФИТУ, ФТК и др. Уже после

нескольких совместных занятий потока студентов, как Вы и сами замечали, внутри него сами по себе образуются несколько сплоченных неформальных групп.

Вертикальное разделение труда. Поскольку работа в организации состоит из многочисленных задач, кто-то должен координировать работу группы для того, чтобы она была успешной. Вертикальное разделение труда отделяет работу по координированию действий от самих действий. Деятельность по координированию работы других людей и составляет сущность управления.

Кто такие менеджеры и что они делают?

Как мы узнали, все организации имеют между собой что-то общее. Неудивительно поэтому, что управленческая работа также имеет много общих характеристик. Руководитель – есть руководитель, и хотя организации, сферы и ответственность могут различаться, работа премьер-министра имеет много общего с работой цехового мастера.

Почти всякая работа требует специализации и носит непрерывный, повторяющийся характер. В то же время для менеджера характерны кратковременность, разнообразие и фрагментарность осуществляемой деятельности. Подсчитано, что мастера на производстве выполняют в среднем 583 операции управления за день, средняя продолжительность управленческой операции – 48 секунд, а половина их рабочего времени проходит в общении.

Г. Минцберг пишет, что руководитель действует как система обработки информации. Руководитель – это нервный центр (или, если угодно, вычислительный центр), на входе которого – информация из внешних и внутренних источников, а на выходе – распространение информации для подчиненных и внешнего окружения.

Управление – это процесс планирования, организации, мотивации и контроля, необходимый для формулирования и достижения целей организации.

Управление так или иначе присутствует в организации. Работник может и не управлять другими людьми, управленческие функции не всегда выделены четко, но основная функция – координирование – выполняется. Даже в крупных организациях руководители часто выполняют неруководящую работу – например, занимаются продажами. Президент IBM вынужден сам работать с наиболее крупными клиентами – например, Министерством обороны США, которое не станет разговаривать ни с кем ступенькой ниже. Но это скорее исключение, чем правило, ибо для успеха надо отделять управленческие функции от других.

В сравнительно больших организациях существует несколько уровней управления. Обычно уровень отражается в названии должности, но название не всегда четко отражает уровень должности. Например, «капитан» в армии – младший офицерский состав, во флоте – старший. Региональный менеджер по продажам – это обычный продавец. За термином «руководитель низового звена» скрываются бригадиры, младшие начальники. Это может быть сержант, мастер смены или, например, заведующий кафедрой информатики. Большая часть руководителей вообще – это руководители низового звена. Большинство людей начинают свою управленческую карьеру в этом качестве.

Руководители среднего звена возглавляют крупное подразделение или отдел в организации. Они представляют своего рода буфер между руководителями высшего и низового звеньев. Исследования показали, что руководители среднего звена проводят около 89% времени в устном общении. Типичными должностями руководителей этого уровня являются: заведующий отделом, декан, офицеры от лейтенанта до полковника. Директор филиала АСБ «Беларусбанк», например, – менеджер среднего звена.

Руководители высшего звена отвечают за принятие важнейших решений. Даже в самых крупных организациях руководителей высшего звена – всего несколько человек. Сильные руководители этого уровня накладывают отпечаток своей личности на весь облик организации. Поэтому успешно действующие руководители высшего звена в больших организациях ценятся очень высоко.

Типичные должности на этом уровне – это Председатель Совета учредителей, Президент, вице-президент, главный бухгалтер. В армии это – генерал, в государственных органах – министры, в университете – ректор.

Минцберг пишет: «Человек на руководящей должности очень одинок. Количество работы, которую приходится выполнять, огромно, а темп очень напряженный. И после долгих часов такой нагрузки главный руководитель не в состоянии уйти от своей среды ни физически, ни в своих мыслях, которые нацелены на непрерывный поиск новой информации».

Основная причина такого состояния руководителя в том, что его работа не имеет четкого завершения. Не существует момента, когда можно было бы считать работу законченной, поэтому нельзя быть уверенным в успешности своей деятельности. Внешняя среда непрерывно меняется, и риск неудачи неизменно высок.

По статистике, загруженность руководителя этого уровня составляет 60 – 80 часов в неделю, в том числе: встречи – 60%, поездки, осмотры – 3%, работа с бумагами – 22%, телефон – 6%.

Пять типичных причин неудач в карьере руководителей по Дж.Грехему следующие.

Их толкает вверх потребность получать более высокую зарплату. Само по себе желание хорошо зарабатывать естественно, но, доминируя над всеми остальными целями, оно наносит делу вред.

Их слишком волнуют символы их положения. Плохой руководитель делает себе большой кабинет с евроремонтом, обставляет его дорогой мебелью, вешает на дверь массивную табличку, ездит только на престижных автомобилях, приставляет к себе шофера и даже приобретает самолеты. Опять же, само по себе это отнюдь не плохо. Плохо тогда, когда руководитель уделяет этим второстепенным моментам слишком много внимания. Положительный контрпример: руководители Intel, Bloomberg и многих других передовых компаний сидят в общем зале вместе с другими сотрудниками.

Их волнуют прежде всего собственные персоны. Плохой руководитель беспокоится прежде всего о себе, в то время как хороший – о своих людях. Просьба сделать параллельный телефон для того, чтобы работницы могли позвонить своим детям, воспринимается плохим руководителем как просьба дать ключ от сейфа. Плохие руководители также любят присваивать себе лавры подчиненных.

Они самоизолируются, избегают общаться с подчиненными, ограничивают доступ в свой кабинет, не выслушивают подчиненных, не прибегают к их помощи в процессе выработки решений.

Они скрывают свои мысли и чувства, стремятся спрятать свои гнев и страхи, а это всегда видят подчиненные. Неискренность – спутник плохого руководителя.

Менеджер и предприниматель

Предприниматель – это человек, который берет на себя риск, связанный с организацией нового предприятия, разработкой новой идеи или нового вида услуг, предлагаемых обществу. Важно понимать отличие между менеджером и предпринимателем: предприниматель берет на себя лично риск за успех организации, менеджер – нет.

Предпринимательство не ограничивается бизнесом: так, несомненно, яркими предпринимателями были Петр I, Наполеон Бонапарт, Ленин и многие другие.

Некоторые предприниматели оказываются весьма успешными руководителями. Основатель компании IBM Томас Уотсон разработал и внедрил многочисленные приемы в практику управления, которые позволили превратить IBM в одного из лидеров компьютерной индустрии. Другие примеры – Рэй Крок, McDonalds; Билл Гейтс, Microsoft. Однако такие характеристики, как готовность идти на личный риск, желание долго и упорно работать, не считаясь с отдыхом, т.е. все то, что обычно считается чертами хорошего предпринимателя, совсем не обязательно говорит о способности того же человека управлять большой организацией. Таким образом, нет ничего странного в том, что выдающийся предприниматель

становится посредственным менеджером. Так, Филипп Кан, о котором уже говорилось как об успешном предпринимателе, став главой крупной корпорации, почувствовал себя «не в своей тарелке». Он устранился от решения важных проблем корпорации, часами играя на флейте в собственном рабочем кабинете. В какой-то момент ему так надоела его работа, что он своими действиями начал расшатывать основы бизнеса своей компании, вопреки здравому смыслу затеяв заведомо проигрышную ценовую войну с ненавистными лично ему Microsoft и Биллом Гейтсом.

Речь идет о компиляторе модного в то время языка программирования Pascal, который в начале 90-х годов обе фирмы продавали по примерно одинаковой цене. На тот момент корпорация Microsoft была лишь в полтора раза крупнее Borland, а Windows была редко используемой оболочкой-надстройкой над MS-DOS. Снижение цены компилятора Turbo Pascal вдвое при несколько более высоком качестве компилятора позволило Кану одержать личную победу над Гейтсом: компилятор Microsoft Pascal в считанные месяцы был навсегда вытеснен с рынка. При этом Microsoft потерпела небольшой ущерб, ибо доходы от продаж компилятора Microsoft Pascal занимали лишь малую долю в доходах этой корпорации.

Однако это была пиррова победа. Финансовая база Borland оказалась подорванной, ибо доходы от реализации компилятора Turbo Pascal изначально были главным источником поступлений компании. В результате некогда высокорентабельная компания стала убыточной, стало не хватать ресурсов для разработки новых средств программирования (например, Delphi), курс акций компании на биржах упал, и совет акционеров был вынужден сместить Кана с поста руководителя. Кан распрощался с Borland, получив свою долю капитала в виде некогда известного DOS-органайзера SideKick.

Сегодня Кан руководит небольшой компанией StarFish Software, которая успешно занимается разработкой средств связи органайзеров персональных компьютеров с сотовыми телефонами. Компакт-диски с соответствующим пакетом программ этой фирмы сегодня прилагаются к приобретаемым сотовым телефонам с инфракрасным портом.

Организации, действующие в условиях неустойчивой среды, просто не могут дожидаться изменений и затем на них реагировать. Их руководители должны думать и действовать как предприниматели. Предприимчивый менеджер активно ищет новые возможности и намеренно рискует, добиваясь их реализации.

Предприимчивость требуется на каждом уровне, чтобы организация в целом работала как предприниматель. И чем больше размеры организации, тем труднее этого добиться. Многие полагают, что успех IBM объясняется тем, что ей удается быть предпринимательской, несмотря на громадные размеры. Это значит, что стратегические решения высшего руководства основаны на мыслях, высказываемых на среднем и нижнем уровне. Для этого поощряется свобода действий и высказываний в плане совершенствования работы и повышения эффективности своей организации. Организация, в которой средние менеджеры боятся высказаться, обречена на техническую отсталость.

Коммуникации

«Я не разделяю выдвигаемых продавцами видений «интегрированной автоматизированной системы управления», в которой руководитель вводит запрос с клавиатуры, а на экране вспыхивает нужный ему ответ. Есть множество фундаментальных причин, по которым этого никогда не произойдет, и одна из них состоит в том, что только малая часть, возможно, 20% времени руководителя занята задачами, которые требуют сведений, отсутствующих в его памяти. А все остальное время – это общение: слушать, отчитываться, обучать, убеждать, советовать, ободрять», – пишет Ф.Брукс в 10-й главе своей знаменитой книги «Мифический человеко-месяц».

От 50 до 90% своего времени руководитель тратит на коммуникации. Коммуникации – это связующий процесс, ибо он встроен во все процессы управленческой деятельности. Опросы показали, что 73% американских, 63% английских и 85% японских руководителей считают неэффективные коммуникации основным препятствием на пути достижения успеха их организациями – главным источником проблем.

Эффективно работающие менеджеры – это те, кто эффективен в коммуникациях. Они представляют суть коммуникационного процесса, обладают хорошо развитым умением устного и письменного общения и понимают, как среда влияет на обмен информацией. Речь идет, конечно, прежде всего о формальных коммуникациях.

Слухи – это канал неформальных коммуникаций. Слухи «витают возле автоматов с охлажденной водой, по коридорам, в столовых и в любом другом месте, где люди собираются группами». Поскольку по каналам слухов информация передается намного быстрее, опытные руководители пользуются ими для запланированной утечки информации или сведений типа «только между нами». Приведем выдержку из статьи W.A.Delaney «The Secretarial Grapevine», показывающую, как можно использовать слухи.

«Как только я выявил каналы распространения слухов в конторе, я быстро понял, как их использовать. Как-то раз Сара, мой секретарь, предупредила меня, что на следующем совещании один из управляющих собирается предложить реорганизацию, в результате которой мой отдел перейдет в его подчинение. Он был из породы «создателей империй», а я должен был стать одной из его жертв. Загодя получить предупреждение означает загодя вооружиться для отпора, поэтому я спокойно спланировал свою контрстратегию. Я обратился к «создателю империй» с предложением передать его отдел в мое подчинение. Он так мучался, объясняя, почему этого нельзя делать, что даже не дошел до вопроса относительно моего отдела. Потом я попросил Сару сообщить по каналам распространения слухов, что если этот руководитель отстанет от моего отдела, я сделаю то же самое по отношению к нему. Мы с ним ни разу не разговаривали с глазу на глаз и вслух не признавались в существовании конфликта. Однако по каналам распространения слухов ко мне пришел ответ: «Мир»; и наши отношения оставались мирными все время, пока он и я работали в этой организации».

Вопреки устоявшемуся мнению, слухи оказываются на 80-99% точными в отношении компании или деловой информации. Однако точность слухов в отношении отдельных личностей может быть значительно ниже. Типичные слухи – это сокращения, наказания, изменение организационной

структуры, перемещения и повышения, скандалы наверху, свидания после работы.

Начальству всегда кажется, что процесс коммуникаций более эффективен, чем это видится со стороны подчиненного. Статистика подсказывает, что только 50% контактов считаются эффективными. Чрезвычайно важно понять, что информация, противоречащая собственному опыту воспринимающего, отторгается или искажается. Человек воспринимает мир в рамках собственной парадигмы. Поэтому для руководителя исключительно важно иметь представление о мировосприятии работника и говорить на понятном ему языке.

Первый Президент Республики Беларусь Александр Лукашенко – великолепный коммуникатор, выступления которого регулярно и внимательно слушают как его сторонники, так и противники. Он единственный из белорусских политиков *говорит* с народом, причем на *его* языке, благодаря чему достигнут потрясающий контакт руководства страны с народными массами.

Преграды в межличностных коммуникациях

Преграды можно классифицировать следующим образом.

1. *Преграды, обусловленные восприятием.* Как мы уже сказали, информация, противоречащая собственному опыту человека, отторгается либо искажается.

2. *Семантические барьеры.* Семантика изучает способ использования слов и значения, передаваемые словами (омонимы, толкования). Символ не имеет уникального значения и зависит от контекста. Например: «Сделайте, как только представится удобная возможность» может означать «как можно скорее» или «когда будет время». Автомобиль «Жигули» продвигался за рубежом под маркой «Лада», ибо на французском языке «жиголо» означает «альфонс», «мужчина на содержании женщины». Попытка General Motors по продвижению внедорожника Chevy Nova на латиноамериканский рынок провалилась по единственной причине: nova по-испански означает «не едет».

3. *Невербальные преграды.* Недавние исследования дали шокирующую информацию о том, что 55% сообщений воспринимается через выражение лица, позы и жесты, 38% – через интонации и модуляции голоса. Только 7% информации передается словами.

Как следует из этих цифр, невербальные символы часто полностью подавляют вербальные. Представьте, что Вы входите в кабинет Вашего начальника, чтобы получить данные по проекту, над которым работаете. Вы вошли, а он несколько секунд продолжает рассматривать бумаги у себя на столе. Затем смотрит на часы и говорит отчужденным голосом: «Чем могу быть Вам полезен?». Хотя его слова сами по себе не несут негативного смысла, язык поз и жестов ясно указывает, что Вы для него – нежелательное явление. Представьте теперь, что при Вашем появлении в кабинете, руководитель, напротив, сразу поднимает на Вас взгляд, приветливо улыбается и бодрым тоном говорит: «Как продвигается проект? Чем могу быть Вам полезен?»

Эта тема рассмотрена в известной книге Алана Пиза "Язык жестов".

Совет по ходу: протянутую Вам визитную карточку надо читать, улыбаться и смотреть в глаза, а не мять пальцами.

По мнению антрополога Эдуарда Холла, Ясир Арафат носит темные очки, чтобы окружающие не могли наблюдать за его реакциями по расширению его зрачков – арабы это знают сотни лет.

4. *Плохая обратная связь.* Об этом подробнее говорится ниже.

5. *Неумение слушать.* Руководители слушают с 25%-ной эффективностью. Те руководители, которые слушают подчиненных, уважаемы. О таких подчиненные говорят: «Мне нравится мой начальник – он слушает меня».

Совершенствование искусства общения

- * Проясняйте свои идеи до начала общения.
- * Будьте восприимчивы к потенциальным семантическим проблемам.

- * Следите за языком собственных поз, жестов и интонациями.
- * Излучайте сопереживание и открытость.
- * Добивайтесь установления обратной связи.

Вот 10 правил профессора Кит-Дэвиса по искусству эффективно слушать.

1. Перестаньте говорить.
2. Помогите говорящему раскрепоститься.
3. Покажите говорящему, что Вы готовы слушать.
4. Устраните раздражающие моменты.
5. Сопереживайте говорящему.
6. Будьте терпеливы.
7. Сдерживайте свой характер.
8. Не допускайте споров или критики.
9. Задавайте вопросы.
10. И самое главное – перестаньте говорить!

Стратегическое планирование

Стратегическое планирование – это набор действий и решений руководства, которые ведут к разработке конкретных стратегий, предназначенных для достижения организацией своих целей.

Стратегическое планирование дает основу для принятия решения. Знание того, что организация хочет достичь, помогает находить наиболее подходящие способы действий. Принимая обоснованные и систематизированные плановые решения, руководство снижает риск принятия неправильного решения из-за ошибочной или недостоверной информации. Стратегическое планирование необходимо также для правильного распределения ресурсов, адаптации к внешней среде, улучшения внутренней координации.

Между планированием и успехом организации существует устойчивая связь. В одном исследовании была проанализирована деятельность 105 крупнейших компаний США и 105 средних и малых фирм. Как выяснилось, фирмы, применяющие у себя формальное планирование, имеют самые высокие темпы роста.

Важнейшую роль в стратегическом планировании играет организационное стратегическое предвидение. Визионеры, или провидцы (visionary) – люди, которые умеют предвидеть будущее лучше других. Особенно важны такие люди в высоких технологиях. Яркий пример визионера – Билл Гейтс, бросивший университет ради перспектив зарождающегося рынка программного обеспечения для микрокомпьютеров.

Стратегия – это детальный всесторонний комплексный план, предназначенный для того, чтобы обеспечить осуществление миссии организации и достижение ее целей с учетом перспективы.

Миссия (основная общая цель) организации – это четко выраженная причина ее существования. Цели вырабатываются для осуществления этой миссии. Миссия должна соответствовать масштабу и размерам организации. Важно не выбрать сложную миссию для малой организации.

«Получать прибыль» – это не миссия, а одна из задач предприятия, тем более, что источник прибыли находится во внешней среде. Для формулирования миссии необходимо ответить на вопросы «Кто наши клиенты?» и «Какие потребности наших клиентов мы можем удовлетворить?». Ограничиваться прибылью – значит ограничивать возможности, упускать ключевые факторы и снижать эффективность организации. В Японии задачи вообще не определяют в терминах прибыли. Там все определяют в терминах долгосрочной перспективы, когда главное – это рост.

В качестве примера приведем правильно сформулированную миссию Sun Banks, одного из крупнейших финансовых учреждений Америки: «Миссия компании Sun Banks заключается в содействии экономическому развитию и благосостоянию сообществ, обслуживаемых компанией, путем предоставления гражданам и предприятиям качественных банковских услуг

таким образом и в таком объеме, которые соответствуют высоким профессиональным и этическим стандартам, обеспечения справедливой и соответствующей прибыли акционерам компании и справедливого отношения к сотрудникам компании».

Еще пример: миссия белорусской компании «Надежные программы» звучит так: «Путем внедрения современных компьютерных технологий сделаем жизнь людей лучше».

Чтобы миссия приносила пользу, она должна быть официально сформулирована, сотрудники организации должны знать и хорошо понимать ее. Тогда миссия служит своего рода ориентиром, на котором руководители и рядовые сотрудники основывают свои решения.

У организации также должен быть слоган (краткий лозунг), в афористической форме выражающий миссию фирмы и конкретизирующий ее. Время жизни слогана короче, чем у миссии, и каждые несколько лет слоган может меняться в зависимости от текущей ситуации. Примеры слоганов: у Apple: Think different («Думай иначе», ибо основной ценностью продукции Apple считаются технологические новинки и нетрадиционные решения), у Dell: Be direct! («Будь прямым!», ибо основной бизнес-принцип Dell – прямые продажи компьютеров непосредственно потребителям), у «Надежных программ»: с 1992 года – «Надежным банкам – Надежные программы», а с 1997 года в связи с выходом на новые рынки слоган изменился: «Надежным клиентам – Надежные программы». В 2002 году в связи с переориентацией на корпоративные решения, преимущественно связанные с Интернет, слоган изменился в очередной раз и стал выглядеть так: «Центр электронного бизнеса». Для продвижения этого слогана даже был открыт отдельный сайт www.tutby.com. При этом миссия компании в целом осталась неизменной.

Существует также еще один носитель миссии фирмы – ее логотип, в графической форме выражающий миссию фирмы.

Главная цель высшего руководства – управлять ценностями организации. В то же время важнейшую роль в формулировании и осуществлении миссии играют ценности самого руководства.

Как правильно формулировать цели?

1. Цели должны быть *конкретны* и *измеримы*.
2. Цели должны быть ориентированы во времени.
3. Цели должны быть достижимы. Установление цели, которая превышает возможности организации, может привести к катастрофе.
4. Цели должны быть взаимно поддерживающими. Цели подразделений должны вносить конкретный вклад в цели организации как целого и не вступать в противоречие с целями других подразделений.

Для правильного формулирования цели необходимо ответить на следующие вопросы:

1. Где сейчас находится организация?
2. Где она должна находиться в будущем?
3. Что делать?

После установления своей миссии и целей руководство должно начать изучение внешней среды с целью определения угроз и возможностей для фирмы. Необходимо проанализировать экономические, политические, рыночные, технологические, международные факторы, а также факторы конкуренции и социального поведения. После этого руководство пытается определить сильные и слабые стороны самой организации. Методической оценке должны подвергнуться маркетинг, финансы, операции (производство), кадры, а также культура и образ компании. Приведя внутренние силы и слабости в соответствие с внешними угрозами и возможностями, руководство готово к выбору правильной стратегии. Стратегический выбор должен быть однозначным и окончательным.

Планирование реализации стратегии

После выбора общей стратегии ее необходимо реализовать. Главной задачей руководства при этом является обеспечение согласованных действий всех сотрудников организации. Легко может случиться так, что работники, движимые даже наилучшими намерениями, не работают над достижением

целей, а лишь отдаляют их. Чтобы избежать этого, руководство должно разработать тактики, политики, процедуры и правила.

Тактика – это краткосрочная стратегия. Например, конкретный игровой ход представляет собой тактику, согласованную с общей стратегией игры.

Политика представляет собой общее руководство, облегчающее достижение целей, необходимое для действий и принятия решений.

Процедура описывает действия, которые следует предпринять в конкретной ситуации.

Правило (инструкция) точно определяет, что должно быть сделано в специфической единичной ситуации. Для того чтобы правило работало, необходимо всегда в обязательном порядке объяснять, почему оно принято, в противном случае правило ждет отторжение и саботаж, оно просто не будет работать.

Планы помогают распределять действия, связанные с использованием ресурсов, однако они не дают ответа на следующие вопросы: какие именно ресурсы имеются, как их следует использовать, какие цели являются достижимыми при имеющихся ресурсах. Чтобы решить эти вопросы, используются бюджеты.

Бюджет – это метод распределения ресурсов в количественной форме для достижения количественных целей. Кроме функций планирования бюджет представляет собой еще и эффективное средство контроля.

Управление по целям. Популярный метод управления по целям основан на том, что производительность людей, имеющих конкретные цели, выше производительности тех, кому цели не установлены или кого просто просят стараться. Кроме того, согласно исследованиям, предоставление людям информации о результатах их работы повышает их производительность.

Согласно этому методу каждый руководитель в организации, от наивысшего до самого нижнего уровня, должен иметь четкие цели, которые поддерживают цели вышестоящих руководителей. Это дает возможность оценивать руководителя на основе результатов, а не личных качеств.

Управление по целям состоит из четырех взаимозависимых этапов:

1. Выработка четких, кратких целей.
2. Планирование действий по их достижению.
3. Систематический контроль и оценка работы.
4. Принятие корректирующих мер при выявлении отклонений.

Цели работы руководителя			
Иван Петров (подпись)	2.07	Руководитель отдела разработки ПО	
Подготовлено руководителем	Дата	Должность руководителя	
Петр Сидоров (подпись)	2.07	Президент фирмы	
Проверено контролирующим лицом	Дата	Должность контролирующего лица	

Формулировка целей	Приоритет	Срок исполнения	Итоги или результаты
1. Снизить число жалоб на низкое качество работ со стороны заказчиков — до 3 в год	A	31.01	
2. Снизить число случаев задержки ответов на письма клиентов более чем через 24 часа после получения — до 5 в год	A	31.01	
3. Обеспечить представление каждым программистом и аналитиком ежесуточной отчетности за исключением дней отсутствия	A	31.06	
4. Самостоятельно изучить книгу Ф.Брукса "Мифический человеко-месяц"	B	15.10	
5. Прочитать сотрудникам цикл лекций о том, как выполнить проект в срок	B	15.12	

Рис. 1. Цели работы руководителя.

На рис. 1 представлена простая формулировка рабочих заданий для руководителя, участвующего в программе управления по целям. Последняя колонка в таблице вводится, чтобы сравнить реальные результаты работы руководителя с запланированными.

Принятие решений

Решение – это выбор альтернативы.

На предприятии руководитель принимает решения не для себя. Чем выше уровень, тем значимее влияние принятого решения на жизнь людей. Достаточно вспомнить аварию на Чернобыльской АЭС. Важные государственные решения иногда изменяют ход истории, как, например, решение о денонсации договора о создании Советского Союза.

Существуют следующие подходы к принятию решений. Интуитивные решения возникают на основе ощущения того, что выбор правилен. Чем выше уровень управления, тем важнее роль интуитивных решений. Решения, основанные на суждениях – это выбор, обусловленный знаниями или опытом. Наконец, решение проблем может быть рациональным, почти predetermined – разумеется, при условии, что для него достаточно исходных данных и ситуация хорошо определена и запрограммирована соответствующими инструкциями.

Решать легко, найти хорошее решение трудно. Кроме прочего, на выбор руководителя сильно влияют его индивидуальные ценности. Важно заметить, что решений, не имеющих отрицательных последствий, не бывает. Опытный руководитель понимает и принимает это как факт. Но он не позволяет недостаткам решения, возможно, значительным, парализовать его волю. Принятие решений тесно связано с ходом времени. Если принятие окончательного решения затягивается, то ситуация может измениться настолько, что факторы и условия, которые принимались в расчет, просто станут недействительными. Поэтому решение следует принимать достаточно быстро. Так, почтовая система российского Интернет-портала Aport! по техническим возможностям превосходит своих более удачливых конкурентов Rambler и Yandex, а их почтовые службы, в свою очередь, технически

превосходят mail.ru. Однако по популярности все перечисленные почтовые службы располагаются в обратном порядке в полном соответствии со временем их появления на рынке. Поэтому отнюдь не всегда самое популярное является лучшим, и наоборот.

С другой стороны, бывает так, что правильное решение опережает свое время. Например, компания 3М разработала цветной копировальный аппарат еще в 1961 г., но не смогла довести его до промышленного образца, а фирма Хегох сделала удачный аппарат лишь в 1975 г. Сегодня Хегох – признанный лидер в области копировальной техники.

Еще пример. В 1995 году белорусская компания «Надежные программы» выпустила первую в Беларуси автоматизированную банковскую систему RelBank в архитектуре клиент-сервер. Проект разрабатывался с 1993 года коллективом опытных разработчиков. Продукт был анонсирован весной 1995 года. Однако, несмотря на обилие рекламы, банки не проявили интерес к новой разработке. Заметим, что и сегодня время архитектуры клиент-сервер и Интранет-технологий в белорусских банках не наступило из-за отсутствия насущной потребности в глобальной сети банковского обслуживания клиентов, низкой квалификации персонала и дороговизны требуемого аппаратного и программного обеспечения.

Неизбежные информационные ограничения накладывают отпечаток на решение. Часто бывает, что необходимая для принятия хорошего решения информация недоступна или стоит слишком дорого. Если информации не хватает, то лучший выход – отложить принятие решения до прояснения ситуации. Считается, что чем опытнее руководитель, тем меньше информации требуется ему для принятия решения. В шутку говорят даже, что идеальный руководитель способен принимать решения, не имея вообще никакой информации.

Методы прогнозирования

Прогнозирование – это метод, в котором используется накопленный в прошлом опыт и текущие допущения с целью определения будущего.

К *неформальным методам* прогнозирования относится вербальная и письменная информация, которую собирают руководители для составления прогнозов, а также промышленный шпионаж.

Взаимное перенятие важных идей служит залогом развития индустрии программного обеспечения. Многие обвиняли Microsoft в том, что она полностью позаимствовала идею Windows у компании Apple, где концепция графического интерфейса операционной системы (с пиктограммами и прочими характерными моментами технологии) была воплощена 8 годами раньше еще в системе MacOS. Однако забывают, что этот интерфейс был придуман еще раньше в лаборатории XEROX Palo Alto Research Center, где он так и остался нереализованным проектом, а также о существовании X-Window System, являющейся редко используемой графической оболочкой UNIX. Не приходится говорить о новизне Linux (после SCO UNIX), FoxPro и Clipper (после dBase) и тысяч других продуктов.

Когда есть основания считать, что деятельность в прошлом имела определенную тенденцию, которую можно продлить и в будущем, то применяют *количественные методы*.

Анализ временных рядов (проецирование тренда) предполагает, что случившееся в прошлом дает хорошее приближение в оценке будущего. Данный метод часто используется для оценки спроса на товары и услуги, оценки потребности в запасах, прогнозирования структуры сбыта, характеризующегося сезонными колебаниями (рис.2).

Незаменимым инструментом для работы на фондовом рынке является так называемый технический анализ, который всецело основан на произвольном допущении о том, что история курсов изменения всевозможных показателей является базой для прогнозирования состояния рынка в настоящем и будущем (price discounts everything). На основе этого предположения развит солидный математический аппарат. Существуют сотни показателей (цена, объем торгов, среднее, скользящее среднее, десятки производных показателей и т.д.), на основе которых специалисты по операциям с ценными бумагами принимают решения о купле-продаже, оценке инвестиций, выборе инвестиционного портфеля и т.д. В техническом анализе фондовых рынков используется достаточно серьезный

математический аппарат – числа Фибоначчи, производные, смещенные и несмещенные оценки дисперсии, графические модели (японские подсвечники) и т.д., позволяющие оценить изменения показателей во времени и предсказать их поведение. Разумеется, все эти средства требуют сложного программного обеспечения. Некоторые средства технического анализа доступны в виде Web-служб в Интернете, например, на сайтах www.smartmoney.com или www.bigcharts.com.

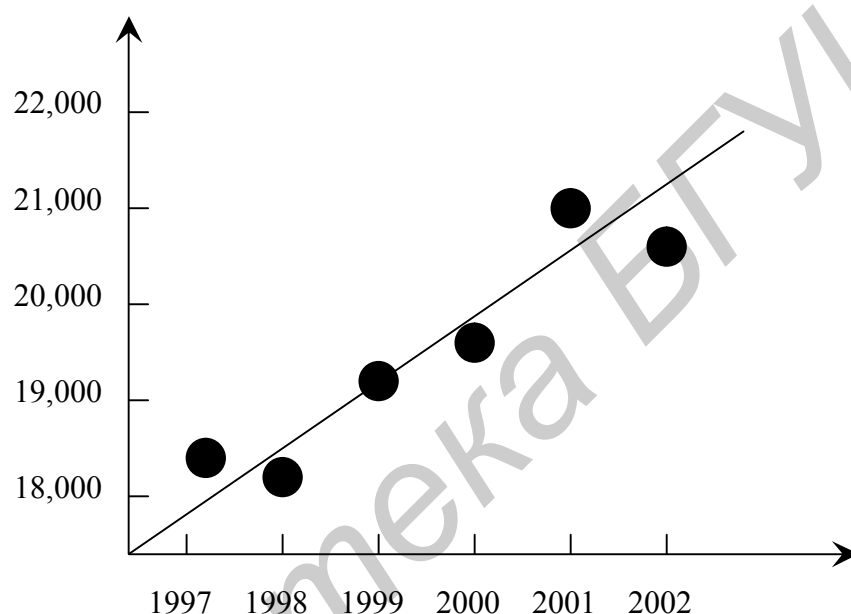


Рис.2. Анализ временных рядов

Каузальное (причинно-следственное) моделирование – наиболее сложный количественный метод прогнозирования. С его помощью можно предсказать, как, например, снижение налогов повлияет на степень легализации бизнеса в Беларуси. Каузальное моделирование – это попытка спрогнозировать то, что произойдет в подобных ситуациях, путем исследования статистической зависимости между рассматриваемым фактором и другими переменными.

Молодые специалисты зачастую увлекаются количественными методами, однако опытные управляющие относятся к ним (и не без основания) с изрядной долей иронии. Обладая большим жизненным опытом,

они хорошо понимают изменчивость внешнего мира и субъективность любых оценок и решений. Больше внимание они уделяют *качественным методам*:

- мнение жюри;
- совокупное мнение сбытовиков;
- модель ожидания потребителя;
- метод экспертных оценок.

Метод экспертных оценок представляет собой процедуру, позволяющую группе экспертов приходить к согласию. Эксперты, практикующие в самых разных, но взаимосвязанных областях деятельности, заполняют подробный вопросник по поводу рассматриваемой проблемы. Они записывают также свои мнения о ней. Каждый эксперт затем получает свод ответов других экспертов, и его просят заново рассмотреть свой прогноз, и если он не совпадает с прогнозами других, просят объяснить, почему это так. Процедура повторяется обычно три или четыре раза, пока эксперты не приходят к единому мнению. Описанный метод получения экспертных оценок принято называть дельфийским, или методом дельфийского оракула.

Организация взаимодействия и полномочий

Говоря о построении организации, обычно имеют в виду деление организации на подразделения соответственно целям и стратегиям. За этим часто забывается не менее важный аспект построения организации – *взаимоотношения полномочий*, которые связывают высшее руководство с низшими уровнями работающих.

Делегирование – это передача задач и полномочий лицу, принимающему на себя ответственность за их выполнение. Делегирование превращает работника в управленца, в противном случае руководитель должен был бы выполнить всю работу сам.

Ответственность – это обязательство выполнять задачи и отвечать за их удовлетворительное разрешение. Важно отметить, что ответственность не может быть делегирована. Так, на рабочем столе у президента США Гарри Трумена стояла табличка: «Больше ответственность сваливать не на кого».

В крупных организациях руководители высшего звена редко общаются с подчиненными на нижних уровнях, которые выполняют большинство конкретных задач. Тем не менее, они несут всю ответственность за дела фирмы и своих подчиненных. Большой объем ответственности – вот основная причина высоких окладов управленцев.

Полномочия представляют собой ограниченное право использовать ресурсы организации и направлять усилия ее сотрудников на выполнение определенных задач. Полномочия делегируются должности, а не конкретному работнику (старая военная пословица говорит: «честь отдается мундиру, а не человеку»). Когда человек меняет работу, он теряет полномочия старой должности и получает полномочия новой.

Часто путают друг с другом полномочия и власть. Полномочия – это ограниченное, присущее данной должности право использовать ресурсы организации. Руководитель имеет полномочия, только когда подчиненные соглашаются с его требованиями. Власть представляет собой реальную способность действовать или влиять на ситуацию. Полномочия, другими словами, определяют, *что* человек, занимающий какую-то должность, *имеет право* делать. Власть же определяет, что он *действительно может* делать.

Чтобы распределение полномочий и ответственности было четким и эффективным, используйте два основных принципа:

Принцип единоначалия заключается в том, что работник должен получать полномочия только от одного начальника и отвечать только перед этим человеком. В этом случае отношения между подчиненным и начальником становятся ясными и простыми. При такой схеме важно соблюдать пределы полномочий по известному принципу «Вассал моего вассала – не мой вассал», т.е. не пытаться управлять через голову промежуточного руководителя. Может показаться, что при большой длине цепочки команд соблюдение этого принципа может усложнить коммуникации. Однако неисчислимое количество руководителей тяжелым путем пришло к пониманию необходимости соблюдать этот принцип.

Норма управляемости – это количество работников, которые непосредственно подчиняются данному руководителю. Важно ограничивать норму управляемости, т.е. следить, чтобы в среднем на одного руководителя

приходилось не более чем 10-12 подчиненных. Если этого не делать, то руководство будет не в состоянии выполнять даже свои прямые обязанности. Известен факт, когда на вопрос японской делегации «Сколько людей у Вас в подчинении?» директор большого российского завода гордо ответил: «30 тысяч человек», после чего шокированные японцы прекратили начатые было переговоры.

Вот пять типичных причин нежелания руководителей делегировать полномочия по Уильяму Ньюмену.

1. «Я сделаю это лучше». Это – главный миф, ибо при перегруженности руководителя большая часть работы вообще никогда не будет выполнена.
2. Отсутствие способности руководить. Есть люди, которые по природе не способны к манипулированию другими, в личные ценности которых входит понятие «быть хорошим для всех».
3. Отсутствие доверия к подчиненным. В этом случае лучше вообще не браться за руководство.
4. Боязнь риска. Людям, не склонным к малейшему риску, также лучше не заниматься руководящей работой. Доверие, тем более поручение работы другим людям – это всегда риск.
5. Отсутствие выборочного контроля для предупреждения руководства о возможной опасности. Это может быть объективной причиной в случае, если контроль невозможен, а подчиненный ненадежен.

А вот шесть типичных причин избегания ответственности подчиненным по У.Ньюмену:

1. Удобнее спросить, чем решить проблему самому.
2. Боязнь критики.
3. Недостаток информации и ресурсов, необходимых для выполнения задания.
4. Перегруженность, действительная или мнимая.
5. Неуверенность в себе.

6. Отсутствие стимулов брать дополнительную ответственность.

Мотивация

Мотивация – это процесс побуждения себя и других к деятельности для достижения личных целей или целей организации.

Обычно для мотивации используется классический метод «кнута и пряника». Однако этот метод хорошо работает лишь в трудных экономических и социальных условиях. В начале нашего века Тейлор, основоположник научного управления, и его современники определили понятие «достаточной дневной выработки» и предложили оплачивать труд тех, кто производил больше продукции, соответственно их вкладу. Увеличение производительности труда, в сочетании с применением специализации и стандартизации, было впечатляющим. Однако, постепенно уровень жизни в промышленно развитых странах начал повышаться. И чем больше жизнь улучшалась, тем лучше управляющие начинали понимать, что простое денежное вознаграждение не всегда заставляет человека трудиться лучше. Вот почему современные теории мотивации основаны на результатах психологических исследований.

Мотивация осуществляется через потребности, которые у каждого человека различны. То, что важно одним работникам, может быть не важно другим.

Вознаграждение – это то, что человек считает ценным для себя. Внутреннее вознаграждение дает сама работа – это чувство достижения результата, содержательности и значимости выполненной работы, самоуважения. Это может быть также дружба и общение, возникающие в процессе работы. Внешнее дается организацией. Примеры внешних вознаграждений – деньги, продвижение по службе, символы служебного статуса и престижа (личный кабинет), похвала и признание.

Существуют десятки любопытнейших теорий мотивации, каждая из которых имеет свои достоинства и недостатки. Рассмотрим четыре наиболее известные из них.

Теории мотивации

Иерархия потребностей по Абрахаму Маслоу. Маслоу предположил, что все множество различных потребностей можно разделить на пять основных категорий:

Первичные:

1. Физиологические потребности. Являются необходимыми для выживания; включают потребности в еде, воде, убежище, отдыхе, а также сексуальные потребности.
2. Потребность в безопасности и уверенности в себе. Включают потребность в защите от внешних угроз и уверенность в том, что физиологические потребности будут удовлетворены в будущем.

Вторичные:

1. Социальные потребности включают в себя чувство принадлежности, чувство социального взаимодействия, привязанности и поддержки
2. Потребности в уважении включают потребности в самоуважении, личных достижениях, уважении и признании со стороны окружающих.
3. Потребности самовыражения – это потребности в реализации своих возможностей и росте как личности.

Все эти потребности можно расположить в виде строгой иерархической структуры (рис.3).

Маслоу считает, что каждая последующая потребность возникает только после удовлетворения предыдущей. В каждый конкретный момент времени человек будет стремиться к удовлетворению той потребности, которая для него наиболее важна. Например, человек, испытывающий голод, сначала будет искать пищу и только после этого попытаться построить убежище. Живя в удобстве и безопасности, человек захочет общаться с себе равными, а затем начнет стремиться к уважению с их стороны. Только после того, как он почувствует внутреннюю удовлетворенность и уважение окружающих, его важнейшей потребностью станет самовыражение.

Поскольку с развитием человека расширяются и его возможности, потребность в самовыражении никогда не может быть удовлетворена. Поэтому и процесс мотивации поведения через потребности бесконечен.

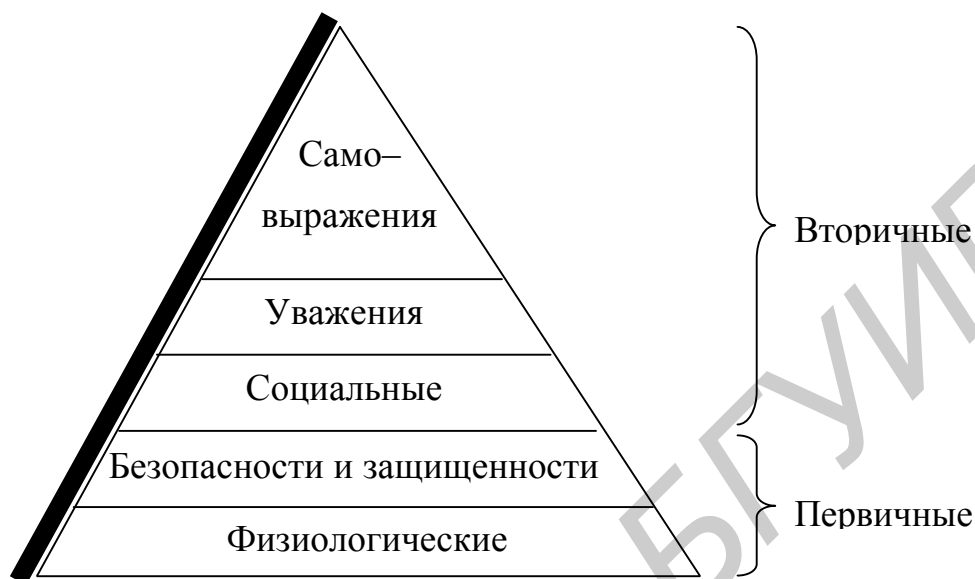


Рис.3. Иерархия потребностей по Маслоу

Критики Маслоу говорят, что на самом деле иерархия условна и зависит от культуры. Менее строго иерархия потребностей выражена ближе к Востоку, в чем мы смогли лишний раз удостовериться 11 сентября 2001 г.

Как же использовать теорию Маслоу для мотивации труда работников? Для этого служат методы удовлетворения потребностей высших уровней.

Социальные потребности:

1. Давайте сотрудникам такую работу, чтобы они могли общаться.
2. Создайте на работе дух единой команды.
3. Проводите с подчиненными совещания.
4. Уважительно относитесь к неформальным группам.
5. Создайте условия для взаимодействия работников вне работы.

Потребности в уважении:

1. Предлагайте подчиненным содержательную работу.

2. Поощряйте и высоко оценивайте достигнутые подчиненными результаты.
3. Привлекайте подчиненных к выработке целей и решений.
4. Делегируйте подчиненным права и полномочия.
5. Продвигайте их по служебной лестнице.

Потребности в самовыражении:

1. Создайте возможность для обучения и развития работников.
2. Давайте подчиненным сложную и важную работу, требующую от них полной отдачи.
3. Поощряйте и развивайте творческие способности подчиненных.

Теория потребностей Мак-Клелланда. В этой теории считается, что людям присущи три потребности: власти, успеха и причастности (принадлежности к группе). Для руководителей «от бога» характерна потребность во власти (вовсе необязательно связанная с тиранией), чтобы иметь возможность воздействовать на окружающий мир. Такие люди часто проявляют себя как откровенные и энергичные личности, не боящиеся конфронтаций и стремящиеся отстаивать свои позиции.

Потребность успеха по Мак-Клелланду находится где-то посередине между потребностью в уважении и потребностью в самовыражении в системе Маслоу. Люди с высокоразвитой потребностью успеха рискуют умеренно, любят ситуации, в которых они могут принять на себя ответственность и хотят получать конкретные вознаграждения за достигнутые ими результаты. Чтобы мотивировать людей такого склада, ставьте перед ними не слишком рискованные задачи, делегируйте достаточные полномочия для проявления инициативы и регулярно поощряйте их за достигнутые результаты.

Люди, для которых характерна потребность в причастности, заинтересованы в компании знакомых, налаживании дружеских отношений, оказании помощи другим. Их привлекает работа, которая в первую очередь дает им возможность общаться. Руководителю не надо ограничивать их контакты, а лучше дать свободу, и пусть работают.

Двухфакторная теория Герцберга. Теория Герцберга уточняет модель поведения людей. Согласно этой теории, нужно различать гигиенические факторы (рабочая среда) и мотивацию (характер и сущность работы). Гигиенические факторы сами по себе не приносят удовлетворения работой. Работники начинают обращать внимание на них только тогда, когда они неадекватны или несправедливы. Удовлетворение приносят факторы мотивации.

Теория ожиданий Виктора Врума. Наличие потребности не является единственным условием мотивации человека на достижение цели. Человек должен также надеяться на то, что выбранный им тип поведения действительно приведет к удовлетворению или приобретению желаемого. Особенно важными считаются ожидания зависимости между 1) затратами труда и результатами, 2) ожидания в отношении результатов и вознаграждений, 3) валентность или ценность вознаграждения.

Так, например, студент после очередной «отгремевшей» сессии ожидает, что если он будет систематически посещать лекции и лабораторные занятия, то подойдет к следующей сессии более подготовленным. Конечно, он может и не рассчитывать, что его усилия приведут к желаемым результатам. В этом случае, согласно теории ожидания, мотивация будет слабой.

Ожидания в отношении результатов–вознаграждений есть ожидания поощрения в ответ на достигнутый уровень результатов. Мужественно посещающий лекции студент может ожидать, что полученные им знания позволят ему получить «автоматы» и легко сдать остальные экзамены.

Не менее важен и третий фактор – ценность вознаграждения. Поскольку у различных людей потребности отличаются, то вознаграждения они оценивают по-разному. Вряд ли наш студент будет доволен, если вместо долгожданных «автоматов» его в знак поощрения отправят на межвузовскую профсоюзную конференцию.

Таким образом, получается формула мотивации:

Мотивация = З–Р * Р–В * Валентность.

Вывод из всех известных теорий сколь важен, столь и банален: *к удовлетворению ведет результативный труд.*

Мотивация и деньги

На этот счет у специалистов не сложилось единого мнения. Герцберг считает, что большинство людей относят зарплату к гигиеническим факторам, а сторонники теории научного управления считают, что деньги мотивируют.

Э.Лоулер предложил следующую систему заработной платы: «Зарплата любого работника может быть разбита на три части. Одна часть выплачивается за выполнение должностных обязанностей, и все, кто занимается в организации такими же обязанностями, получают за это одинаковое вознаграждение. Вторая часть зарплаты определяется стажем. Все сотрудники фирмы получают эту часть зарплаты, но ее величина ежегодно возрастает. Третья часть зарплаты варьируется для каждого работника, и ее величина определяется достигнутыми результатами. Плохой работник скоро увидит, что эта часть его зарплаты минимальна, а хороший поймет, что у него она, по крайней мере, столь же велика, как и первые две части, вместе взятые. Эта третья часть зарплаты не будет автоматически возрастать. Из года в год она может меняться в зависимости от результатов, достигнутых работником в прошлом году. Смысл этой системы состоит в увязывании зарплаты работника с результатами его работы».

В то же время исследования показали, что в большинстве организаций вознаграждение по результатам декларируется только на словах, а на самом деле зарплата зависит от стажа работы и выполнения распорядка рабочего дня. Вопросы справедливой оплаты труда – всегда одни из самых острых не только для работника, но и для руководителя, от которого зависит уровень оплаты работника.

Взгляды руководителя и работника на уровень оплаты труда неизбежно различаются. Руководитель может считать, что работник недорабатывает, а работник – что перерабатывает, и т.д. Бывают и субъективные факторы, не зависящие от трудового вклада – например, повышение оклада перед

пенсией. Также обычно при прочих равных условиях у семейных работников (особенно мужчин) оплата труда традиционно выше, т.к. работодатель вынужден считаться не только с традициями нашего патриархального общества, но и с объективными и субъективными потребностями работников.

В нашем обществе до сих пор существует идущая от советских времен традиция публичного обсуждения зарплаты между работниками. Поэтому, назначая зарплату, руководитель в наших условиях вынужден считаться не только с качествами работника и пользой, которую он приносит данному предприятию, но и с возможным мнением всех членов группы, в которой трудится работник. Согласно нашим законам, зарплата сотрудников не может быть отнесена к коммерческой тайне предприятия. Если частные предприятия обходят это нелепое положение, как могут, то на государственных предприятиях тайны не существует вовсе. Зарплаты работников группы сравниваются и публично обсуждаются, как оценки в школе или вузе.

Отсутствие тайны приводит к следующему. В один прекрасный день работник узнает, что его коллега, выполняющий вроде бы точно такую же работу, зарабатывает на 20% больше него, или что коллеге, но не ему, дали небольшую, но, с его точки зрения, ничем не обоснованную премию. Работник будет глубоко оскорблен и уязвлен – недооценили, пренебрегли и дали ясно это понять! Этакая публичная пощечина, плевок в лицо – ведь знают-то все. Его мотивация в тот же день на долгие месяцы, если не навсегда, понизится («руки опускаются»), он обязательно станет работать хуже, чем может. А уж про авторитет руководства можно забыть. Причем, не имеет никакого значения, действительно ли коллега объективно лучший или нет – важно лишь субъективное мнение об этом самого работника.

Принципы трудовой этики

1. Увязывайте вознаграждение с вкладом в работу предприятия в целом.
2. Выражайте публичное и осязаемое признание.

3. Каждый работник должен получить свою долю от увеличения производительности организации в целом.
4. Поощряйте участие с руководителями в разработке целей и показателей оценки труда.
5. Не допускайте возникновения и развития ситуации, при которой цели фирмы ущемляют интересы сотрудников (безопасность труда, сверхурочная работа).
6. Не пытайтесь повысить стандарты качества, пока их нельзя обеспечить.
7. Не создавайте значительного разрыва между заявлениями руководства и фактической системой вознаграждения.
8. Не пытайтесь представить выполнение целей фирмы как способ повышения удовлетворенности работой и степени ее значимости.
9. Не поддерживайте привилегии руководства, увеличивающие разрыв с подчиненными, действительно выполняющими работу.

Контроль

Для многих людей контроль означает прежде всего ограничение свободы, принуждение и тому подобные вещи. Однако сводить контроль просто к ограничениям, заставляющим вести себя дисциплинированно, — значит упускать из виду его основную задачу:

Контроль — это процесс, обеспечивающий достижение целей организации.

Контроль очень важен для успешности организации. Без него начинается хаос, и объединить деятельность многих людей становится невозможно. Контроль позволяет обнаружить и разрешить возникающие проблемы раньше, чем они станут слишком серьезными.

Для осуществления контроля необходимо выполнить три действия:

1. Выработать критерии. Критерии — это конкретные цели, достижение которых поддается измерению. Каждый критерий имеет две важные

особенности: временные рамки и критерий степени выполнения работы. Пример критерия: получить прибыль в размере 1 млн. долл. в 2005 году.

2. Сравнить достигнутые результаты с критериями. Необходимо установить допустимые отклонения, — если подходить чрезмерно строго, то контроль неэффективен. Система контроля будет действовать более эффективно, если обязательно доводить до сведения работников организации как установленные критерии, так и достигнутые результаты.
3. Действовать. По итогам контроля возможны следующие действия: ничего не предпринимать (если все идет хорошо), устранить отклонения (тщательно обдумав решение и выявив истинную причину отклонений от критериев) либо пересмотреть сами критерии. Если все менеджеры по продажам превышают свои планы продаж на 50%, то, видимо, план занижен.

Контроль — это важный аспект управления, который тесно связан с поведением людей. Избежать негативного воздействия контроля помогут следующие правила:

- устанавливайте осмысленные критерии, воспринимаемые сотрудниками;
- устанавливайте двухстороннее общение;
- избегайте чрезмерного контроля;
- устанавливайте жесткие, но достижимые критерии;
- вознаграждайте за их достижение.

Группы

Хоторнские эксперименты

Хоторнские эксперименты были проведены в 1924 г. на одном из заводов Western Electric. Эксперимент проводился с группами сборщиц реле. Было проведено четыре эксперимента над двумя группами — экспериментальной и контрольной.

К удивлению исследователей, когда они увеличили освещение для экспериментальной группы, производительность *обеих* групп возросла. То же самое произошло и тогда, когда освещение было уменьшено. После этого сокращали рабочую неделю, увеличивали перерывы, вводили бесплатные обеды – производительность росла. Когда же исследователи вернули первоначальные условия, производительность и тут осталась на прежнем уровне. Интересно, что мастера во время эксперимента контролировали работниц меньше обычного, что и явилось истинной причиной повышения производительности труда.

Выводы.

1. Хоторнский эффект – это условия, в которых новизна, интерес к эксперименту или повышенное внимание к данному вопросу приводят к искаженному, чаще слишком благоприятному, результату.
2. При планировании эксперимента необходимо всячески избегать хоторнского эффекта, чтобы люди вообще не знали, что подверглись эксперименту, или думали, что исследуются какие-то другие факторы. Вот пример правильно проведенного эксперимента без хоторнского эффекта: в США исследовали причины ожирения. В ходе экспериментов испытуемым предлагали дегустировать различные сорта печенья, причем разрешалось есть сколько угодно печенья. Психологи-экспериментаторы преподнесли тест как дегустацию, на которой предлагалось определить самые вкусные сорта. Истинная цель состояла в оценке поведения худых и толстых людей. Худые люди быстро насыщались, а у толстяков насыщения не наступало, они были готовы поглощать неограниченное количество печенья.
3. Условия эксперимента – замечательный стимул продуктивной работы. Человек работает лучше, зная, что участвует в эксперименте, поэтому нужно подавать работникам всякую работу как эксперимент, как вызов их способностям. Это особенно важно

при проектной системе производства, характерной для программистов.

Неформальные группы

Неформальная группа – это спонтанно образовавшаяся группа людей, вступающих в регулярное взаимодействие для достижения определенной цели. У неформальной группы есть свои цели, иерархия, задачи, нормы, поощрения и санкции. Главное отличие от формальной организации заключается в отсутствии плана.

Причины вступления в неформальную группу:

1. Принадлежность. Потребность в принадлежности – это одна из самых сильных наших эмоциональных потребностей.
2. Помощь. В идеальном варианте подчиненные должны без всякого стеснения обращаться за помощью к своему начальству. На практике они гораздо чаще прибегают к помощи своих коллег. Например, новый программист скорее попросит своего соседа-программиста с другого проекта объяснить ему, как использовать новое средство разработки. Это приводит к тому, что новые программисты тоже стремятся участвовать в уже сформировавшейся группе, где есть опытные программисты. Так потребность в помощи приводит к возникновению неформальной организации.
3. Защита. Пословица «в единстве – сила» остается актуальной и сегодня. Рабочие на заводе, например, могут использовать пропуска друг друга, чтобы скрыть опоздания на работу, или объединить усилия, чтобы опротестовать вредные условия работы.
4. Общение. Люди хотят знать, что происходит вокруг них, особенно, если это касается их работы. Одной из важных причин принадлежности к группе является доступ к неформальному каналу поступления информации – слухам (см. раздел «Коммуникации»).
5. Симпатия. Наконец, люди присоединяются к неформальным группам просто для того, чтобы быть ближе к тем, кому они симпатизируют.

Людей обычно тянет к тем, кто, по их мнению, может удовлетворить их потребности.

Для неформальных групп характерны:

1. Социальный контроль над работниками. Чтобы быть принятым группой человек должен соблюдать групповые нормы. Тех, кто нарушает эти нормы, может ждать отчуждение.
2. Сопротивление переменам. Причина сопротивления кроется в том, что перемены могут разрушить неформальную группу. Реорганизация, расширение производства или появление большой группы новых сотрудников может привести к распаду группы.
3. Наличие неформальных лидеров. Отличие неформального лидера состоит в том, что его опора – признание группой, а не делегированные ему официальные полномочия. В своих действиях он делает ставку на людей и их взаимоотношения.

Существует два правила управления неформальной группой. Во-первых, необходимо признать ее существование, работать с ней и не угрожать ее существованию, не «давить». Во-вторых, нужно выслушивать их лидеров, но поощрять только тех из них, которые способствуют достижению целей организации.

Как повысить эффективность групп

Общие факторы, влияющие на эффективность группы, следующие:

1. Размер. Эффективная группа должна быть небольшой, 5 – 10 человек.
2. Состав. Лучше, если группа состоит из непохожих личностей.
3. Групповые нормы. Положительные групповые нормы повышают эффективность группы. Это может быть гордость за организацию, достижение ее целей, забота о качестве продукции, защита честности.

Таким образом, эффективная группа – это та, чей размер соответствует ее задачам, в составе которой находятся люди с непохожими чертами

характера, чьи нормы способствуют достижению целей организации и созданию духа коллективизма, где здоровый уровень конфликтности и где имеющие высокий статус члены группы не доминируют.

Проведение совещаний

Многие руководители считают, что проведение совещаний – одно из самых трудных дел в управлении группой.

Как организовать эффективное совещание?

1. Составьте конкретную повестку для собрания и лично пересмотрите ее перед началом собрания.
2. Обеспечьте свободный обмен мнениями между членами группы.
3. Стремитесь полностью использовать индивидуальные способности членов группы для решения общих задач.
4. Создайте атмосферу доверия.
5. Управляйте конфликтом, рассматривая его как положительный фактор, способный выявить новые подходы к решению проблемы.
6. Заканчивая совещание, кратко суммируйте обсуждение и определите, что делать дальше.

Совещание группы, где хотят извлечь пользу из различия мнений, может быть успешным, только если на нем не проявится групповое единомыслие, в результате которого получаются посредственные решения, которые никого не задевают.

Как бороться с единомыслием на совещаниях:

1. Внушите членам группы, что они могут свободно сообщить любую информацию.
2. Нужен «адвокат дьявола» – человек, защищающий явно неправо дело.
3. Критика должна восприниматься так же спокойно, как и конструктивные комментарии.
4. Отделите усилия по генерированию идей от их оценки.

5. Сначала необходимо выслушивать идеи подчиненных, а только потом говорить самому.

Лидерство и власть

Несмотря на то, что способность быть лидером – это важная черта для менеджера любого уровня, сильные лидеры не всегда становятся хорошими руководителями. Между управлением и лидерством есть большое различие.

Управление – это процесс (умственный и физический), приводящий к выполнению подчиненными предписанной им задачи. Лидерство является процессом, с помощью которого одно лицо оказывает влияние на членов группы.

Лидерство может и мешать формальному управлению. Например, влиятельный неформальный лидер может сделать так, что трудовой коллектив начнет производить товары или услуги низкого качества. С другой стороны, когда качества лидера и руководителя организации объединяет в себе один человек – это залог успеха для организации. Классические примеры лидеров: руководители Intel Эндрю Гроув и Гордон Мур, руководитель Bloomberg Майк Блумберг, руководитель Sun Скотт Макнили.

Влияние среды на личность и поведение велико. Известно, что группы могут оказывать большое влияние на поведение конкретных людей (вспомните хотя бы рок-концерты), навязывая групповые ценности, которые могут совпадать (например, коллективизм), а могут и не совпадать с целями организации. И в то же время базовые свойства личности важнее. Так, исследования Стенфордского университета показали, что те его выпускники программы MBA добились большего успеха и больше зарабатывают, кто имеет значительную энергию, уверенность в себе, социальную смелость и малые опасения при принятии решений.

Власть – это возможность влиять на поведение других. В управлении власть необходима, одних полномочий бывает недостаточно. Во-первых, руководители часто зависят от людей, которые не подчинены им напрямую. Во-вторых, мало кто в современных организациях будет подчиняться

непрерывному потоку приказов начальника только потому, что он — начальник.

Подчиненные тоже имеют власть над своим начальником, т.к. они предоставляют ему необходимую информацию, оказывают влияние на своих коллег, и, наконец, выполняют сами задания. Руководитель должен осознать, что раз у подчиненных тоже есть власть, неограниченное использование им своей власти может вызвать у них ответную реакцию. Поэтому опытный руководитель старается поддерживать разумный баланс власти: достаточный для достижения его целей, но не вызывающий у подчиненных чувство протеста.

Харизма — это власть, построенная не на логике или традиции, а на силе личных качеств и способностей лидера. В представлении исполнителя его отношения с лидером строятся как равные, даже если они никогда не встречались. Исполнителю кажется, что у него много общего с лидером. На уровне подсознания он ждет, что подчинение сделает его похожим на лидера или, по крайней мере, вызовет к нему уважение.

Характерные признаки харизматических личностей.

1. *Обмен энергией.* Создается впечатление, что эти люди излучают энергию и заряжают ею окружающих.
2. *Внушительная внешность.* Харизматический лидер не обязательно красив, но привлекателен, обладает хорошей осанкой и прекрасно держится.
3. *Независимость характера.* В своем стремлении к благополучию и уважению эти люди не полагаются на других.
4. *Хорошие риторические способности.* У них есть умение говорить и способность общаться.
5. *Восприятие восхищения.* Они чувствуют себя комфортно, когда другие выражают им восхищение, не впадая в надменность или себялюбие.

6. *Достойная и уверенная манера держаться.* Они выглядят собранными и владеющими ситуацией.

Люди чаще испытывают влияние тех, кто обладает восхищающими их чертами характера и кто является их идеалом, похожими на который они хотели бы быть. Всем известные примеры – Черчилль, Гитлер, Сталин, Рузвельт, Кеннеди, Де Голль, Меир. Наиболее известные харизматические личности мира информационных технологий – Скотт Макнили (Sun Microsystems), Ларри Эллисон (Oracle Corp.) и Стив Джобс (Apple Computer Products). Все они – великолепные ораторы, их охотно приглашают на различные публичные мероприятия (причем иногда даже конкуренты) и затем наперебой комментируют их речи, оставляющие неизгладимое впечатление у слушателей.

Личности высшего управленческого звена компаний накладывают отпечаток на деятельность всей компании. Все мы знаем, насколько изменяется ситуация при смене, например, президента страны. Известно, каким образом повлияла на IBM смена ее президента, когда Джона Янга заменил Луи Герстнер, до этого управлявший компанией по производству кондитерских изделий. После многих лет убытков дела IBM быстро поправились, и IBM уже много лет подряд с тех пор не знает убытков даже в эпоху кризисов. Таким образом, на уровне первого лица в крупной компании уже не столь важно, чем именно руководитель руководит и разбирается ли он в предмете: важно быть хорошим руководителем. Руководитель крупной организации фактически вообще не управляет, а является своеобразным флагом, знаменем, носителем ценностей организации.

Показательна также история руководства компании Apple. В середине 80-х годов ее основателя визионера Стива Джобса за издевательства над сотрудниками, привычку командовать через головы нижестоящих руководителей, минуя несколько уровней иерархии, и ряд других ошибок сместили акционеры, ибо никто больше не мог терпеть этого несносного, высокомерного человека. Вместо него назначили известного менеджера Джона Скалли, не обладавшего указанными недостатками. В результате за несколько лет убытки достигли фантастических цифр, а доля рынка упала с

десятков до нескольких процентов. Компания уверенно шла к банкротству. Тогда акционеры вынуждены были вернуть на место Стива Джобса, что и было проделано при помощи хитрой операции (покупки 100% акций компании, принадлежащей Стиву Джобсу и возглавляемой им после ухода из Apple), и спустя непродолжительное время дела снова пошли на лад, хотя, конечно, былого могущества Apple уже никогда не вернуть.

«Лучший» стиль руководства

Стиль руководства – это привычная манера поведения руководителя по отношению к подчиненным.

Руководство бывает автократичное (авторитарное) и демократичное. Авторитарный руководитель считает, что люди не любят трудиться, у них нет честолюбия, и они стараются уйти от ответственности. Чтобы заставить людей трудиться, необходимо использовать принуждение и угрозу наказания. Как следствие, автократ старается централизовать полномочия.

Напротив, демократичное руководство считает, что труд – естественный процесс. Люди при благоприятных условиях примут на себя ответственность и будут даже стремиться к ней. Если люди верят в цели организации, они будут использовать самоуправление и самоконтроль. Необходимо приобщение, творческое решение проблем. Демократичное руководство обращается к потребностям более высокого уровня.

Иначе можно сказать, что авторитарный руководитель – это руководитель, сосредоточенный на работе, а демократичный – ориентированный на человека.

В организациях с преобладанием специалистов с высшим образованием и сложными видами деятельности типа программистских фирм более характерен демократичный стиль управления, а, скажем, на заводах – автократичный.

Некоторые исследования показали, что автократичное руководство обеспечивает более высокую производительность, но более низкую степень удовлетворенности. Однако это, во-первых, не подтверждено другими исследованиями, а во-вторых, для hi-tech это определенно не так.

Вот типичная иллюстрация достоинств и недостатков обоих подходов. Генри Форд (Ford Motors) изобрел в 1913 г. конвейерную сборку и внедрил ее в производстве автомобиля «Ти». В результате цена автомобиля упала с 2100 до 290 долл., а США стали первой страной массового автомобилизма. Достигнув высот успеха, Форд уверовал в собственную непогрешимость. Авторитарность Форда, некогда помогшая ему достичь успеха, впоследствии сослужила плохую службу его компании, когда он с ходу отвергал все, что не согласовалось с его личными воззрениями, а всех выразивших несогласие с его мнением немедленно увольнял. «Любой клиент может выбрать автомобиль любого цвета, пока он остается черным», – заявил Форд в ответ на призывы разнообразить продукцию компании и жестоко поплатился за это.

Напротив, гибкость руководителя General Motors Альфреда Слоуна, исповедовавшего демократические методы управления, позволила привлечь к управлению сотрудников всех уровней, собрать лучшие предложения по совершенствованию производства, внедрить их и, в конце концов, в течение десятилетия победить Ford Motors, выпуская разнообразные модели автомобилей разных цветов для разных потребителей.

На самом деле в одной и той же организации, но в разных ситуациях могут лучше действовать разные подходы. «Лучшего» стиля руководства не существует. Эффективно работающий менеджер – это тот, кто может менять свой стиль в зависимости от ситуации.

Однако действовать сообразно ситуации не всегда удастся. Стиль работы руководителя зависит от системы его жизненных ценностей и мало поддается изменению. Понимая это, нужно просто правильно назначать руководителей, соразмеряя поручаемые задачи и функции их личным качествам.

Управление и люди

Все внутренние переменные организации взаимосвязаны. Организация – это социотехническая система. Занимаясь только технологией, нельзя добиться успеха.

Если руководство или отдельные менеджеры не признают, что каждый работник представляет собой личность с ее неповторимыми переживаниями и запросами, способность организации достичь намеченных целей будет поставлена под угрозу. Руководство достигает своих целей при помощи людей. Люди – центральный фактор любой модели управления. Работа должна быть интересной и содержательной! Если в работе есть монотонность, то содержание труда необходимо изменить.

Управление конфликтами

Конфликт – это отсутствие согласия между двумя и более сторонами. Каждая сторона делает все, чтобы была принята ее точка зрения, и мешает другой стороне сделать то же самое. Большинство людей думают, что конфликт – явление нежелательное и его необходимо, по возможности, избегать. Однако современная точка зрения заключается в том, что в хорошо управляемых организациях некоторые конфликты не только возможны, но даже могут быть и полезны. Часто конфликт помогает выявить разнообразие точек зрения, дает дополнительную информацию, выявляет большее число проблем. Это улучшает качество решений, принятых группой, а кроме того, дает возможность людям выразить свои мысли.

Типичный межгрупповой конфликт на заводе – конфликт между работниками цехов и заводоуправлением. Специалисты заводоуправления обычно более молоды, лучше образованы и любят оперировать техническим жаргоном, чем вызывают презрение цеховых работников: «Эти АСУПчики ничего реально делать не умеют, только бросаться умными словами и распечатывать горы никому не нужных бумаг». В то же время специалисты-компьютерщики заводоуправления презируют цеховиков за их недалекость и невежество: «Хорошо еще, что ползавода не сгорело! Эти неучи даже не знают, на каком минимальном расстоянии кабель локальной сети должен быть от силового».

Одна из наиболее распространенных причин конфликта – конфликт ролей, когда к одному человеку предъявляются противоречивые требования со стороны начальства. Например, руководитель отдела разработки может потребовать, чтобы программист все время находился на рабочем месте и

занимался тестированием программы, а в другой момент высказать недовольство тем, что программист уделяет недостаточно времени контактам с заказчиком.

Конфликты часто возникают из-за противоречий между различными подразделениями организации. Служба сбыта заинтересована в больших запасах продукции, а производство – в низких. Дневная смена системных администраторов Интернет-провайдера может конфликтовать с ночной из-за того, что ночная смена оставила ей клиента с проблемами.

Методика разрешения конфликтов:

1. Определите проблему в категориях целей, а не решений.
2. Определите решения, приемлемые для обеих сторон.
3. Сосредоточьтесь на проблеме, а не на личных качествах другой стороны.
4. Создайте положительное отношение друг к другу, симпатию.

Стресс

Стресс – обычное и часто встречающееся явление. Мы все временами его испытываем, например, как повышенную раздражительность во время сессии или бессонницу перед самым трудным экзаменом. Небольшие стрессы неизбежны и безвредны. Только чрезмерный стресс создает проблемы.

Как снизить стресс на работе:

1. Разработайте систему приоритетов в своей работе. Оцените свою работу следующим образом: «должен сделать сегодня», «сделать позже на этой неделе» и «сделать, когда будет время». Научитесь говорить «нет», когда Вы уже не можете взять на себя больше работы.
2. Наладьте доброжелательные отношения с Вашим начальником. Поймите его проблемы и помогите ему понять Ваши.

3. Не соглашайтесь с Вашим начальником, когда он начинает выставлять противоречивые требования (конфликт ролей). Не занимайте обвинительной позиции, просто объясните, какие проблемы создают для Вас противоречивые требования.
4. Сообщите своему руководителю, если почувствуете, что ожидания или стандарты неясны (неопределенность ролей).
5. Найдите каждый день время для отдыха. Закройте дверь на пять минут, расслабьтесь и выбросьте работу из головы. Не задерживайтесь подолгу на работе после того, как Вам следовало бы уже идти домой.

При изменении целей, технологии или задач часто приходится изменять и структуру организации, а это тоже стресс. Необходимо мотивировать людей, чтобы они приняли изменения. Хорошо, когда работники сами принимают участие в осуществлении перемен.

Эффективность организации

Понятие о системе управления производством

Производство на предприятии можно рассматривать как открытую систему, которая преобразует входные величины в выходные. Система управления производством состоит из трех подсистем:

1. *Перерабатывающая подсистема* выполняет работу, непосредственно связанную с выпуском продукции (не обязательно в области промышленности). Для университета основными результатами деятельности являются разработка и распространение знаний. Следовательно, работа перерабатывающей подсистемы заключается в создании этих результатов через научно-исследовательскую и педагогическую деятельность преподавателей. Требуются также капиталовложения в виде аудиторий, лабораторий, кабинетов, мебели, оборудования, а также бумаги и канцелярских товаров.

2. *Подсистема снабжения* не связана прямо с производством, но выполняет необходимые для работы перерабатывающей подсистемы функции. В университете подсистема обеспечения включает библиотеку, вычислительный центр, ремонтно-эксплуатационную службу, книжные киоски, кафетерий, типографию.

3. *Подсистема планирования и контроля.* В университете функции планирования и контроля выполняет администрация университета, в том числе и деканаты. Расписание занятий строится на основе учебных планов, наличия аудиторий и т.д. Качество преподавания контролируется силами профессорско-преподавательского состава и оценками студентов.

На Западе вместо термина «управление производством» используется термин «операционная система» – полная система производственной деятельности организации. Руководит операционной системой в западной компании вице-президент по операциям – Chief Operation Officer, сокращенно СОО. В нашей стране производством руководит заместитель директора по производству, либо первый заместитель директора, либо исполнительный директор.

На Западе принято сокращать названия руководящих должностей. Вот типичные сокращения:

- СОО (Chief Operation Officer) – вице-президент по операциям;
- CEO (Chief Executive Officer) – исполнительный директор, генеральный директор;
- CFO (Chief Financial Officer) – финансовый директор (главный бухгалтер);
- CIO (Chief Information Officer) – директор по информационным технологиям;
- CTO (Chief Technology Officer) – директор по технологиям.

Слоганом известной американской рекламы системы скидок на авиаперелеты является «Fly like a CEO, spend like a CFO» («Летай, как директор, трать, как главбух»).

Эффективность производства – это рыночная стоимость произведенной продукции, деленная на общую величину затрат. Как видите, обороты и размер компании не имеют значения при оценке ее эффективности.

Классификация систем управления производством

Классификацию можно проводить разными способами. Производство бывает дискретным (если продукция измеряется в штуках) и непрерывным (если изделия измеряются килограммами, метрами, литрами, кубометрами и т.д.). Перерабатывающая система производства может быть проектной, мелкосерийной, массовой и непрерывным процессом (см. табл. 1).

Таблица 1

Классификация операционных систем

Характер предложения	Тип перерабатывающей системы			
	Проектная	Мелкосерийная	Массовое производство	Непрерывный процесс
Продукция	Строитель Подрядчик Писатель	Типография Столярный цех	Автосборочный завод Швейное предприятие	Нефтепереработка Угольная шахта
Услуги	Врач-терапевт Маляр	Клиническая больница Авторемонтная станция	Аэропорт Мойка машин	Радиостанция Патруль

Типы производства хорошо иллюстрируют предприятия из сферы обслуживания и питания. Фирмы, обслуживающие банкеты и другие мероприятия, работают по типу реализации проектов. Основная часть ресторанов представляют собой предприятия с мелкосерийным производством. Массовое производство – это МакДональдс.

Три тенденции последних лет в производстве:

- комплексное управление качеством (международные стандарты качества серии ISO9000),
- управление взаимоотношениями с клиентами (CRM)
- управление производственными цепочками (MPR).

Жизненный цикл изделия и процесса

Срок жизни любого изделия ограничен. Каждый товар проходит в своем развитии этапы внедрения, роста, зрелости и спада (рис. 4).

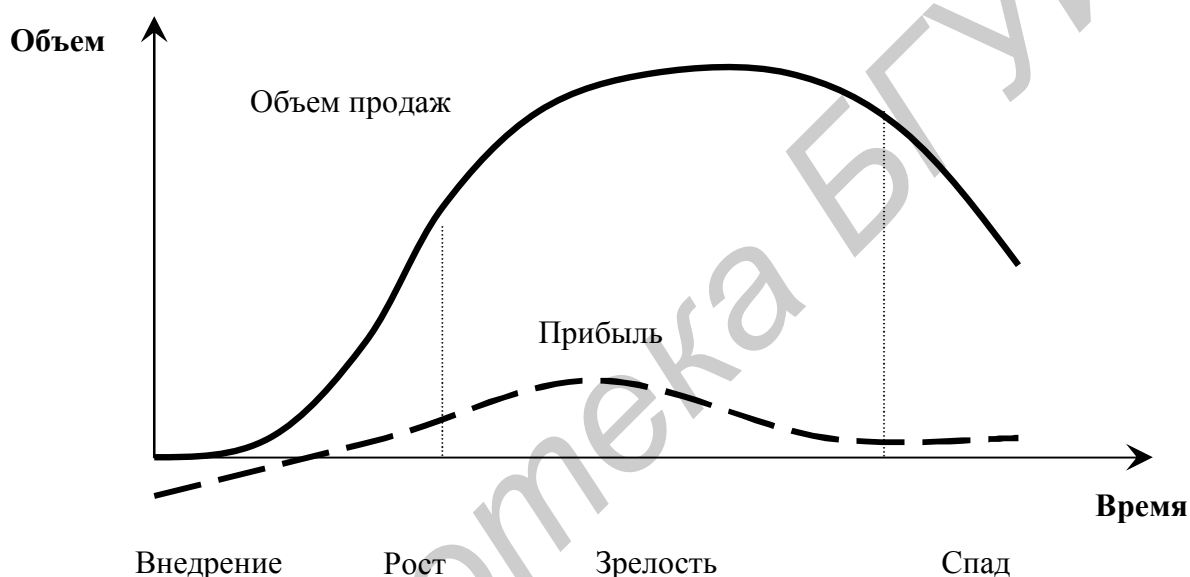


Рис. 4. Изменение объема продаж и прибыли на протяжении жизненного цикла товара.

Этап внедрения характеризуется медленным увеличением объема продаж товара в период, когда он впервые поступает на рынок и только «завоевывает» покупателей. В связи с большими затратами компании на данном этапе продажи товара не приносят прибыли. Действительно, кроме решения производственных проблем, фирме необходимо: 1) информировать потенциальных потребителей о новом, еще не известном товаре, 2) убедить их опробовать продукт и 3) обеспечить его реализацию через предприятия розничной торговли. Цены на новый товар достаточно высоки, его потребителями становятся, главным образом, любители всего нового и хорошо обеспеченные люди.

На **этапе роста** отмечается резкое увеличение объемов продаж. Потребители, «признавшие» товар с момента его появления на рынке, будут продолжать покупать его, и их примеру последуют другие. По мере увеличения спроса цены на товар остаются на прежнем уровне или немного снижаются. Прибыли на этом этапе растут быстрыми темпами. Однако со временем темпы роста неизбежно начнут снижаться. Менеджменту компании необходимо своевременно определить этот момент и переходить к новым маркетинговым стратегиям.

Открытый в 1994 г. и ориентированный на студентов портал Yahoo! является одним из самых популярных сайтов глобальной «Паутины». Yahoo! – не только мощная поисковая система, но и поставщик самого разного рода информации и услуг – от недвижимости и финансов до новостей, предложений о продажах и т.д. В 1998 г. рыночная цена акции компании достигла \$200, а капитализация составила \$9,1 млрд. Причем усиливающаяся рыночная конкуренция (порталы типа Infoseek, возвращенные финансовыми и телевизионными сетями, готовые к бою за киберпространство Microsoft, Netscape, American Online Inc, General Electric-NBS) никак не повлияла на фантастический рост компании. Стратегия Yahoo! – рост, рост и еще раз рост. Компания планирует освоить сферы, недоступные ни одному из порталов: телефоны, пейджеры, органайзеры и т.п. Помимо прочего, она стремится укрепить позиции своей торговой марки и совместно с Visa International планирует заниматься е-коммерцией.

Столь грандиозные планы помешали руководству компании вовремя увидеть надвигающийся кризис в компьютерной отрасли и правильно среагировать. Акции большинства фирм этой отрасли сильно упали, но акции Yahoo! пострадали больше других, упав за 2 года с 255 долл. в конце 1999 г. до 10–15 долл. к осени 2001 г.

Этап зрелости. По времени этап зрелости является самым протяженным. Большинство товаров на рынке находится в стадии зрелости. Объем продаж на этой стадии сохраняется на примерно постоянном уровне, компания получает высокие прибыли.

Этап спада. Наконец, неизбежен тот момент, когда объем продаж товара или марки начинает снижаться. Здесь стоит рассказать об интересном психологическом факторе: мы прощаемся с умирающими продуктами едва ли не так же, как расстаемся со старыми верными друзьями. «Шестиугольный крендель был первым изделием компании... Вы можете представить себе, что когда-нибудь его не будет?». Но менеджер обязан помнить о логике. Ведь для продвижения «стареющего» товара необходима дополнительная реклама и напряженные усилия продавцов. Кроме того, слабые товары не пользуются доверием потребителей и бросают тень на имидж компании. Поэтому в случае отсутствия очень веских причин для сохранения «жизни» слабого продукта следует принять решение о прекращении его выпуска.

Проектирование продуктов и процессов в сфере услуг

В сфере услуг занято около 80% работающих развитых стран и более половины развивающихся, и эта доля продолжает увеличиваться.

Многие принципы одинаково применимы как для производственного сектора, так и для сферы услуг, но есть и целый ряд различий:

1. Тесное взаимодействие с потребителем. Рабочий на заводе не общается с потребителями, в больнице же или в кинотеатре служащие тесно взаимодействуют с клиентами.
2. Высокая степень индивидуализации продукта.
3. Работы более трудоемки.

Из-за этих различий крупные предприятия в сфере услуг нетипичны.

Проблемы в проектировании услуг:

1. Для проектирования услуг важно месторасположение предприятия. Оно определяется месторасположением потребителей, а не сырья или какими-либо другими факторами.
2. В процессе работы потребности и пожелания потребителей важнее эффективности.

3. Календарное планирование работ невозможно, ибо диктуется потребителями.
4. Создание запасов невозможно, поэтому производственные мощности рассчитываются по пиковому спросу, а не по среднему.
5. Измерение качества и эффективности затруднены: иногда даже годы спустя невозможно оценить, например, качество выполнения конкретной диагностической процедуры.
6. Большинство работников предприятия сферы услуг должно общаться с потребителями в процессе работы, в результате чего маркетинг и производство иногда трудно отличить друг от друга.

Перечисленные проблемы делают управление в сфере услуг более трудным делом. Тем не менее, существует ряд методов повышения производительности в этой области:

1. Усиление участия потребителя в производстве услуг. Организация продажи товаров по почтовым заказам, бензозаправочные станции самообслуживания, кафетерии – вот примеры применения этого принципа.
2. Замена ручного труда техникой там, где это выгодно. Примерами могут служить торговые автоматы, считывающие устройства в кассах продовольственных магазинов, автоматические станции мойки автомашин.
3. Упрощение или устранение части процесса по предоставлению услуги. Примером может служить заблаговременная «выписка» из отеля. Клиент заранее заполняет платежные документы и указывает дату выезда, что экономит труд работников и время клиента.

Управление проектами и качеством

Управление проектами

Проект – это совокупность задач или мероприятий, связанных с достижением запланированной цели, которая обычно имеет уникальный и

неповторимый характер. Руководители проектов отвечают за три стороны реализации проекта: сроки, расходы и качество. Считается, что управление сроками работ является ключом к успеху по всем трем показателям. Там, где сроки затягиваются, весьма вероятно перерасходование средств и возникновение серьезных проблем с качеством. Поэтому во всех методах управления проектами основной акцент делается на календарном планировании работ и контроле за соблюдением графика.

Для управления проектами широко используются специальные методы и инструменты сетевого планирования – сетевой график, метод критического пути (при детерминированных сроках), метод оценки и пересмотра планов ПЕРТ (при вероятностных сроках). Из программных средств, поддерживающих перечисленные методы, наиболее известным и широко применяемым является Microsoft Project.

Метод критического пути. Согласно методу критического пути календарный график работ разрабатывается в четыре этапа. Для наглядности возьмем пример по строительству гаража.

I. *Цели и ограничения.* Цели и ограничения проекта определяются тремя характеристиками его реализации (сроки, стоимость и качество), а также присущими каждому проекту особенностями. На нашем примере целью может быть скорейшее завершение строительства гаража при следующих ограничениях:

1. Стоимость всего проекта не должна превышать 1000 долл.
2. Соблюдение технических условий проекта и всех строительных норм является обязательным.
3. Некоторые виды работ нельзя выполнять в ненастную погоду.

II. *Продолжительность работ.* Вторым этапом заключается в определении работ, входящих в проект, и расчете длительности каждой работы или операции. Ожидание того, пока бетонная плита затвердеет, считается операцией, поскольку на это требуется время, и начать выполнение других работ до затвердения плиты невозможно. Перечень работ при строительстве гаража показан в табл. 2.

Таблица 2

Продолжительность работ по строительству гаража

Обозначение	Работа	Продолжительность, дней
ВГ	Выемка грунта	1
ЗБ	Заливка и затверждение бетона	3
ВК	Взведение каркаса	2
УК	Укладка крыши	2
УО	Установка окон	1
БС	Обшивка боковых стен	2
ЭП	Монтаж электрической проводки	2
УД	Установка дверей	1
ЖВ	Монтаж желобов и водостоков	1
ОУ	Окончательная уборка строительной площадки	1

III. *Сетевой график работ.* Третий этап включает анализ очередности работ и составление сетевого графика (рис. 5). Некоторые работы должны выполняться последовательно, другие можно делать параллельно. Отношения очередности в основном определяются техническими причинами. Например, технически невозможно уложить крышу до тех пор, пока не будет возведен каркас, но вместе с тем, должны учитываться и другие факторы (например, требования техники безопасности).

IV. *Календарный сетевой график.* На четвертом этапе строится календарный сетевой график на основе таблицы продолжительности работ и полученной сетки (рис.6). Методика расчета не входит в задачи нашего пособия. Продолжительность работ по проекту составляет 15 дней.

Критический путь в проекте – это самая длительная (по срокам) последовательная цепочка операций. На нашем графике критическим путем

является непрерывная цепочка работ ВГ–ОУ, проходящая через центр графика. Длина критического пути определяет продолжительность проекта. Любые задержки на критическом пути ведут к задержкам всего проекта. Таким образом, для сокращения сроков проекта необходимо сокращать длину критического пути.

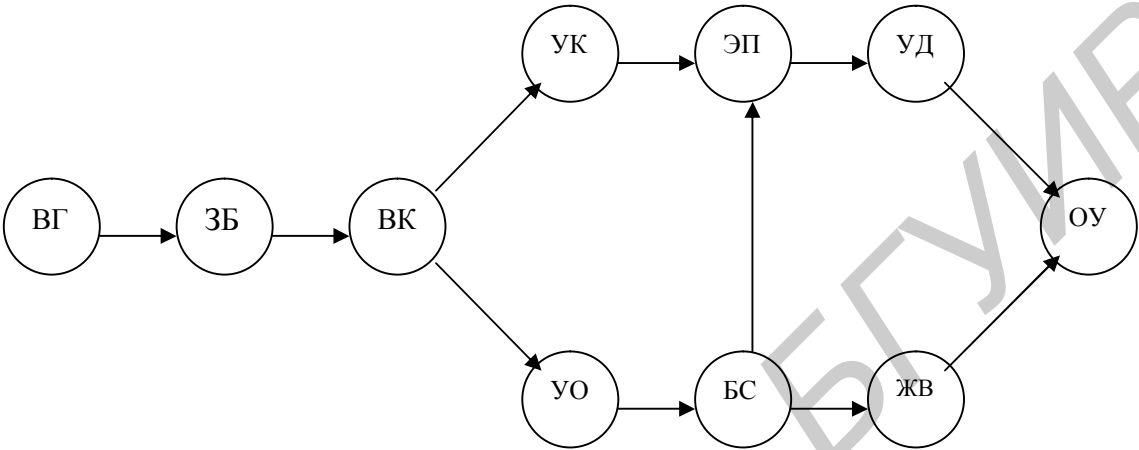


Рис. 5. Сетевой график выполнения проекта по строительству гаража

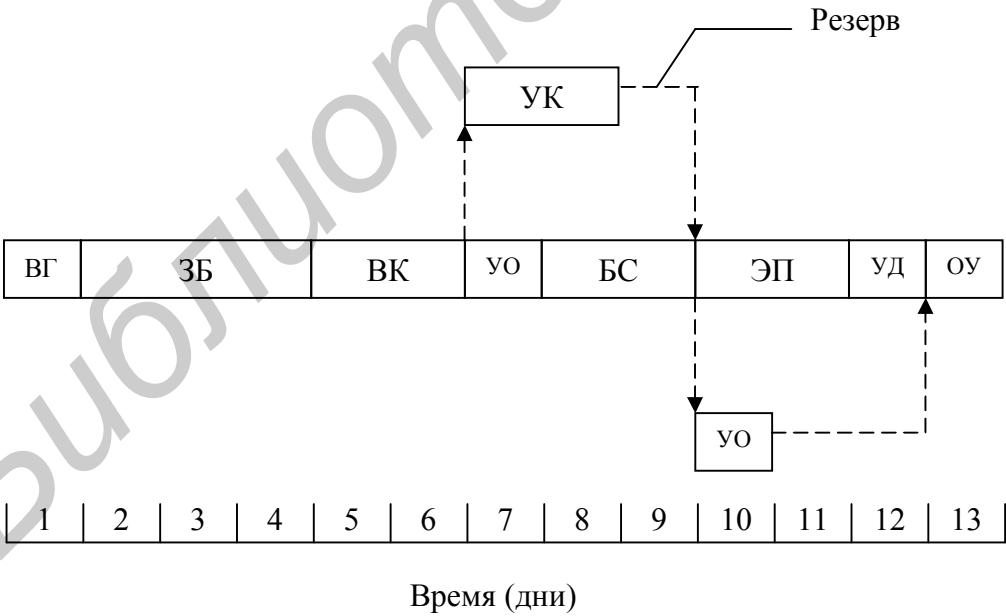


Рис. 6. Сетевой график строительства гаража (фрагмент)

Резерв, или запас времени – это разность между самым ранним возможным завершением работы и самым поздним допустимым временем ее выполнения. Резерв времени имеется только в тех операциях, которые не лежат на критическом пути, и дает некоторую гибкость при календарном планировании работ.

Для автоматизации управления проектом используются специализированные программные средства.

Microsoft Project – популярный программный продукт, имеющий удобные средства построения отчетов и ведения информации, необходимой для принятия управленческих решений, эффективные алгоритмы ее анализа, а также возможность интеграции с другими приложениями Microsoft Office.

Microsoft Project рассматривает проект как совокупность задач. Каждая задача имеет название, длительность, дату начала и дату конца. Ключевой характеристикой задачи является ее длительность. Руководитель планирует длительность задач проекта и связывает их между собой.

Проекты могут делиться на фазы. В проектах по разработке ПО можно выделить пять фаз:

1. Анализ.
2. Проектирование.
3. Реализация.
4. Стабилизация.
5. Внедрение.

Для выполнения задач назначаются ресурсы. Основным ресурсом являются люди, которые могут посвящать проекту часть или все рабочее время. В Microsoft Project можно указать, какие ресурсы будут связаны с каждой задачей. При этом длительность задачи связана с назначенными ей ресурсами – при добавлении или уменьшении выделенных задаче ресурсов Microsoft Project автоматически изменяет ее длительность. Управление ресурсами в рамках Microsoft Project позволяет определить загрузку персонала, стоимость выполнения отдельных задач, а значит и бюджет

проекта, посмотреть, как изменение ресурсов может повлиять на длительность отдельных задач и ход выполнения проекта в целом.

Системы качества и стандарты качества

Качество зависит не только от внутренних факторов, но и от факторов, находящихся во внешней среде организации. Большое значение имеют не технические качества продукта сами по себе, а то, насколько продукт соответствует требованиям потребителей. Качество – характеристика субъективная и может не соответствовать техническим качествам продукта. Например, фирма Opera выпустила особый, мощный, изящный Интернет-браузер, который занимает в 5 раз меньше места на жестком диске, чем его конкуренты MS Internet Explorer или Netscape Communicator. В то же время клиентам для повседневной работы нужна простая программа, которая не отнимает время на установку и настройку, не показывает рекламу и вообще не привлекает к себе особого внимания. Место на жестком диске в объеме нескольких десятков мегабайт не является критической характеристикой, сколько-нибудь важной для пользователя. Поэтому программой Opera пользуется менее 1% работающих в Интернете, а около 98% пользователей используют в качестве браузера встроенную в систему Windows программу MS Internet Explorer.

Существует два основных подхода к качеству: европейско-американский, когда вопросами качества занимается независимое подразделение, и японский, когда качество обеспечивается на местах, в том числе и через кружки качества.

Кружок качества – это группа служащих или рабочих одного подразделения, которые добровольно собираются на регулярные собрания для обсуждения проблем качества на своем участке и выдвигают идеи по улучшению качества.

Контроль качества заключается в разработке и регулярном осуществлении процедур контроля качества. Общий принцип организации контроля качества состоит в том, что, чем раньше найдется ошибка, тем дешевле обходится ее устранение.

При разработке системы качества предприятия обычно используются стандарты качества серии ISO9000.

ISO (International Organization for Standardization, www.iso.ch) – Международная организация по стандартизации, расположенная в Швейцарии. В 1987 г. впервые была принята серия документов ISO9000 – ISO9004, касающаяся комплексных систем управления качеством продукции и включающая следующие основные документы. С тех пор каждые несколько лет стандарты пересматриваются. В 2000 году принята новая редакция стандартов, предусматривающая так называемую процессную модель. Любая деятельность или операция, получающая входные данные и преобразующая их в выходные данные, может рассматриваться как процесс. Почти все операции на предприятии являются различными процессами, или, как сейчас говорят, бизнес-процессами.

ISO 9000 и ISO9004 содержат общие руководящие указания. ISO 9001-9003 описывают три различные модели качества: ISO 9001 – при проектировании и разработке, производстве, монтаже и обслуживании; ISO 9002 используется при производстве; наконец, ISO 9003 – при окончательном контроле и испытаниях. Последние две модели рекомендуются для серийного установившегося производства.

Основой разработки системы качества программистской организации служит ISO 9001, как, впрочем, для любой проектной организации. Главная цель этого стандарта – способствовать предупреждению отклонений от заданных требований на всех стадиях разработки от маркетинга до поставки. Для этого на предприятии выделяется специальный человек, отвечающий за качество. Если такого человека нет – нет и системы.

Рекомендации ISO9001 носят настолько общий характер, что их прямое использование невозможно. На базе стандартов ISO предприятию необходимо разработать собственную систему качества с учетом специфики и реалий предприятия, а затем уже внедрять ее.

Существуют специальные организации, которые имеют право сертифицировать другие организации на соответствие действующей системы

качества требованиям ISO. Подобная сертификация обходится весьма недешево. Белорусский государственный институт стандартизации и сертификации (БелГИСС) проводит испытания на соответствие стандартам РБ, которые идентичны стандартам ISO, фактически являясь их переводом с английского языка. В БелГИСС существует специальное подразделение, занимающееся сертификациями систем качества. Таким образом, белорусские предприятия могут пользоваться услугами БелГИСС.

Существуют и другие известные системы управления качеством – например, американская CMM (Controlling Maturity Management – контроль зрелости системы управления), концептуально несколько отличающаяся от ISO и считающаяся более жесткой, чем ISO.

Следует заметить, что сертификация системы качества необязательна. Главное – чтобы система была разработана, внедрена и использовалась в повседневной работе. Официальную сертификацию с привлечением внешних экспертов используют в маркетинговых целях для публичной демонстрации конкурентных преимуществ предприятия.

Сертификация – не самоцель, а средство повседневного улучшения всей деятельности предприятия. Периодическими аудиторскими проверками сертифицирующая организация не только понуждает к поддержанию высочайшей дисциплины, но и сотрудничает в поиске дальнейших мер по совершенствованию системы.

По данным Университета Суррея (Великобритания), исследовавшего 200 британских сертифицированных предприятий, рост производства в сертифицированных компаниях составил 4% для крупных предприятий и 6,8% для малых по сравнению с общим ростом в 2%. Уровень сбыта в расчете на одного работника в сертифицированных компаниях на 40% выше, окупаемость вложенного капитала на 7-17% выше. Отмечено, что прирост производительности труда от внедрения систем качества на малых предприятиях оказался значительно выше, чем на больших, а сбыт в расчете на одного работника, как обычно, значительно выше на крупных предприятиях.

Управление проектами разработки программного обеспечения

Законы Брукса

Фредерик П. Брукс был руководителем проекта IBM OS/360 в начале 60-х годов. Используя свой опыт, он написал неустаревающую книгу «Мифический человеко-месяц». В 1995 году вышло ее второе издание через 25 лет после первого. Оказалось, что за это время книга практически не устарела – в своем роде уникальный факт для компьютерной литературы, большинство которой устаревает уже в год выпуска. В США полагают, что без прочтения книги Брукса не может состояться ни один крупный руководитель программного проекта.

Итак, время от времени в прессе можно прочесть о том, как пара суперпрограммистов в переоборудованном гараже написала очень важную программу, превосходящую все, что было написано до нее. Тогда почему не закрыть софтверные фирмы и не создавать все программные продукты в переоборудованных гаражах?

Давайте разберемся с тем, что такое системный программный продукт.

Создание системного программного продукта состоит из 4 стадий (рис.7):

1-я стадия – это написание программы (ее можно написать и в гараже). Такая программа является завершенным продуктом, пригодным для запуска своим автором в системе, в которой была разработана.

2-я стадия – это создание программного продукта, т.е. программы, которую любой человек может запускать, тестировать, исправлять и развивать. Она может использоваться в различных операционных средах и со многими наборами данных. Чтобы стать программным продуктом, программа должна быть написана в обобщенном стиле, в частности, диапазон и вид входных данных должны быть настолько обобщенными, насколько это допускается базовым алгоритмом. Затем программу нужно протестировать,

чтобы быть уверенным в ее надежности. Наконец, развитие программы в программный продукт требует создания подробной документации, с помощью которой другой человек мог бы использовать ее, делать исправления и расширять.

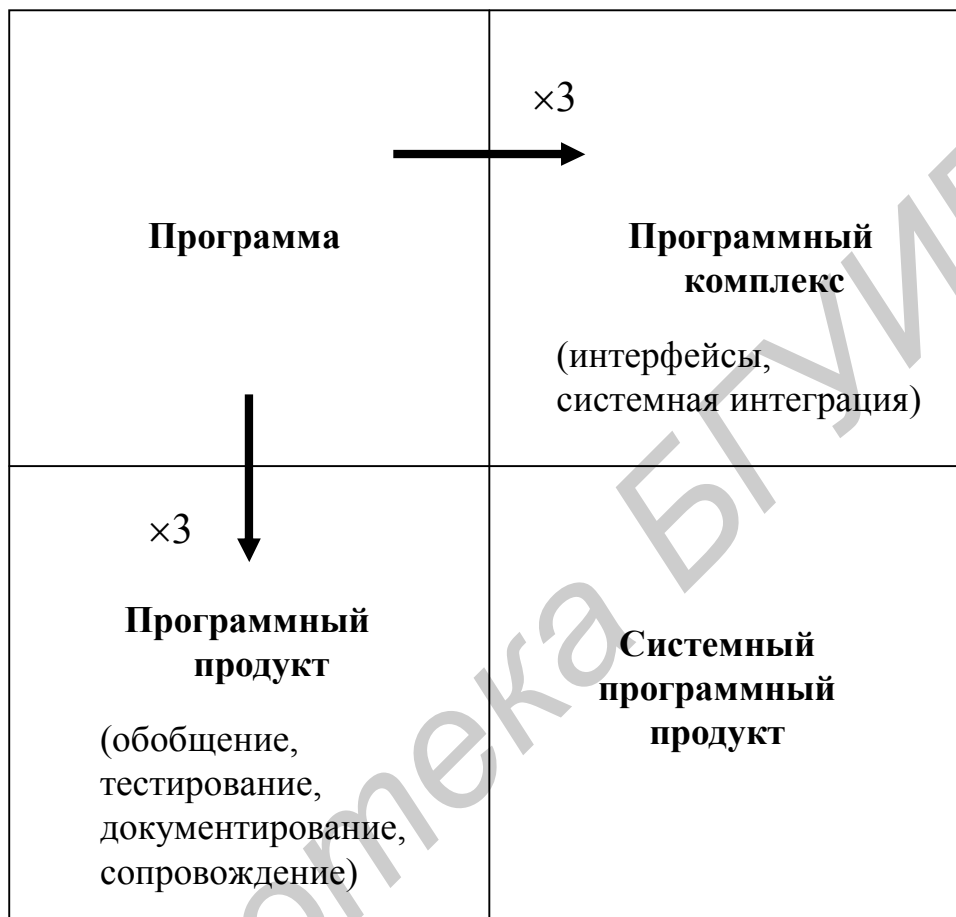


Рис. 7. Эволюция системного программного продукта.

На 3-ей стадии программа становится компонентом программного комплекса — набора взаимодействующих программ, согласованных по функциям и форматам, составляющих полное средство для решения больших задач. Чтобы стать частью программного комплекса, синтаксис и семантика ввода и вывода программ должны удовлетворять точно определенным интерфейсам. Программа должна быть также спроектирована таким образом, чтобы использовать заранее оговоренный бюджет ресурсов. Наконец, программу нужно протестировать вместе с прочими системными компонентами во всех сочетаниях, которые могут встретиться. Это тестирование может оказаться большим по объему, поскольку количество тестируемых случаев растет экспоненциально.

4-ая стадия – это создание системного программного продукта, объединяющего этапы 2 и 3.

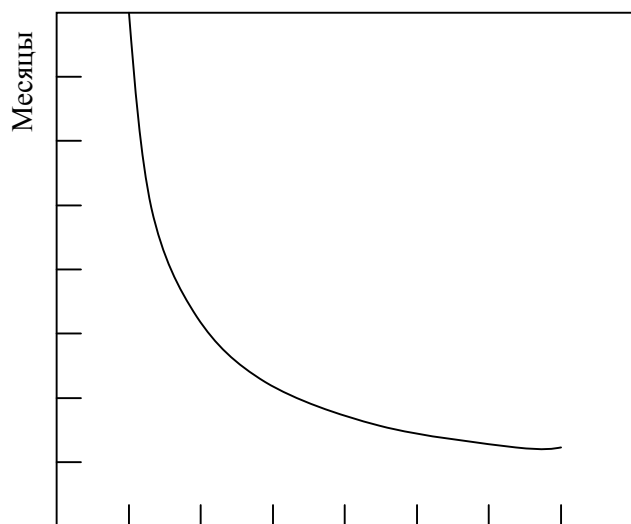
Любой из переходов с первой на вторую или с первой на третью стадии повышает трудозатраты втрое. Следовательно, системный программный продукт стоит в 9 раз дороже обычной программы.

Оптимизм. Все программисты – оптимисты. Как часто приходится слышать: «На этот раз она точно пойдет!» или «Я только что выявил последнюю ошибку!» Практика показывает, что оптимизм при создании программ порожден легкостью материала, из которого они сделаны. Программист осуществляет свои построения на основе чистого мышления – понятий и очень гибких их представлений. Раз материал так податлив, мы не ожидаем трудностей при реализации. Однако ошибки в программах возникают из-за ошибочности наших идей, поэтому глубокий оптимизм программистов ничем не обоснован.

Мифический человеко-месяц

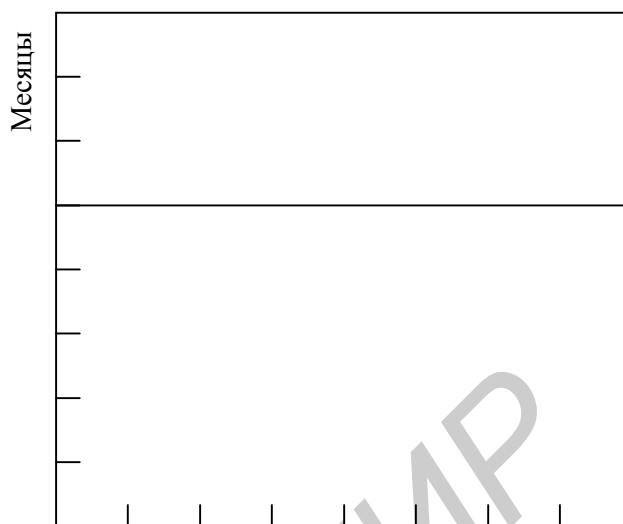
Вторая ошибка заключается в самой единице измерения, используемой при оценке сроков программных проектов: человеко-месяц. Люди и месяцы не взаимозаменяемы. Обычно задачу нельзя разбить на независимые части, и из-за этого разработка задач плохо распараллеливается. Чтобы родить ребенка, нужно 9 месяцев независимо от того, сколько женщин привлечено к решению этой задачи. Многие программистские задачи относятся к этому типу. Возможны следующие крайние случаи:

1. Задача полностью распараллеливается (рис. 8, а).
2. Задача вообще не распараллеливается (рис. 8, б).
3. Задача, требующая обмена данными. Для задач, которые могут быть разбиты на части, но требуют обмена данными между подзадачами, затраты на обмен данными должны быть добавлены к общему объему необходимых работ. Поэтому при добавлении рабочей силы срок разработки уменьшается медленно (рис. 8, в).



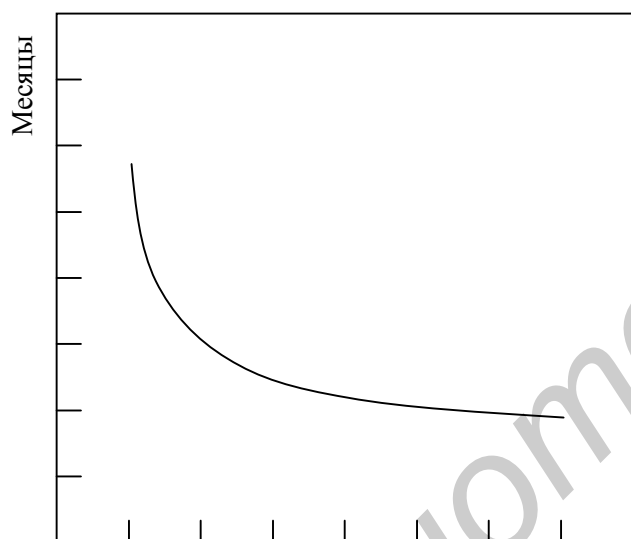
Число занятых

а



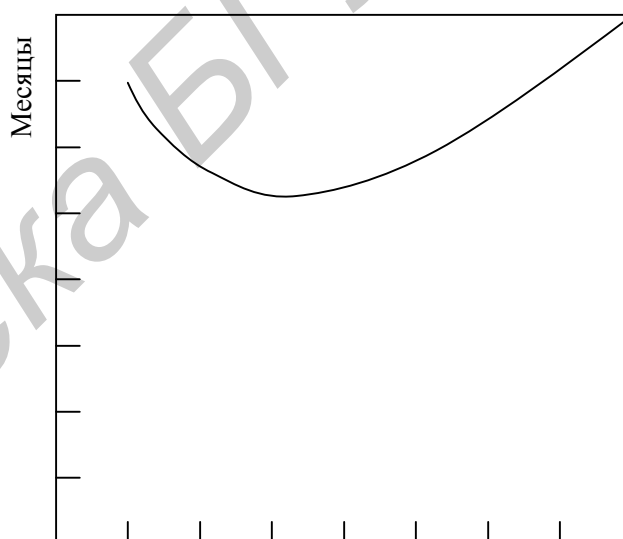
Число занятых

б



Число занятых

в



Число занятых

г

Рис. 8. Зависимость времени от числа занятых:

а — полностью делимая задача;

б — неразделимая задача;

в — делимая задача, требующая обмена данными;

г — задача со сложными взаимосвязями.

4. Задача со сложными взаимосвязями. Если все части задания должны быть отдельно скоординированы между собой, то затраты на обмен данными возрастают как $n(n-2)/2$. Для трех работников требуется втрое больше попарного общения, чем для двух, для четырех – вшестеро. При чрезмерном добавлении срок разработки снова начинает расти (рис. 8, г).

Существует ряд факторов, говорящих «против» добавления программистов в «горящие» проекты. Новые работники, какими бы знающими они ни были, и как бы быстро не удалось их найти, должны изучить задачу с помощью одного из опытных разработчиков. Обычно на это уходит не меньше месяца. Кроме того, задача, разбитая на определенное количество частей, теперь должна быть перекроена заново. Это значит, что часть уже сделанной работы будет потеряна, а системное тестирование нужно будет продлить. В итоге чаще всего получается худший результат, чем при сохранении прежней численности разработчиков.

Первый закон Брукса: *Если проект не укладывается в сроки, то добавление рабочей силы задержит его еще больше.*

Концептуальное единство

Концептуальное единство либо разобщенность лучше всего просматривается в архитектуре и строительстве. У большинства европейских соборов части, построенные разными поколениями строителей, имеют различия в планировке и архитектурном стиле. Более поздние строители испытывали соблазн «улучшить» проект своих предшественников, чтобы отразить новые веяния моды и свои личные вкусы. В результате общий вид такого собора создает впечатление перегруженности и неискренности. Архитектурное единство Реймского собора находится в прямой противоположности с таким смещением стилей. Его гармоничность и чистота стиля явились результатом того, что восемь поколений строителей смиряли свою гордыню в пользу сохранения концепции собора.

Хотя программные системы не создают восемь поколений разработчиков, в большинстве своем они демонстрируют еще меньшую

согласованность концепций, чем в любом собрании. Обычно это происходит не оттого, что главные проектировщики сменяют друг друга, а потому, что проект расщепляется на ряд задач, выполняемых разными разработчиками.

Концептуальная целостность является *важнейшей* характеристикой большого системного проекта. Концептуально целостная система быстрее разрабатывается и быстрее отлаживается. Поэтому, если есть важная идея, не вписывающаяся в основные концепции, то ее лучше оставить в стороне.

Поскольку целью проектирования является простота использования, окончательную оценку проекту дает достигнутое отношение функциональности к сложности концепций. Ни функциональность, ни простота в отдельности не являются признаками хорошего проекта. Важна простота и непосредственность.

И то и другое вытекает из концептуальной целостности.

Архитектура – это полная и подробная спецификация интерфейса пользователя. Архитектор является представителем пользователя, а не продавца или разработчика. Для того чтобы добиться концептуальной целостности, архитектура и разработка должны быть разделены. Архитектура говорит, *что* должно произойти, а разработка – *как* сделать, чтобы это произошло, т.е. предусматривает реализацию.

Второй закон Брукса: *Архитектура системы должна исходить из одной головы.*

Естественно, эта голова должна прислушиваться ко всем мнениям, но оставлять только то, что вписывается в первоначальную концепцию. Могут возникнуть возражения: не захватят ли архитекторы всю творческую деятельность, сделав исполнителей лишь винтиками в механизме? Не окажется ли, что более совершенный продукт можно получить, используя идеи всей бригады?

В действительности, разработка проекта требует не меньше творчества, новых идей, изобретательности, чем создание архитектуры. Это просто другая работа.

Практический опыт говорит о том, что в проектах, где проектирование архитектуры и разработка тщательно разделены, исполнители сразу сосредотачиваются на своей части задачи, в результате чего изобретательность бьет ключом. А вот в ничем не ограничиваемой группе большая часть обдумывания и обсуждения посвящена архитектурным решениям в ущерб реализации. Дисциплина идет на пользу. «Форма освобождает» – говорят художники.

Если Вы хотите, чтобы система обладала концептуальной целостностью, то управлением концепциями должен заниматься кто-то один.

Примеры концептуально целостных систем: UNIX, APL, Pascal, Modula, SmallTalk, Fortran, Windows NT.

Отрицательные примеры: Cobol, PL/I, Algol, MVS/370, MS-DOS, Windows 95.

Эффект второй системы

Когда руководитель проекта делает свою первую систему, проект стремится к скромности и ясности. При работе ему постоянно приходят в голову то одни, то другие «украшения». Они откладываются в сторону для использования «в следующий раз». После того как система закончена, архитектор с твердой уверенностью в себе готов к созданию нового проекта.

Эта вторая система является наиболее опасной для проектировщика. При работе над третьей и последующими системами, он уже приобретает необходимый опыт. Разрабатывая вторую систему, проектировщик стремится реализовать все идеи и украшения, отложенные в сторону при работе над первым проектом. В результате получается «большая куча», из которой используется едва ли половина возможностей. Примерами могут служить Windows 95 и MS Exchange.

Третий закон Брукса: *для руководителя проекта система не должна быть «второй», а если это случилось, то и ему самому, и его окружению необходимо максимально снизить соответствующие риски.*

Упрощение концептуальной сложности

Всемирно известный специалист в области программирования Э.Дейкстра как-то заметил, что «программирование сложно не только из-за больших задач, но и из-за наших маленьких голов».

Как определить, сколько времени потребует задача системного программирования? Прежде всего, нужно учесть, что оценки, полученные при создании отдельных небольших программ, нельзя применять для больших системных продуктов. Даже для небольших программ увеличение размера программы вдвое приводит к увеличению затрат и времени в 4, а то и в 6 раз. А ведь для больших проектов нужно еще добавить затраты на планирование, составление документации, тестирование, системную интеграцию и обучение. Линейная экстраполяция бессмысленна. Это все равно, что экстраполировать время, за которое возможно пробежать стометровку и получить, что можно пробежать три километра за пять с половиной минут.

Исследования показывают, что объем работы в больших проектах растет как степенная функция с показателем степени приблизительно 1,5:

$$\text{Объем работы} = (\text{константа}) \times (\text{число командных строк})^{1,5}$$

Из этой формулы следует, что производительность программиста, измеренная в элементарных операциях, постоянна. Известно также, что число ошибок прямо пропорционально числу командных строк. Следовательно, необходимо выбирать язык программирования как можно более высокого уровня. Статистические данные говорят о том, что при использовании подходящего языка высокого уровня производительность кодирования можно повысить в 5 раз.

Программа не перестает изменяться после своей поставки клиенту. Изменения после поставки – это сопровождение. Сопровождение программ состоит главным образом из изменений, исправляющих конструктивные дефекты. Очень часто эти изменения включают в себя дополнительные функции, которые видны пользователю.

Общая стоимость сопровождения широко используемой программы обычно составляет 40 и более процентов стоимости ее разработки. Как ни странно, на стоимость сопровождения сильно влияет число пользователей. Чем больше пользователей, тем больше ошибок они находят.

Главная проблема при сопровождении программ состоит в том, что исправление одной ошибки с большой вероятностью (20–50 процентов) влечет появление новой. Поэтому весь процесс идет по принципу «два шага вперед, шаг назад».

Все исправления имеют тенденцию к разрушению структуры, увеличению энтропии и дезорганизации системы. С течением времени все меньше сил тратится на исправление ошибок исходного проекта и все больше – на ликвидацию последствий предыдущих исправлений. Начиная с какого-то момента, число ошибок в программе больше не снижается. Кроме того, меняются машины, конфигурации, требования пользователя, так что фактически ни одна система не может оставаться вечной. Становится необходим новый проект, выполненный с самого начала.

Правила системной отладки:

1. Используйте отлаженные компоненты.
2. Создайте побольше инструментов среды разработки: отладочных тестов, программ, заглушек, генераторов тестовых данных.
3. Ведите жесткий контроль за изменениями.
4. При сборке добавляйте компоненты по одному.
5. При сопровождении уже установленного продукта делайте изменения большими порциями, чтобы периоды стабильности были больше.

«Серебряной пули» нет

Этот доклад Ф.Брукса на конференции 1986 г. в Дублине вызвал большой отклик в прессе. Речь идет о методах борьбы с концептуальной сложностью.

Хорошо знакомый программный проект весьма напоминает оборотней из фильма ужасов, считает Брукс. Будучи простым и невинным на вид, он может стать чудищем проваленных графиков работы, раздувшихся бюджетов и неработающих продуктов. Мы ищем серебряную пулю, которая могла бы волшебным образом уложить оборотней наповал. Нечто, способное снизить стоимость программных продуктов так же резко, как снизилась стоимость компьютеров.

Брукс утверждал, что в ближайшее десятилетие после этого доклада (1986 г.) не будет сделано открытий ни в технологии, ни в методах управления, использование которых обещало бы хоть на порядок повысить производительность, надежность и простоту программного обеспечения. Сейчас идет уже третье тысячелетие, а его предсказание все еще остается в силе. Панацеи вроде объектно-ориентированного программирования, систем искусственного интеллекта и экспертных систем себя не оправдали. «Серебряной пули» нет, и вряд ли она скоро появится.

Однако скептицизм – это не пессимизм. И хотя Брукс совершенно справедливо считает ошеломляющие прорывы не свойственными самой природе программирования, он указывает на многие перспективные нововведения, дисциплинированное и последовательное внедрение которых действительно может дать многократный рост:

1. *Покупать, а не создавать.* Развитие массового рынка позволяет избежать создания того, что можно купить. Любой «коробочный» продукт намного дешевле купить, чем создавать заново. Кроме того, массовый рынок и распространение программных пакетов привели к появлению метапрограммирования – создания нового слоя, приспособляющего функции к нуждам определенной группы пользователей пакета. (Например, как в 1С:Предприятие). Метапрограммирование значительно повышает производительность программиста.
2. *Уточнение требований и быстрое прототипирование* как часть запланированных итераций для установления требований к ПО. Это направление предусматривает быстрое создание макета системы,

который моделирует главные интерфейсы и выполняет основные функции предполагаемой системы. Назначение макета – показать, как воплощается выбранная концептуальная структура, чтобы клиент мог проверить ее пригодность к использованию и непротиворечивость.

3. *Пошаговая разработка.* Следует органично наращивать (выращивать) программы, а не строить сразу. Сначала систему надо заставить просто выполняться, даже если при этом она не делает ничего полезного, кроме вызова некоторых фиктивных подпрограмм. Затем нужно добавлять к системе все большую функциональность по ходу ее разработки, внедрения и эксплуатации, причем делать это параллельно с тестированием.
4. *Выращивание выдающихся проектировщиков.* Хорошим практическим приемам можно обучить, однако создание программ является *творческим* процессом. Выдающиеся проекты создаются выдающимися проектировщиками. Поэтому каждая организация, занятая в программировании, должна как можно раньше выявлять и растить выдающихся разработчиков концепций нового поколения.

Стратегия открытых исходных текстов как альтернатива концепции Брукса

Эрик Раймонд в своей нашумевшей статье «Собор и базар» попытался опровергнуть Брукса, утверждая, что якобы стратегия открытых исходных текстов OpenSource, при помощи которой был разработан Linux, полностью опровергает основные положения книги Брукса. Однако профессор университета в Нью-Джерси Н.Н. Безруков (автор некогда знаменитой «Вирусологии» – первой книги о вирусах на русском языке) в своей известной статье опроверг Э.Раймонда по всем пунктам.

Во-первых, в случае Linux речь идет о переписывании уже существующей системы, когда всем известно, что в точности система делает и из каких модулей должна состоять. При ее разработке не требовалось вырабатывать концепции, создавать спецификации и заниматься

проектированием. Во-вторых, никакой демократии в OpenSource нет и в помине, дисциплина на самом деле очень жесткая. Ее осуществляют высшие модераторы по своему произволу, давая дорогу модулям, разработанным одними разработчиками, и отбрасывая модули, разработанные другими. Занимается этим и Линус Торвальдс – главный модератор проекта Linux.

Лучшее качество программ OpenSource или GNU из-за их всеобщей доступности – это тоже миф, ибо никто не ставит цель выявить в них ошибки. Кроме того, пока еще никем не доказано, что самодеятельное тестирование лучше профессионального. На самом деле, серьезных ошибок в Linux и «дыр» в ее безопасности – предостаточно, просто никто не пытается их искать с той же тщательностью, с какой их ищут, скажем, в продуктах корпорации Microsoft.

Классические ошибки управления проектами

Любой проект по созданию программного обеспечения зависит от четырех переменных: люди, процесс, продукт и технология. Это своего рода четыре *измерения* скорости разработки. Люди могут работать быстро, а могут затягивать задания. Процесс разработки и технология могут способствовать усилиям людей, а могут заставить их спотыкаться на каждом этапе. Конечный продукт может быть определен таким образом, что он будет наполовину готов исходя из одного только определения, а может быть определен так, что будет препятствовать даже самым героическим усилиям по его созданию.

Занимаясь разработкой программного обеспечения, нельзя уделять внимание какому-то одному фактору, забывая об остальных. Так как это четыре *измерения* быстрой разработки, то все они должны находиться в центре Вашего внимания постоянно. Вот перечень классических ошибок и методов борьбы с ними.

Люди:

1. *Не определена мотивация.* Никто не объяснил разработчикам, зачем и кому нужна разрабатываемая программа, какой выигрыш получат

ее заказчики и пользователи, чему научатся разработчики в результате работы над проектом и какое вознаграждение ждет их по окончании работ.

2. *Слабый персонал.* Этот пункт иллюстрирует старый анекдот советских времен. К работе над проектом обычно привлекаются ЖОРы, ДОРы, ЛОРы и СУКИ. ЖОРы – жены ответственных работников, ДОРы – дети ответственных работников, ЛОРы – любовницы ответственных работников, СУКИ – случайно уцелевшие квалифицированные инженеры.
3. *Неконтролируемые «проблемные» работники.* Таких работников нужно увольнять, иначе они будут разлагать моральный дух коллектива и в конечном итоге могут загубить весь проект.
4. *Героическое поведение.* Многие менеджеры считают, что «героическое» поведение разработчиков является весьма желательным. На деле, подчеркивание героизма в какой-либо форме приносит обычно больше вреда, чем пользы. Если менеджер проекта доверяет больше проявлению «героизма», чем устойчивому и последовательному процессу разработки и обязательным отчетам о проделанной работе, то он рискует срывом всего графика работ.
5. *Добавление людей в запаздывающий проект.* Наиболее классическая из всех классических ошибок (см. Первый закон Брукса).
6. *Шумные, густозаселенные помещения.* О том, какими они должны быть – см. ниже.
7. *Трения между разработчиками и заказчиками.* Заказчики могут быть недовольны разработчиками из-за того, что те отказываются от своих обещаний. Разработчики могут посчитать дополнительные требования заказчика чрезмерными, особенно после того как базовые требования к проекту уже утверждены. Прямой эффект от конфликта между двумя сторонами – плохие взаимоотношения. Косвенный эффект плохих взаимоотношений – плохое понимание

требований, плохо разработанный пользовательский интерфейс и, в самом худшем случае, отказ заказчика от готового продукта.

8. *Нереалистичные ожидания* – как заказчиков, так и разработчиков. Это наиболее распространенная причина трений между заказчиками и разработчиками. Нереалистичные ожидания не уменьшают сроки выполнения проекта, а содействуют пессимистическому настроению разработчиков.
9. *Отсутствие эффективного управления проектом*. Если проект не имеет опытного и авторитетного менеджера, то весьма вероятно, что другие высокопоставленные лица в организации будут навязывать разработчикам нереалистичные сроки выполнения проекта. Кроме того, без строгого контроля за изменениями и версиями проект может потерять свою целостность и даже работоспособность.
10. *Отсутствие одобрения главного представителя заказчика – «держателя акций»*. Тесная координация и в конечном счете успех проекта достигаются только в том случае, когда все ключевые фигуры (команда разработчиков, руководитель проекта, управляющий по маркетингу, спонсоры, конечные пользователи, заказчики, и др.) принимают в разработке проекта самое непосредственное участие.
11. *Пользователи не участвуют в процессе разработки*. Если пользователи не вовлечены в проект на самых ранних стадиях, то есть большой риск, что требования к проекту будут определены не правильно. В результате с вероятностью 75% работа будет отвергнута.
12. *Предпочтение «политической лояльности» профессионализму*. Среди разработчиков условно можно выделить четыре типа личностей. «Политики» сконцентрированы на карьерном росте, главное для них – взаимоотношения с начальством. «Исследователи» сосредоточены на поиске новых решений и сборе информации. «Изоляционисты» могут работать над проектом,

только при условии, что им будет предоставлена полная автономность. И, наконец, «Нормальные» объединяют в себе все вышеперечисленные типы. Начальство любит «политиков» и «нормальных», однако карьера «политиков» через год-полтора терпит фиаско – их «раскусывают».

13. *Надежда на авось (wishful thinking)*. Насколько распространен оптимизм среди программистов, см. в главе «Законы Брукса». Однако надежды на авось – это не только оптимизм. Эти надежды закрывают разработчикам глаза и заставляют их на что-то надеяться, когда для этого нет никаких оснований. Надежда на авось подрывает основной график работ; это первопричина всех остальных проблем проекта, причем в самых различных сочетаниях.

Процесс:

1. *Слишком оптимистический график*. Установление нереалистичных сроков работ подрывает эффективное планирование и ведет к сокращению таких критически важных видов деятельности, как анализ требований и проектирование. В конечном итоге, накопившиеся ошибки могут привести к срыву проекта.
2. *Недостаточное управление рисками*. Если Вы не управляете рисками достаточно активно, то даже одно происшествие превратит Ваш проект быстрой разработки в проект медленной разработки.
3. *Подвел подрядчик*. Если фирма перегружена работой, она может передать часть проекта внешнему подрядчику. Однако подрядчики часто затягивают сроки, выпускают программный продукт низкого качества либо не соответствующий спецификациям. Риск только усиливается, если Вы не знакомы лично с подрядчиком, и общаетесь с ним через электронную почту. Взаимоотношения с подрядчиком нужно всячески поддерживать, в противном случае включение его в работу скорее замедлит, чем ускорит проект.
4. *Недостаточное планирование*. Если Вы не планируете высокую скорость разработки, как Вы можете надеяться ее достичь?

5. *Отказ от планирования из-за отставания.* Наоборот, при отставании необходимо ежедневное планирование и ежедневный контроль.
6. *Потеря времени на слишком детальное предварительное планирование.* Гораздо проще и менее рискованно выделить несколько недель из времени предварительного планирования, чем урезать их из напряженного графика разработки.
7. *Сокращение сроков на проектирование* (анализ требований, архитектуру, дизайн) с ориентацией на кодирование и другие якобы более понятные и легко достижимые цели.
8. *Неадекватные проектные решения.* В проектах, которые приходится выполнять в сжатые сроки, нет времени на тщательное проектирование и поиск потенциальных проблем. В результате сам проект становится неадекватным. Необходимо несколько трудоемких циклов проектирования перед окончательным выпуском системы.
9. *Экономия на контроле качества.* Исследования фирмы IBM показали, что качество и соблюдение сроков коррелируют между собой. Каждый час, затраченный на анализ качества, экономит от 3 до 10 часов работы. Это же соотношение установлено для ошибок, допущенных на этапе проектирования (см. предыдущий пункт) – они, как известно, самые дорогие.
10. *Недостаточный управленческий контроль.* Чтобы удержать проект в запланированных рамках, Вы должны четко знать, когда и где ход выполнения работ впервые вышел за эти рамки.
11. *Слишком ранняя подготовка комплекта поставки задолго до окончания разработки.* В проектах со сжатыми сроками это делается с целью скорейшего окончания работ. В результате работу приходится по 5–6 раз переделывать (улучшать производительность, распечатывать документацию, систему подсказок, шлифовать, разрабатывать и перерабатывать инсталляционную программу).

12. *Появление работ, не предусмотренных графиком.* Чтобы такого не случилось, необходимо протоколировать историю предыдущих проектов. Из-за подобных работ проект затягивается на 20-30%.
13. *Планирование догнать график.* Эта попытка всегда обречена.
14. *«Жертвенные» проекты.* Существует стратегия быстрой разработки под названием Code-like-Hell: нанять лучших людей, попросить их полностью посвятить себя проекту (работа по 80-100 часов в неделю), гарантировать им почти полную автономию, мотивировать их до высшей степени. Так была сделана Windows NT 3.0. Такая стратегия подходит, когда нужно срочно выпустить продукт на рынок, а денег мало. Однако она имеет ряд недостатков: код весьма запутан, нет гарантии успеха, люди могут сломаться морально посередине проекта. Нельзя узнать заранее, когда такой проект закончится. После окончания проекта наблюдается большая утечка мозгов. Проект значительно поглощает ресурсы человека: его участники забывают о семьях, друзьях, хобби, здоровье; люди «сгорают». Серьезные личностные конфликты в таких проектах – скорее правило, чем исключение. Подобная жертвенность, возможно, и нужна для запуска человека на Луну или в выигрыше войны, но не для разработки делового ПО. В жертвенности нет необходимости: за небольшими исключениями, те же результаты достигаются хорошим управлением и техническим планированием с куда меньшими усилиями. Подход Code-like-Hell гарантирует жертвенность, но не гарантирует успех.

Продукт:

1. *Слишком большое число требований.* Многие проекты имеют больше требований, чем это необходимо. Например, заданный уровень производительности формулируется в виде требования гораздо чаще, чем это необходимо, а это может увеличить сроки разработки.

2. *Изменение требований в процессе разработки.* Это, в сущности, неизбежно. В среднем, в течение разработки требования к проекту изменяются на 25%. Поэтому обычно не имеет практического смысла начинать разработку проекта, который нельзя внедрить уже через несколько месяцев.
3. *Попытки разработчиков использовать новые возможности там, где это не нужно.* Разработчики бывают очарованы различными новыми технологиями либо могут захотеть создать свою собственную реализацию блестящих возможностей, которые они видели в других продуктах независимо от того, нужно это для их собственного проекта или нет. Усилия, необходимые для проектирования, реализации, тестирования, документации и поддержки этих возможностей, тем не менее, вовсе не нужны для соблюдения графика проекта.
4. *Добавление задач при переносе сроков.* Часто при запаздывании проектов говорят: «Хорошо, мы согласны на перенос сроков, но тогда Вы нам сделайте еще то, то и то». Такие просьбы нужно отвергать, в противном случае, новый график работ окажется под угрозой срыва.
5. *Превращение разработки в исследование.* Яркий пример – история Сеймора Крея. Первая фирма Крея по разработке суперкомпьютера – Cray Research – была весьма успешной. Ее суперскалярные суперкомпьютеры из 64 и 256 процессоров известны по всему миру. Крею захотелось создать еще более мощные суперкомпьютеры, но он не нашел понимания среди акционеров. Тогда Крей продал свою долю в фирме Cray Research другим владельцам и на вырученные деньги основал другую фирму – Cray Supercomputer. Новая фирма чрезмерно увлеклась исследованиями и затянула выход дорогого суперкомпьютера на несколько лет. За это время лучшие компьютеры других производителей приблизились по характеристикам к проектируемым суперкомпьютерам. В результате стратегические заказчики Cray Supercomputer (НАСА,

астрофизические лаборатории, правительство), для которых и предназначались будущие суперкомпьютеры Крея, в принципе отказались от идеи их приобретения и авансирования дальнейшей разработки. Фирма обанкротилась, а ее 140 высококвалифицированных сотрудников быстро стали добычей других фирм Кремниевой долины. Буквально через пару лет Крей умер, будучи далеко не старым человеком.

Технология:

1. *Синдром «серебряной пули».* В главе «Серебряной пули нет» показывалось, что широко разрекламированные средства, повышающие производительность труда программистов, в действительности не дают нужный эффект, особенно, если нет полной информации об их применении в конкретной интегрированной среде. Если команда разработчиков хватается за новые технологии, чтобы «вытянуть» запаздывающий проект, их применение затянет проект еще больше.
2. *Завышенные ожидания от применения новых технологий разработки.* Практика показала, что если поставщики декларируют повышение производительности на 100%, то в действительности этот показатель редко превышает 10%.
3. *Смена средств в середине разработки.* Иногда имеет смысл постепенно обновлять средства разработки, при переходе, например, с версии 3 на 3.1. Но полная смена инструментария в разгар работ по проекту не приведет ни к чему хорошему из-за затрат на обучение, переделывание работы и неизбежных ошибок.
4. *Отсутствие автоматического контроля версий исходных текстов.* Это приводит к ручной координации программных проектов. Часто может оказаться, например, что используется не та, что нужно, версия библиотек компонентов, что ведет к переделыванию работы. Типичные средства автоматического контроля версионности – PVCS, InterSolv, Microsoft Visual SourceSafe.

В итоге, индустрия ПО эффективна всего на 35% (по результатам анализа 4000 проектов). Остальные 65% времени занимают:

- использование средств повышения производительности, которые не работают;
- корректировка беззаботно разработанных модулей;
- потеря результатов работы из-за отсутствия управления конфигурацией (версионностью).

Три самые опасные вещи в процессе разработки – это переделывание уже готовых модулей, изменение постановки задачи и задержки принятия решений заказчиком.

Планирование быстрой разработки

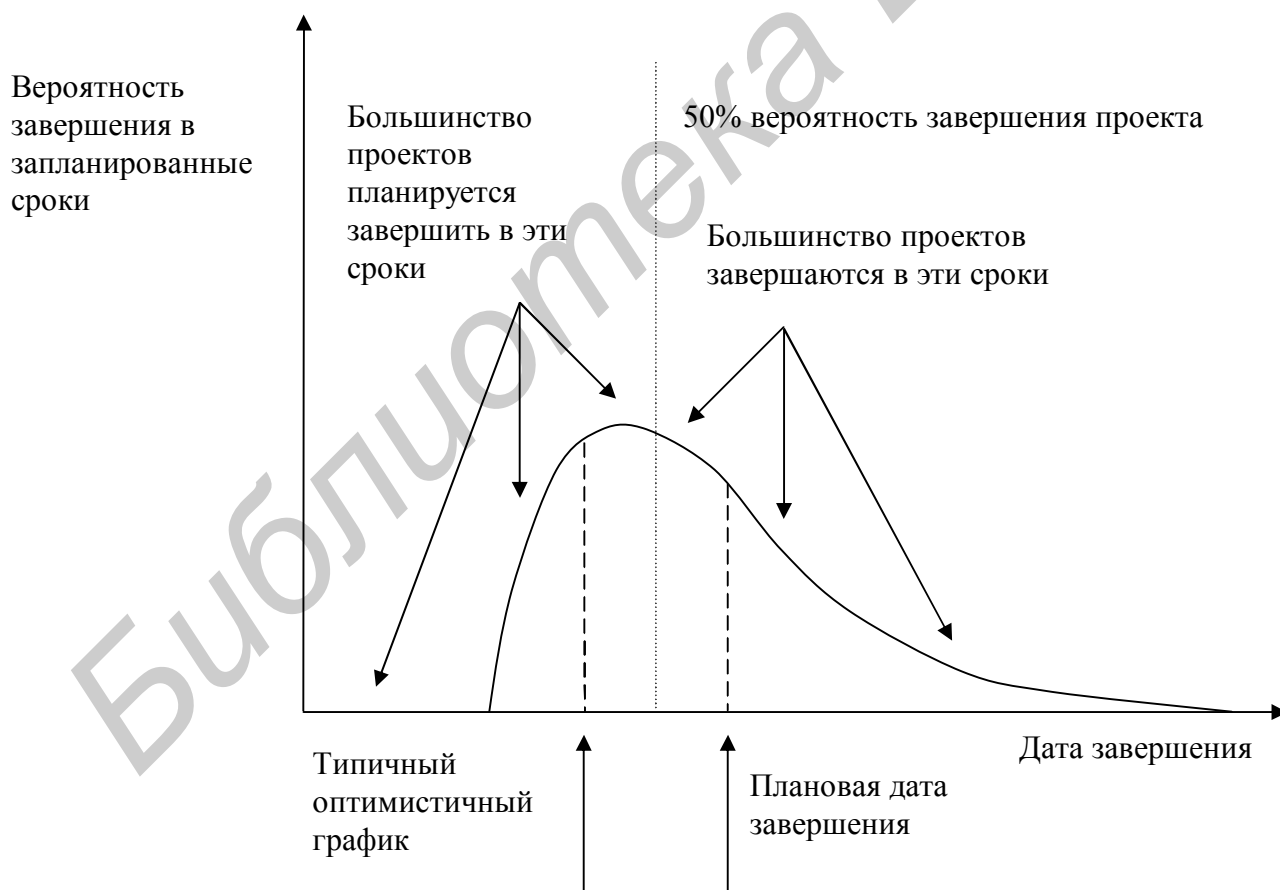


Рис.9. Типичный график выполнения проекта

Большинство сегодняшних проектов планируется окончить в сроки, которые не могут быть достигнуты ни при каких обстоятельствах. Менеджеры проектов часто даже не представляют всей амбициозности своих планов, и, как показывает рис. 9, обычно реализуют их гораздо позже запланированных сроков.

Основной причиной медленной разработки является надежда на авось. График работ должен быть реалистичным, чтобы между планируемой и реальной датой завершения проекта не образовывался временной зазор. Полезно использовать методы, позволяющие визуально отслеживать продвижение в выполнении проекта. Часто заказчики не столько заинтересованы в высокой скорости разработки, сколько просто хотят быть в курсе дел.

Трудоемкость проекта по этапам

Табл. 3 показывает, как обычно расходуется время при выполнении больших и малых программных проектов. Еще Ф.Брукс заметил, что кодирование занимает не более 1/6 части проекта, в то время как обыденное сознание программиста подсказывает обратное соотношение, и к нему опасно прислушиваться.

Таблица 3

Трудоемкость проекта по этапам, %

Этап работы	Малый проект (2,500 строк),%	Большой проект (500,000 строк),%
Проектирование архитектуры	10	30
Подробный проект	20	20
Кодирование и отладка	25	10
Тестирование модулей	20	5
Объединение	15	20
Тестирование системы	10	15

В малых проектах основное время занимают этапы разработки подробного проекта, кодирования и отладки и тестирование модулей. Если бы можно было каким-либо волшебным образом исключить все эти этапы из проекта, то мы сократили бы его на целых 65%. Для больших проектов все иначе. Исключение этих этапов снизит трудоемкость всего на 35%, так как основные усилия разработчиков направлены на разработку архитектуры, проектирование и системную интеграцию.

Для успеха проекта необходимо держать в равновесии три вещи – график, затраты и продукт (рис. 10). Последний включает в себя качество программного продукта, функциональность, полноту, пригодность для использования, модифицируемость, надежность и т.п.



Рис.10. Баланс графика, затрат и продукта

Чтобы треугольник был в равновесии, Вы должны, «растягивая» вершину «продукт», также «растянуть» «затраты» или «график», а чаще всего и то, и другое вместе. Стив МакКоннелл говорил, что во время обсуждения с заказчиками графика работ по проекту, ему очень помогал большой

треугольник, сделанный из картона. Когда заказчики показывали на вершину «График» или «Продукт» и говорили о своих дополнительных требованиях, он указывал им на оставшиеся вершины, и говорил, *что* необходимо сделать, чтобы треугольник оставался в равновесии. Если заказчик не хочет «отдать» Вам ни одной вершины треугольника, считает МакКоннелл, проект не может быть выполнен.

Существует два способа составления графика работ: максимальное ускорение и минимизация рисков. Возможности и недостатки этих методов иллюстрирует рис. 11.

Опыт показывает, что при разработке программного обеспечения возврат к предыдущим этапам и переделывание работы неизбежно. Лучше сразу учитывать это при планировании, т.е. спланировать график работ так, чтобы возврат на предыдущие этапы было легко осуществить.

Жизненный цикл разработки программного обеспечения

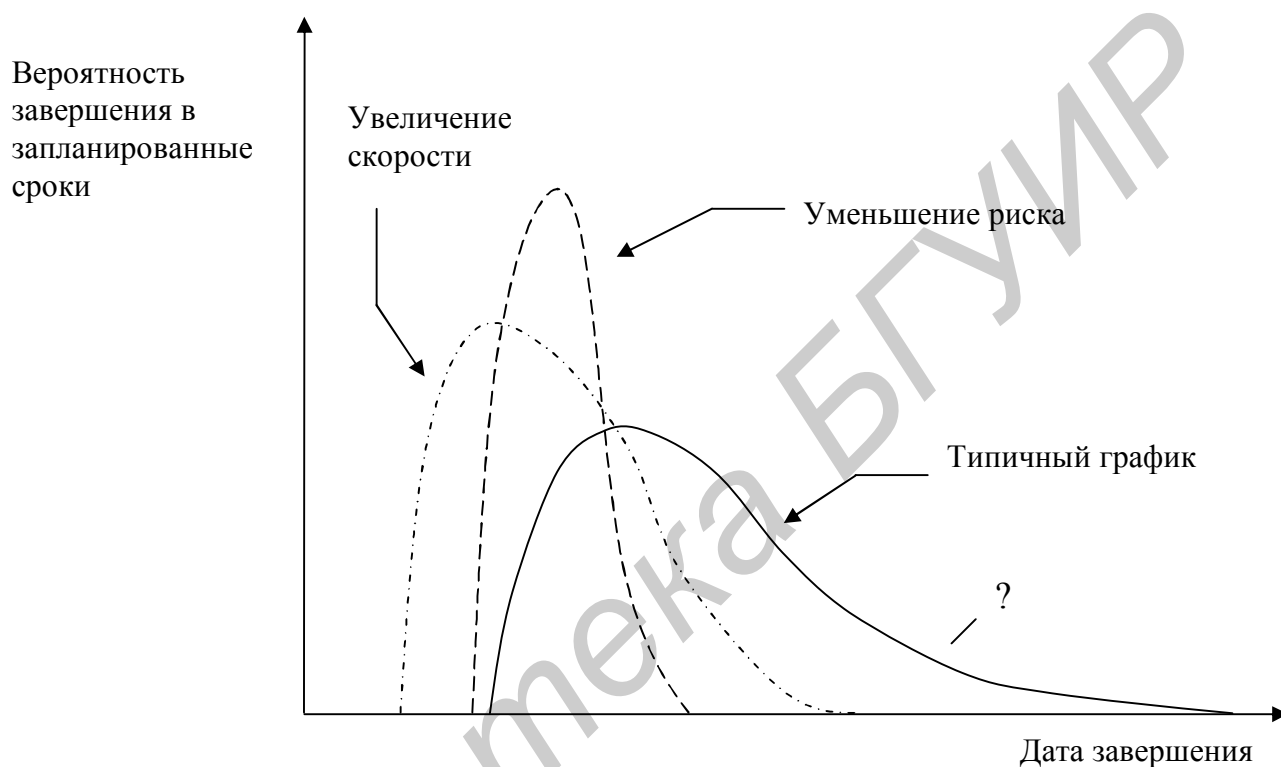


Рис. 11. Графики разработки, ориентированные на максимальную скорость и минимизацию рисков.

Типичные модели проектирования:

Метод «тыка» (Code-and-Fix). Программист, который использует этот метод, в начале имеет только общее представление о программе, которую он хочет написать. У него могут быть формальные спецификации, а могут и не быть. Затем он использует любые подходящие ему средства проектирования, кодирования, отладки и тестирования, пока не получает готовый продукт (или *не* получает).

Пожалуй, единственным достоинством метода «тыка» является отсутствие накладных расходов. Не нужно тратить время на планирование,

документацию, контроль за выполнением требований и т.п. Поэтому этот метод иногда используется для индивидуальной разработки коротких модулей. Для любых других проектов эта модель просто опасна, т.к. она не позволяет оценивать качество работ и уровень риска. Метод Code-and-Fix не подходит для проектов быстрой разработки.

Чистый водопад. Эта классическая модель проектирования является базовой для всех остальных моделей. Согласно ей, проект должен пройти ряд четко определенных этапов: Техническое задание – Технический проект – Рабочий проект – Эскизный проект – Опытный образец – Промышленный образец (сокращенно: ТЗ-ТП-РТ-ЭП-ОО-ПО) (рис. 12.). Новый этап не может начаться, пока не закончится предыдущий. Таким образом, получается нисходящая цепочка, напоминающая водопад.

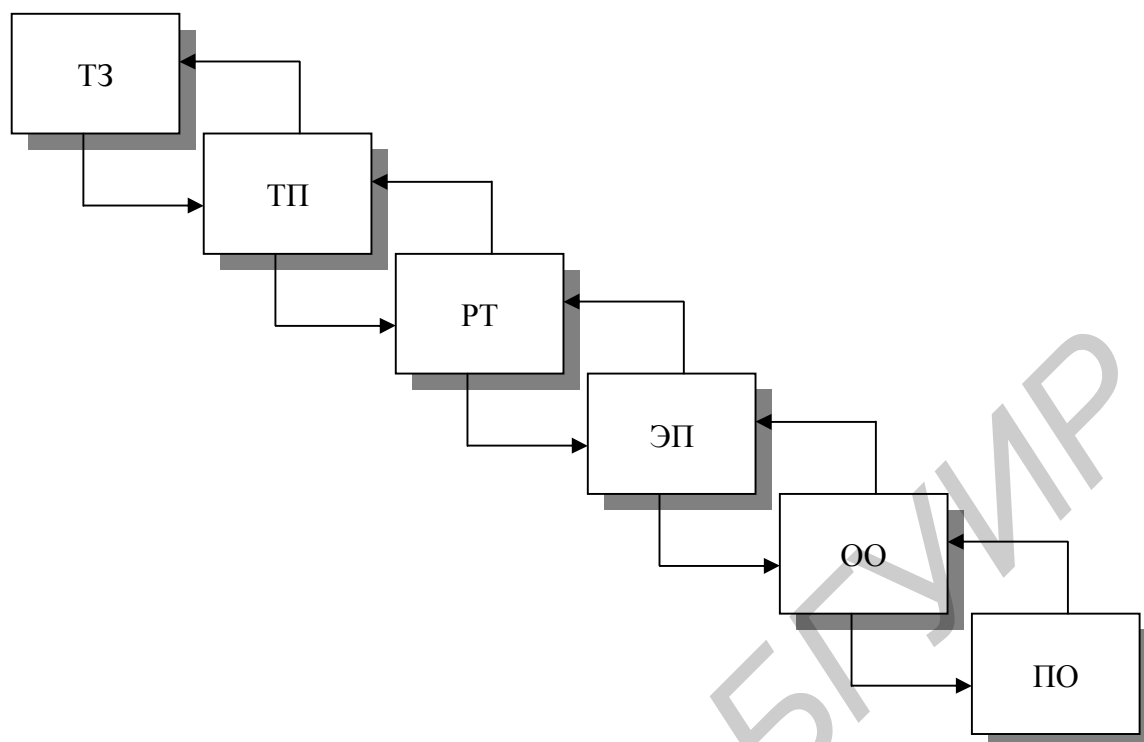


Рис. 12. Чистый водопад

Данная модель, широко применяющаяся в технике и соответствующая ГОСТам, исходит из предположения о том, что выполняемый проект типовой и хорошо определен, а используемые материалы и технологии заранее известны. Однако это допущение по отношению к программированию верно лишь для очень малых или хорошо определенных проектов, из-за чего в применении к разработке ПО модель чистого водопада практически утратила актуальность. В то же время ряд перечисленных выше документов (особенно ТЗ) широко используются в качестве юридических документов при заключении договоров для фиксации основных функций и характеристик разрабатываемой системы.

Модифицированный водопад представляет собой развитие предыдущей модели. Предусматривается перекрытие этапов во времени, разделение на подзадачи, снижение рисков. Эта модель также устарела.

Спиральная модель. Спиральная модель – это ориентированная на риски модель жизненного цикла, которая позволяет разбить большой

программный проект на минипроекты. В каждом таком минипроекте устанавливаются один или более главных рисков (непонятные требования, плохо разработанная архитектура, потенциальные проблемы в исполнении и т.п.). После того, как риски определены и намечены пути их снижения, минипроект выполняется по обычной схеме «чистого водопада». Затем следует переход на следующую итерацию, которая перемещает проект в больший масштаб. Вы заканчиваете один уровень, проверяете – то ли это, что Вы хотели получить, и начинаете работу над следующим уровнем (рис. 13).

Каждая итерация включает 5 этапов:

1. Определение объектов, альтернатив и ограничений.
2. Определение и снижение рисков.
3. Оценка альтернатив.
4. Разработка продукта для данной итерации, проверка его корректности.
5. Планирование и переход к следующей итерации.

Преимущество спиральной модели заключается в том, что она позволяет снижать риски по мере возрастания стоимости разработки. После каждой итерации есть контрольная точка, позволяющая определить, соответствует ли разработка ожиданиям заказчика. Если проект не может быть выполнен по техническим или каким-либо другим причинам, использование спиральной модели позволит узнать об этом достаточно рано, пока в разработку еще не вложены большие средства.

Для того чтобы спиральная модель разработки принесла пользу, необходимо тщательное, сложное и точное управление.

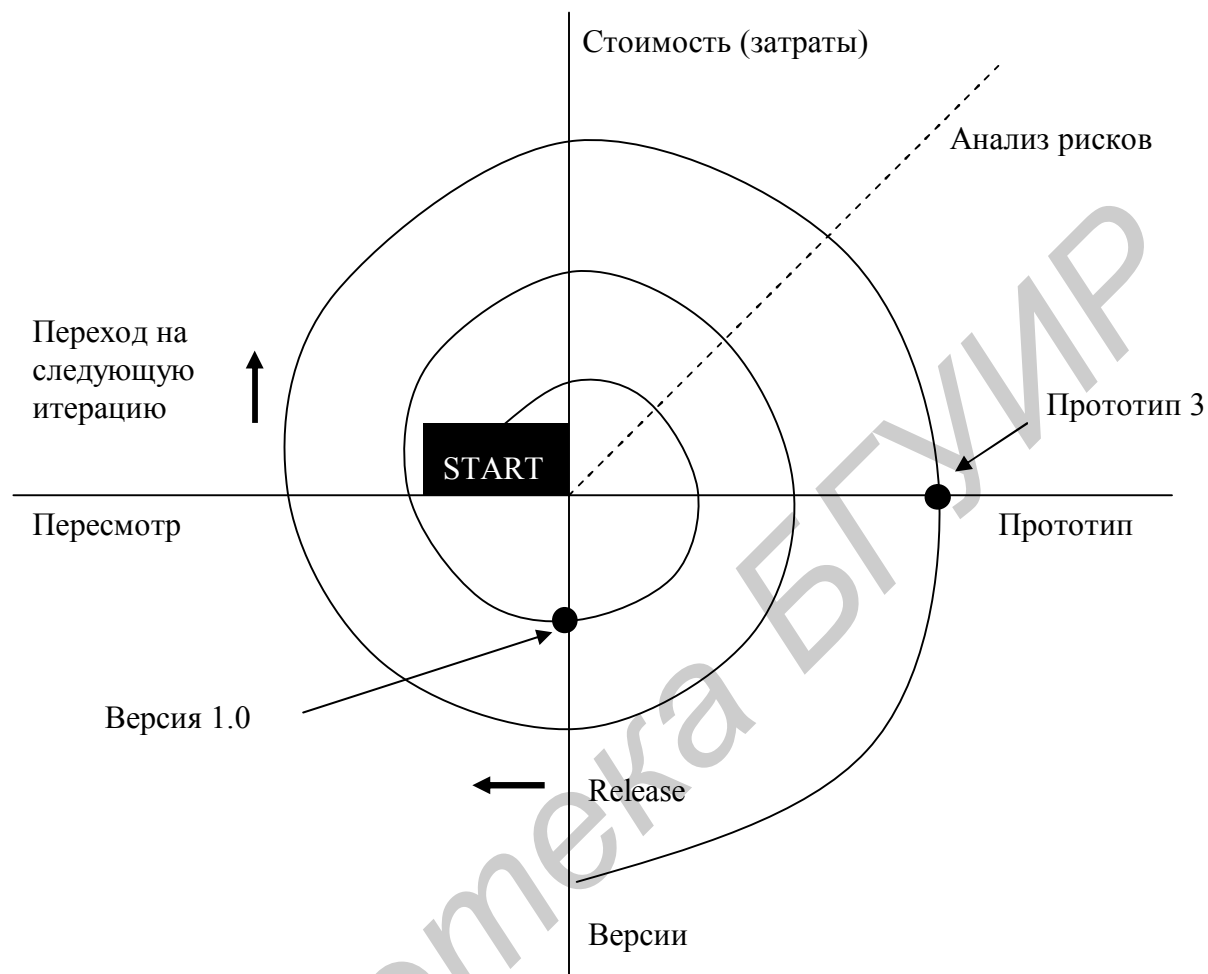


Рис. 13. Спиральная модель разработки программного обеспечения

На сегодняшний день наиболее перспективными считаются эволюционные модели.

Эволюционная поставка по рабочим версиям. Согласно этому методу, в первую очередь реализуются самые важные функции, во вторую очередь – менее важные и т.д. Данный подход особенно эффективен при быстром изменении обстановки. Модель отлично работает при организации подряда, ибо хорошо ложится на договорные отношения и понятна

заказчику, который платит только за те функции, которые относятся к задачам соответствующей версии.

Эволюционная поставка, исходя из имеющихся ресурсов характеризуется разработкой максимального набора функций, укладываемого в установленные ресурсы: сроки или деньги. Разработка идет до тех пор, пока указанный ресурс не исчерпается. Такая модель характерна для проектов, предназначенных для собственного использования либо для продажи «коробочных» решений на рынке.

Эволюционное прототипирование. Данный подход подразумевает создание прототипа и его длительную доводку до рабочей программы. Разработку следует начинать с наиболее существенных, видимых аспектов системы. Эта часть демонстрируется заказчику, корректируется, после чего разработка прототипа продолжается. Когда и разработчики, и заказчик соглашаются с тем, что прототип «достаточно хорош», работы завершаются, и прототип выпускается как готовый продукт.

Эволюционное прототипирование особенно полезно, когда требования к проекту постоянно изменяются, или когда и вы, и заказчик не владеете предметной областью достаточно хорошо.

Этот метод обычно используется Microsoft – обратите внимание на номера выпусков (build) любой выпускаемой ими программы, которые всегда можно найти в разделе «О программе» пункта «Справка».

И последний, ныне ведущий метод: **внедрение готового ПО**. Опыт показывает, что заказчик, который еще недавно так настаивал на собственной разработке, очень скоро об этом забывает. Горячее желание пользоваться собственной разработкой исходит из его нереалистических ожиданий, которые на практике никогда не оправдываются. Речь идет, конечно же, о якобы повышенных характеристиках функциональности, надежности, качестве собственной разработки по отношению к типовой. Опыт показывает, что это не так. Качество стандартной разработки всегда выше доморощенной, а попытки догнать и перегнать типовую разработку – обречены. Ведь

разработчик типового проекта тоже не стоит на месте и непрерывно работает над улучшением своей продукции.

Руководству непрограммистской организации (или его подразделения, занимающегося эксплуатацией) следует строжайшим образом следить за тем, чтобы это подразделение не начало выполнять несвойственные ему функции. Превращаясь в центр разработки программ для собственной организации, оно «проедает» прибыль предприятия и снижает его эффективность. Так, крупные западные банки, которые, как правило, используют исключительно собственные разработки, тратят до 10% прибыли на компьютеризацию, в то время как для отечественных банков и небольших банков, использующих типовые проекты и типовую технику, этот показатель редко превышает 1-2%.

Оценка трудоемкости

Без сомнения, планирование – это самый ответственный для менеджера момент. По Ф.Бруксу, неумение правильно оценить объем проекта является главной причиной срыва сроков разработки.

Для оценки трудоемкости необходимо:

1. Оценить объем продукта.
2. Оценить усилия разработчиков, необходимые для создания продукта.
3. Оценить график работ по проекту.

Пример истории создания Microsoft Word 1.0 – отличная иллюстрация классической ошибки «надежды на авось» (wishful thinking) в планировании. Эта ошибка характерна как для заказчиков, так и для руководителей высшего уровня, которые свято верят, что любая работа может быть выполнена за любое время, стоит только того пожелать.

Итак, в 1984 году Билл Гейтс, впечатленный успехами первых текстовых редакторов, приказал дословно «создать лучший в мире текстовый редактор за 12 месяцев». Как выяснилось впоследствии, для подобного проекта, который в результате занял 250000 строк, по нормам необходимо 26

месяцев. Даже если сильно сжать сроки, все равно проект никак нельзя было сделать менее чем за 22 месяца. В результате вышло следующее – табл. 4.

Таблица 4

Из отчета Microsoft Corporation: Office Business Unit, 1994

Месяц составления плана	Планируемый срок поставки	Планируемое число дней до поставки	Фактическое число дней до поставки	Относительная ошибка плани- рования, %
09.84	09.85	365	1887	81
06.85	07.86	395	1614	76
01.86	11.86	304	1400	78
06.86	05.87	334	1245	73
01.87	12.87	334	1035	68
06.87	02.88	245	884	72
01.88	06.88	152	670	77
06.88	10.88	122	518	76
08.88	01.89	153	457	67
10.88	02.89	123	396	69
01.89	05.89	120	304	61
06.89	09.89	92	153	40
07.89	10.89	92	123	25
08.89	11.89	92	92	0

Дело не только в том, что из-за неправильного планирования первая версия текстового редактора Microsoft Word создавалась 5 лет. Именно неверное планирование работ и попытки выполнить работу «нахрапом» привели к тому, что MS Word до сих пор является чрезвычайно громоздким продуктом со значительным количеством ошибок. Он требует для своей работы значительные ресурсы компьютера и резко замедляет работу при

редактировании даже не очень длинных документов. Его менее именитые конкуренты куда изящнее, компактнее, мощнее и эффективнее, как, впрочем, и другие продукты, входящие в Microsoft Office. Например, Microsoft Excel является весьма удачным продуктом.

Типичные причины оптимистических планов

1. Существуют внешние причины, определяющие срок выполнения проекта: компьютерные выставки, изменения в законодательстве, распродажи.
2. Менеджеры или заказчики строят планы на случай, что «все будет хорошо».
3. Необходимость заключить договор с заказчиком.
4. Разработчики хотят поучаствовать в интересном для них проекте.
5. Менеджер проекта считает, что под прессингом программисты будут работать лучше.
6. Давление внешнего заказчика, руководства или службы маркетинга.
7. Плохая оценка сроков.

Удивительно, что реалистические оценки сроков вообще появляются. Обычно программисты недооценивают время реализации проектов на 20-30%.

Как же ускорить разработку? Оказывается, чаще всего для быстрой разработки достаточно лишь правильно организовать работу. Быстрая разработка и эффективная разработка – это синонимы.

Методики оценки трудоемкости выполнения проекта

Оценка объема работ по функциональным баллам

Функциональные баллы – это специальная величина, которая используется для оценки размера программы. Функциональные баллы легче использовать, чем количество строк или слов программы, и они дают более точную оценку размера программы.

Количество функциональных баллов зависит от следующих величин:

- Число входных форм. Сюда входят диалоги, формы, сообщения, посредством которых конечный пользователь или другие программы могут добавлять, удалять или изменять данные программы.
- Число выходных форм – экраны, отчеты, графики или сообщения, которые генерирует программа для использования конечным пользователем или другими программами.
- Число запросов. Количество комбинаций «ввод/вывод», когда ввод имеет своим результатом немедленный вывод.
- Логические внутренние файлы – логические группы данных, которые полностью контролируются программой. Это может быть простой файл или таблица в реляционной базе данных.
- Внешние файлы – данные, контролируемые другими программами. Включают любые логические группы данных, которые вводятся или выводятся из программы.

Табл. 5 показывает, как подсчитать количество функциональных баллов в программе.. Множители взяты на основе данных нескольких тысяч проектов.

Таблица 5

Множители для определения функциональных баллов Характеристика программы	Низкая сложность	Средняя сложность	Высокая сложность
Число входных форм	×3	×4	×6
Число выходных форм	×4	×5	×7
Число запросов	×3	×4	×6
Логические внутренние файлы	×7	×10	×15
Внешние файлы (интерфейс)	×5	×7	×10

После умножения числа входных и выходных форм, запросов, внутренних и внешних файлов на соответствующие коэффициенты и суммирования этих чисел мы получим количество функциональных баллов. Однако для получения окончательного ответа это число необходимо домножить на дополнительный коэффициент сложности (передача данных, сложность обработки и внедрения), колеблющийся от 0,65 до 1,35.

Пример. Небольшой проект в средней организации (число входных форм – 6, выходных форм – 7, запросов – 0, внутренних файлов – 5, внешних файлов – 9) оценивается в 304 балла. Дополнительный коэффициент примем равным 1.15, тогда окончательное количество баллов – 350.

Срок выполнения проекта в месяцах определяется по формуле

$$\text{Срок} = \text{Баллы}^{0.44},$$

0.44 – это усредненный показатель. На практике он может колебаться в пределах 0.39–0.48. Уточнение показателей степеней Capers Jones для оценки сроков выполнения проекта, исходя из функциональных баллов, приведено в табл. 6.

Таблица 6

Уточнение показателей степеней Capers Jones

Тип программного обеспечения	Характеристика организации в своем классе		
	Лучшая	Средняя	Худшая
Системное	0.43	0.45	0.48
Бизнес-приложения, программы для использования внутри организации	0.41	0.43	0.46
Коробочный тиражируемый продукт	0.39	0.42	0.45

Пример. Если проект оценивается в 350 баллов и речь идет о разработке коробочного продукта в средней организации своего класса, то проект займет $350^{0,42} = 12$ календарных месяцев. Если такой же продукт разрабатывается в лучшей в своем классе организации, то сроки оцениваются в $350^{0,39} = 10$ месяцев.

Оценка длительности проекта по трудоемкости

Зависимость длительности проекта от его трудоемкости определяется следующим соотношением:

$$(\text{Длительность проекта в месяцах}) = 3.0 \times (\text{Человеко-месяцы})^{1/3}$$

Наиболее опытные специалисты рекомендуют прибавлять к этой оценке еще 30 процентов, а также использовать коэффициент от 2.5 до 4.0 вместо 3.0. Для правильного подбора коэффициентов необходимо использовать исторические данные организации.

Исходя из этих данных можно определить число разработчиков:

$$(\text{Число разработчиков}) = (\text{Человеко-месяцы}) / (\text{Длительность проекта в месяцах}).$$

Пример. Если известна трудоемкость проекта – 64 человеко-месяца, то несложно определить, что срок разработки составит 12–16 месяцев, при этом необходимо привлечь 5–6 разработчиков.

Оценка, основанная на рисках

Пример. Допустим, рассчитанная продолжительность проекта составляет 6 месяцев. Рассмотрим возможные риски и оценим их влияние на срок выполнения проекта.

Отрицательные факторы:

+1 месяц из-за запоздания поставки графического интерфейса;

+1 месяц из-за неработоспособности средства разработки;

+0,5 месяца по болезни персонала;

+0,5 месяца из-за недооценки размеров проекта;

Итого: +3 месяца.

Положительные факторы:

–1 месяц из-за меньшей, чем ожидалось, задержки найма новых работников;

–1 месяц из-за того, что новые средства разработки работают лучше, чем ожидалось;

Итого: –2 месяца.

Итоговая оценка: $\begin{matrix} +3 \\ -2 \end{matrix}$ 6 месяцев

Подобная оценка дает заказчику конкретную информацию о возможных рисках, а также позволяет управлять ими на ранних стадиях проекта.

Оценка по «случаям»

Оценка по «случаям» представляет собой вариант оценки, основанной на рисках. Согласно этому методу, необходимо оценить сроки проекта для наилучшего, наихудшего, запланированного и текущего случаев. Представляют интерес взаимозависимости между этими оценками. Если, к примеру, сроки для запланированного и наилучшего случая совпадают, а также совпадают сроки для текущего и наихудшего случая, то проект в

опасности. Табл. 7. показывает пример оценок для проекта, у которого пока нет серьезных проблем.

Таблица 7

Не самый худший проект

Тип случая	Оценка
В лучшем случае	1 апреля
В планируемом случае	15 мая
В текущем случае	30 мая
В худшем случае	15 июля

Оценка по вероятности

Один из вопросов, которые часто задают люди о рабочем графике, звучит примерно так: «Какова вероятность, что проект будет завершен к этой дате?» Используя оценку сроков проекта, основанную на вероятности, разработчики могут ответить на этот вопрос на основе взвешенных оценок, подобных тем, что приведены в табл. 8.

Таблица 8

Вероятностная оценка даты завершения проекта

Дата готовности	Вероятность завершения до указанной даты, %
1 апреля	5
1 мая	50
1 июня	95

Общие рекомендации по оценке сроков

Указанные измерения по мере разработки проекта необходимо периодически повторять с целью уточнения.

Найдите время для оценки сроков проекта, запланируйте его. Используйте данные из предыдущих проектов. Пусть оценивают не те, кто

будет делать работу, а другие программисты. И учтите, что их оценка занижена на 20–30%.

Чтобы оценка получилась объективной, она должна быть коллективной. Оценивайте проект по деталям, а затем суммируйте полученное время. При суммировании ошибки взаимно гасятся, и суммарная оценка погрешности уменьшается.

Функциональность системного продукта и время его разработки зависят от числа входных и выходных форм, запросов, внутренних и внешних файлов. Не забывайте, что обычные рутинные задачи – конвертирование данных, инсталляция, настройка, управление бета-тестированием, создание демо-версии, необходимость присутствия на совещаниях, сопровождение старого ПО, контроль качества, исправления, оформление документации, отпуска, праздники, болезни, производственные мероприятия, обучение – также требуют затрат времени.

Ведите исторические записи. Приобретите и используйте специализированное программное обеспечение для оценки объема работ.

Для получения более точных оценок необходимо использовать несколько различных способов, чтобы вывести усредненный результат. Усреднять можно способами, известными из статистики: получение среднего, среднегеометрического, среднего с отбрасыванием крайних значений-выбросов, без него и т.п.

При согласовании графика постарайтесь найти возможность, чтобы все участники проекта были довольны. Не становитесь в позу.

Не пытайтесь выжать из людей больше, чем можно. Существует определенный уровень стресса, при превышении которого положительное влияние стресса сменяется отрицательным. Чем сильнее давление графика, тем больше стресс, тем больше допускается ошибок и тем больше приходится исправлять работу. Это, в свою очередь, ведет к еще большему стрессу; возникает опасная положительная обратная связь, неизбежно ведущая к срыву графика.

Основные правила общения программистов с менеджерами и руководством при составлении графика:

1. Отделяйте людей от проблемы. Менеджеры часто вынуждены давать нереальные сроки из-за внешнего давления или по незнанию. Не реагируйте на эмоции, выработайте совместное решение.
2. Сосредоточьтесь на интересах, а не на позициях. Ссылайтесь на объективные обстоятельства, увеличение шансов на успех, привлеките истории других проектов.
3. Придумайте общую цель. Торг – это не игра с нулевой суммой, а упражнение в творческом решении задачи. Лучший торговец найдет путь, чтобы выиграли обе стороны.
4. Настаивайте на объективных критериях.

Правило корректировки графиков. Если проект опаздывает, то необходимо *растянуть* график, экстраполировав его пропорционально коэффициенту отставания (сжатия).

Просто отодвинуть дату окончания на число дней опоздания – это классическая ошибка. И уж, во всяком случае, не пытайтесь догнать график!

$$\text{Коэффициент сжатия графика} = \frac{\text{Желаемая длительность}}{\text{Номинальная длительность}} ;$$

$$\text{Цена сжатия графика} = \frac{\text{Номинальная цена}}{\text{Коэффициент сжатия графика}} .$$

Из этих двух формул следует, что при сжатии графика, например, на 17%, затраты растут на 21%, т.е. быстрее, чем сжимается график. В истории еще никому не удавалось сжать сроки более чем на 20–25%.

По этим же формулам можно оценить снижение затрат при растяжении графика.

Как вести исторические записи

К счастью, это совсем просто и не отнимает много времени.

Пример. Проект X.

Затраты, чел.-дней: (табл. 9).

Таблица 9

Затраты на проект по этапам, чел.-дней

Предварительная концепция	100
Окончательная концепция	100
Спецификация требований к продукту	135
Спецификация продукта	145
Подробная спецификация отдельных приложений	160
Завершение	170

Кроме того, необходимо вести записи следующих типов:

1. Менеджеры должны составить планы; все, что касается отношений с заказчиком или конечным пользователем; управление разработчиками; управление качеством; управление изменениями.
2. Администрации необходимо определить и зафиксировать время (продолжительность) отказа системы; время создания группы разработчиков.
3. Руководители и специалисты на каждом этапе должны обеспечить документацию по разработке программного обеспечения; пересмотре или исправлении модулей; по обучению персонала;
4. В задачу тестировщиков входит планирование; создание руководств по тестированию; создание автоматических тестов; прогон ручного теста; прогон автоматического теста; отчет о выявленных дефектах.

Сроки выпуска продукта, создания альфа- и бета-версии, окончательной версии должны также обязательно фиксироваться в исторических записях.

Подобные исторические записи позволяют на порядок повысить точность планирования и предотвратить необходимость гадания. По данным К. Джонса за 1994 год, такие измерения повышают качество на 40%, а производительность – на 15%. Исследование К.Джонса показало, что компании, которые практикуют измерения, являются лидерами в своей отрасли.

Стоимость измерений составляет всего 4–5% от стоимости проекта. Остается только сожалеть, что ведение исторических записей, систематических измерений и оценок на их основе, пока еще не стало привычкой у наших программистов и менеджеров.

Разработка, ориентированная на заказчика

Начнем с рассказа о разработчиках и заказчиках.

Группа разработчиков программного обеспечения и их заказчиков поехала на выставку на дизель-поезде. Все заказчики купили по билету, а разработчики купили один билет на всех. Заказчики решили, что разработчики им попались какие-то глуповатые.

По дороге один из разработчиков сказал: «Кондуктор идет!» – и все разработчики спрятались в туалет. Кондуктор прошел, говоря всем «Билеты, пожалуйста», и отобрал их у каждого заказчика. Затем он прошел к туалету, постучал в дверь и сказал: «Билет, пожалуйста!» Разработчики подсунули билет под дверь. Кондуктор взял его, а разработчики вышли из туалета несколькими минутами спустя. Теперь заказчики почувствовали себя в дураках.

На обратном пути с выставки группа заказчиков решила быть умнее и купила один билет на всех. На этот раз разработчики не купили ни одного билета, и заказчики с недоумением косились на них. Через некоторое время впередсмотрящий разработчик сказал: «Кондуктор идет!» Все разработчики

нырнули в один туалет, а все заказчики – в другой. Затем, еще до прихода кондуктора, один из разработчиков вышел из туалета, постучал в дверь туалета, в котором прятались заказчики, и сказал: «Билет, пожалуйста».

Мораль: не все решения, найденные разработчиками, хороши для заказчиков, и рано или поздно заказчики это поймут.

Распознайте заказчика, определите заказчика правильно! Заказчик – это не тот, кто будет пользоваться программой, а тот, от кого зависит ее приемка и оплата. Это может быть группа маркетинга, начальник отдела информационных технологий, хотя иногда заказчиком может быть и конечный пользователь.

Фактор успеха № 1 – это участие в проекте конечных пользователей. В обзоре Standish Group на опыте 8000 проектов приведены три основные причины провала проектов (т.е. нарушение графика, бюджета или функциональности):

1. Недостаточное участие пользователей.
2. Неполная постановка задачи.
3. Изменение требований в процессе разработки.

Мотивация труда программистов и их менеджеров

Разные люди мотивируются по-разному. Разработчики не всегда мотивируются теми же факторами, что и их менеджеры. Табл. 10. показывает относительную ценность факторов мотивации для программистов, менеджеров и «общей популяции».

Итак, по сравнению с общей популяцией разработчик программного обеспечения мотивируется возможностью личного роста (achievement), возможностью профессионального совершенствования, а также межличностными отношениями с равными себе по службе. Программист слабо мотивируется служебным статусом и отношениями с подчиненными. Характерно, что менеджеры проектов в первую очередь мотивируются ответственностью и признанием, в то время как программистов и первое, и

второе заботит мало. Если Вы менеджер и пытаетесь мотивировать разработчиков теми же факторами, которые мотивируют Вас, – Вас постигнет неудача.

Чтобы мотивировать разработчиков возможностью личного роста, необходимо обеспечить среду, позволяющую им сосредоточиться на разработке ПО. Программисты *любят* работать. Обычно они делают именно то, что их просят. Пусть программисты сами составляют графики – все равно они занижены. Пусть работают для достижения своих собственных целей, а не чужих. Цели должны быть поставлены четко, и меняться лишь настолько, чтобы оставаться достижимыми. Но целей должно быть не более одной, в крайнем случае, две.

Необходимо удовлетворять потребности разработчиков в профессиональном росте. Давайте им возможность знакомиться с техническими новинками, оплачивайте курсы по повышению квалификации, предоставляйте свободное время для учебы, оплачивайте литературу по специальности, давайте проекты, связанные с приобретением профессиональных навыков, прикрепляйте к новичкам опытных программистов. Добейтесь того, чтобы технические сложности и проблемы проекта разработчики воспринимали как личный вызов себе.

Таблица 10

Сравнение мотивирующих факторов для программистов, менеджеров и «общей популяции»

N	Программисты, постановщики задач	Менеджеры программистов	Общая популяция
1.	Личный рост	Ответственность	Личный рост
2.	Профессиональный рост	Личный рост	Признание
3.	Работа как таковая	Работа как таковая	Работа как таковая
4.	Личная жизнь	Признание	Ответственность

5.	Техническое совершенствование	Возможность роста	Карьера
6.	Карьера	Отношения с подчиненными	Зарплата
7.	Отношения с равными	Отношения с равными	Возможность роста
8.	Признание	Карьера	Отношения с подчиненными
9.	Зарплата	Зарплата	Статус
10.	Ответственность	Отношения с руководством	Отношения с руководством
11.	Отношения с руководством	Политика и управление компанией	Отношения с равными
12.	Безопасность труда	Безопасность труда	Техническое совершенствование
13.	Отношения с подчиненными	Техническое совершенствование	Политика и управление компанией
14.	Политика и управление компанией	Статус	Условия труда
15.	Условия труда	Личная жизнь	Личная жизнь
16.	Статус	Условия труда	Безопасность труда

Нельзя «давить» графиком, иначе программист будет думать, что высший приоритет для начальства – это выпуск продукта независимо от затрат. Необходимо запускать каждый проект как эксперимент, чтобы использовать эффект Хоторна.

Команда может иметь только одну из целей: решение конкретных задач (например, сопровождение), творчество (разработка нового ПО) либо техничность (upgrade). Соответственно должны подбираться и люди. Чем более творческая задача поставлена, тем более творческой должна быть команда.

Гигиенические факторы для программистов

Данные исследования, проведенного в программной индустрии, говорят о том, что 70% офисов переполнено; среднее время между событиями, отвлекающими разработчиков от работы, составляет 11 минут. В то же время погружение программиста в задачу занимает 15 минут, поэтому важно предупреждать нежелательные отрывы от работы.

Для плодотворной работы необходимо предусмотреть достаточное освещение и тепло, свежий воздух, наличие свободного места на столе и полках. На рабочем месте должно быть достаточно тихо, для того чтобы можно было сконцентрироваться. Нужно обеспечить приватность, т.е. так организовать рабочие места, чтобы никто не глядел другому в спину или в лицо.

С. МакКоннелл утверждает, что необходимо выделить 8 кв.м. на каждого разработчика и еще 1,5–4 кв.м. для книг, полок и компьютерного оборудования, обосновывая это незначительной стоимостью площадей в США по сравнению с зарплатой программистов. К сожалению, эти нормы хороши для Запада, но в отечественных условиях из-за наших финансовых реалий их обычно соблюсти невозможно. У нас часто бывает, что стоимость владения производственными площадями в расчете на одного работника сопоставима с его зарплатой. Скорее, нормы МакКоннелла – это некий идеал, к которому нужно стремиться. В наших условиях компромиссом является 5–6 кв.м на человека, минимум – 4 кв.м. Идеальный случай – это индивидуальные офисы для каждого сотрудника, как это сделано в Microsoft. Здания в Редмонде не случайно имеют форму многоугольной звезды: каждый луч представляет собой длинный коридор с маленькими комнатами по обеим сторонам. Может быть, единственным недостатком такой планировки является дефицит личного общения, что может снизить производительность рабочих групп.

Рабочее место не должно ограничивать движения ног. Хорошо, если с рабочего места можно увидеть небо, хотя бы через ряд окон. Необходим легкий доступ к принтеру, ксероксу, факсу, канцтоварам, неограниченный доступ к компьютеру. Необходимо современное компьютерное

оборудование, обеспечение немедленного или почти немедленного ремонта сломанного оборудования.

Для эффективной работы требуются современные телекоммуникации – электронная почта, индивидуальные телефоны с автоответчиками, доступ к комнатам для совещаний. Необходимое (желательно лицензионное) программное и аппаратное обеспечение (например, среда разработки или принтер, необходимый для отладки). Электронные книги и документация должны быть установлены в первую очередь, включая хотя бы минимальное обучение новинкам. О том, как это организовано в Microsoft, пишет сам Билл Гейтс [9]: «Предположим некая соискательница – назовем ее Шарон Холлоуэй – принимает наше предложение о поступлении на работу. Прежде чем Шарон прибудет в Microsoft, помощник по административной работе из ее будущей группы заполняет в корпоративной интрасети электронную форму регистрации нового сотрудника, чтобы заказать учетные записи голосовой и электронной почты, офисную мебель и компьютер с заранее установленными программами – все это должно быть готово к тому моменту, когда Шарон впервые выйдет на работу. Заполнение этой же формы гарантирует, что Шарон внесут в список телефонных номеров компании, на двери ее кабинета появится табличка с ее именем, а в канцелярии для нее будет установлен почтовый ящик. Введенные в форму данные направляются непосредственно группам, отвечающим за подобные вопросы; электронная регистрация обеспечивает, что ничего не будет упущено.

После вводной беседы, на которой менеджер отдела кадров рассказывает об общем подходе компании к производственным и кадровым вопросам, Шарон и другие новые сотрудники знакомятся с внутренним сайтом компании, с помощью которого они будут решать большую часть своих административных проблем. Шарон входит в онлайн-режим, чтобы просмотреть руководство для сотрудников (оно больше не выпускается на бумаге), скопировать программы, которые ей потребуются сверх стандартной конфигурации, и заполнить свою электронную налоговую форму.

Затем Шарон обращается к MS Market, интрасетевому инструменту службы снабжения, где она может заказать себе канцелярские товары, книги, доску для презентаций и визитные карточки. MS Market автоматически вносит в бланк заказа ее фамилию и имя, адрес электронной почты, фамилию вышестоящего менеджера, санкционирующего ее заказы, и другую стандартную информацию. Шарон нужно заполнить только несколько выделенных полей с информацией, относящейся именно к этому заказу. Исполнители получают заказ в электронном виде и доставляют все требуемое в ее кабинет. Если сумма заказа превышает некий оговоренный лимит, заказ должен быть утвержден вышестоящим руководством. Электронная система в этом случае автоматически направляет форму соответствующим менеджерам для визирования – также электронного.

Шарон посещает сайты архивов и библиотеки Microsoft, а также электронного бюллетеня компании, с помощью которых она будет изучать материалы фирмы. Подписавшись на одну или несколько услуг службы новостей нашей библиотеки, Шарон будет получать последние известия о событиях в компании и во всей отрасли в электронных версиях таких изданий, как The Wall Street Journal, The New York Times, CNet. Онлайновая библиотека ведет каталоги книг, программного обеспечения и видеоматериалов, доставку которых в свои кабинеты сотрудники могут заказывать по Сети. Кроме того, библиотекари ведут веб-страницы с новостями и материалами исследований по каждой группе продуктов Microsoft».

Проследите, чтобы члены одной группы работали на одном этаже. И последняя очень важная деталь – сделайте график рабочего времени разработчиков настолько свободным, насколько это позволяют обстоятельства.

Управление функциональностью

Тактика Microsoft в управлении функциональностью заключается в следующем.

Первоначально разрабатывается концепция, или, как ее называют в Microsoft, vision statement, представляющая собой текст объемом до 10 страниц. В дальнейшем никакие изменения в нее больше не вносятся. Концепция служит отправной точкой, путеводной звездой, определяющей, *что* в системе должно быть, а что не должно. Концепция нужна для того, чтобы все заинтересованные стороны – разработчики, пользователи, заказчики, руководство – получили адекватное представление о продукте. Концепция нужна и для того, чтобы определять, что именно впоследствии придется урезать для соблюдения графика или бюджета проекта. Крис Петерс из Microsoft говорит: «Самое тяжелое – определить, что делать не нужно. В каждом релизе мы убираем 2/3 намеченных возможностей».

Остальное отдается на откуп программистам, особенно такие вещи, как размеры кнопок, полей ввода и т.п. Слишком подробная спецификация не нужна, ибо малейшее изменение задачи делает ее бессмысленной.

Следующим шагом является создание прототипа. Прототипы пользовательского интерфейса выполняются на MS PowerPoint или MS Visual Basic, ибо пользователь никогда не знает, что ему надо, пока не увидит прототип. Правильно нарисованный экран стоит тысячи слов. Можно использовать и бумажные картинки с динамикой, особенно при работе с конечными пользователями. После того как разработан работающий прототип системы, внешний вид системы изменяется уже незначительно. Теперь можно разрабатывать руководство пользователя, или online help, которые и могут служить постановкой задачи (спецификациями).

Девиз – это отличный способ выделить главное. Например, при создании MS Excel 3.0 девиз был следующим: «сделать ПО сексапильным». В ходе разработки трехмерные таблицы отвергли из-за их «несексапильности», а графические возможности пакета оставили. История показала, что это решение оказалось верным.

Преимущества спецификаций, учитывающих только основные требования, заключаются в улучшении мотивации и морального климата среди разработчиков; повышении эффективности; уменьшении пустых усилий по разработке того, что потом выбросится; сокращении сроков

разработки. Возможные недостатки таких спецификаций проявляются в том, что можно упустить принципиальные требования, поставить неясные и невозможные цели. Возможно также чрезмерное увлечение неограниченным спецификацией процессом разработки. Разработчики часто не понимают, что реализация какой-то возможности может занять массу времени, и в Microsoft наибольшие проблемы с графиком возникают тогда, когда программисты заняты добавлением новых возможностей.

Минимальная спецификация успешна, когда требования могут изменяться в процессе разработки, в противном случае лучше делать подробные спецификации. Она должна быть сделана с привлечением ключевых пользователей с использованием преимущественно графической формы представления спецификаций.

Создайте подробный интерфейс и позвольте программистам проявить гибкость до тех пор, пока они не создали что-то похожее на прототип. Не разрешайте им развивать функциональность, пока нет прототипа. Определите точку в проекте, когда в программе перестают появляться заметные на глаз изменения. В этот момент передавайте систему на документирование и тестирование.

По статистике, каждый месяц требования изменяются минимум на 1%, поэтому необходимо как можно раньше иметь работающий прототип. Иногда нужно иметь мудрость отказываться от изменений. Разрешайте только те из них, которые позволяют улучшить продукт в имеющиеся сроки, остальные изменения отвергайте. Не забывайте объяснять заказчику цену изменений. Полезно создать комиссию по изменениям, состоящую из всех заинтересованных сторон.

Когда приходит время урезать функциональность, то применяются следующие правила:

1. Уберите все, что не является абсолютно необходимым.
2. Упрощайте все, что более сложно, чем требуется.
3. Заменяйте дорогие возможности дешевыми.

4. Если из 100% Вы оставите 70%, то и проект будет на 30% короче. Но если потом добавить вычеркнутые 30%, то в результате проект будет стоить более 100%.

Средства повышения производительности

Средства повышения производительности не являются панацеей от всех проблем при разработке программного обеспечения. Практика показывает, что все, кто декларирует выигрыш от внедрения RAD больше 25%, занимаются самообманом. 75% всей рекламы средств повышения производительности не соответствует действительности, особенно реклама CASE-технологий. Как указывает Ф.Брукс, последняя революция средств разработки была при переходе от языка ассемблера к языкам 3-го поколения. Остальные средства могут снизить затраты и повысить производительность, но не уменьшают сроков.

CASE-технологии не улучшают качество разработки! Общее заблуждение на этот счет чрезвычайно стойко. Многие забывают, что CASE – это средство для рисования диаграмм и нужно для того же, для чего текстовый редактор необходим писателю.

RAD (средства быстрой разработки – Delphi, PowerBuilder и пр.) – еще один миф. RAD ориентированы на системы баз данных, вне их эти замечательные инструменты ничего не дают.

Автоматическое программирование – миф. Если даже *конечные* пользователи не знают, что им нужно, то что можно ожидать от средств программирования? Настоящее автоматическое программирование должно работать на уровне конечного пользователя-профессионала. Примером могут служить всевозможные надстройки над электронными таблицами, финансовые калькуляторы и т.п.

Объектно-ориентированное программирование (ООП). Естественность и простота ООП опровергнуты практикой. ООП слишком сложно в изучении для того, чтобы его *правильно* использовать. ООП аккумулирует передовые идеи за последние 35 лет и предназначено для асов в программировании. Повторное использование объектов – это миф, ибо к

моменту возможности повторного применения реализация объектов устаревает. Так что и ООП не является панацеей.

Языки 4-го поколения низкоэффективны, поэтому их необходимо использовать для прикладного ПО и редко – для системного и коробочного ПО. Замена языка программирования 3-го поколения на скриптовый язык или 4GL (напр., Visual C++ на Visual Basic) ускоряет процесс собственно кодирования небольших программ вчетверо, однако остальные этапы ускоряются куда менее значительно или не ускоряются вовсе (например, проектирование). Чем больше проект, тем меньший эффект дает применение языков 4-го поколения. С ростом проекта ограничения 4GL нарастают, а эффективность падает.

Повторное использование. Так или иначе, но делать повторную постановку задачи и повторное проектирование все равно приходится. Даже если можно использовать старый проект на 80%, то участие в проекте новых программистов часто сводит эффект повторного использования к нулю. Можно покупать готовые библиотеки, однако нужно помнить, что повторное использование окупается не ранее чем через 2 года. ООП не имеет преимуществ при повторном использовании и не является ни необходимым, ни достаточным для него.

Не беспокойтесь о том, что никто не будет пользоваться стандартными библиотеками: 70% программистов предпочитают использовать готовые модули. Но не торопитесь делать библиотеки стандартными, т.к. стоимость разработки увеличивается в 3 раза. Чтобы узнать, есть ли необходимость делать библиотеку стандартной, используется «правило двух»: модуль делается стандартным, если он будет использоваться более чем в 2 местах.

Выбор средств. При выборе средства разработки необходимо обратить внимание на стабильность поставщика, качество и зрелость продукта, время освоения, применимость к проекту, совместимость со средой и перспективы развития. Однако, в первую очередь, конечно, нужно определить способность средства решить Ваши проблемы. Не обращайтесь на достоинства, которые Вам не нужны, а также на красоту продукта.

Как «вытянуть» безнадежный проект

Признаки провала проекта следующие: никто не знает, когда проект закончится, и не пытается узнать. У продукта имеются явные дефекты, хотя люди работают по 60 часов в неделю. Менеджмент не контролирует ситуацию, а заказчик перестал верить, что программа когда-нибудь появится. Команда занимает оборонительную позицию и пугается, когда кто-то извне считает, что не все в порядке. Ухудшаются отношения между всеми сторонами, активно рассматривается возможность закрытия проекта. Дух команды в упадке, настроение подавленное, работает без удовольствия.

Обычная первопричина провала проекта – отсутствие контроля.

Рекомендации руководителю следующие:

1. Срезать часть возможностей.
2. Увеличить производительность.
3. Сдвинуть график.

Думайте не о том, как быстрее закончить проект, а о том, как вообще его закончить. Соберите команду и спросите: «Что делать?» Пусть каждый член команды выдвинет не менее 5 идей. Оцените и реализуйте все, что можно. Если люди не хотят перемен – закрывайте проект, ибо нельзя исправить работу продолжением прежней тактики. Подготовьте всех к серьезным переменам и будьте реалистом.

Восстановите дух коллектива. Лучший способ для этого – предпринять символическую акцию, показывающую людям, что Вы – на стороне разработчиков. Для этого надо сокрушить какую-либо из «священных коров» организации – пусть программисты, например, нарушат график, недорабатывают, приходят на работу в любой одежде, которая им нравится, работают на дому. Купите мониторы с большим экраном, снабжайте разработчиков едой, заставьте их не работать хотя бы одни выходные. Одним словом, дайте им почувствовать свою важность.

Устраните причину провала. Отследите интересы начальства – возможно, именно его действия и привели к краху проекта. Уберите из

коллектива проблемных людей (обычно их немного), не позволяйте им разлагать дух всего коллектива. Если проблемы вызвал технический или административный руководитель проекта, который не понимает, что ему нужно уйти в отставку, то постарайтесь устроить так, чтобы над ним сменилось руководство, от которого это зависит.

Лучше вообще не добавлять людей в проект (см. Первый закон Брукса). Сосредоточьтесь на качестве. Исправьте проваленные части проекта, перепишите плохие части программы. Добавляйте код только к здоровой части программы.

Создайте механизм отслеживания (по возможности, наглядный), позволяющий определить минимальный прогресс в завершении проекта. Контролируйте результаты каждые 1–2 дня, не больше. График работ спроектируйте таким образом, чтобы каждый шаг можно было выполнить за 12 рабочих часов. Оценку делайте по принципу «да–нет», то есть даже если шаг выполнен на 99%, необходимо считать, что он не выполнен. Необходимо быть жестким в вопросе оценки. Когда люди увидят прогресс, их дух поднимется. Ведь как только сделан последний шаг – сделан весь проект!

Через 1–2 недели после начала работ перепланируйте график, растянув его в соответствии с реальной производительностью рабочей группы, и не пытайтесь нагнать его.

Испытанные приемы быстрой разработки

Ежедневный выпуск прототипа

Наиболее популярный прием быстрой разработки – ежедневный выпуск прототипа. Как известно, ежедневный выпуск очередного релиза программы (build) практикует Microsoft. Наверное, все видели надпись на экране в начале процесса загрузки операционных систем, например: Microsoft Windows NT 4.0 Build 1381. Это означает, что фактически NT4 – это 1381-я версия операционной системы! Если разделить на число рабочих дней в году (250), то выйдет 5,5 лет по состоянию на 1997 год, т.е. разработка NT началась не позднее 1991 года. Ежедневный выпуск релиза снижает

стоимость, риск проекта системной интеграции и риск низкого качества. Он облегчает поиск дефектов, помогает отслеживать график, улучшает дух команды и отношения с заказчиком.

Наращивайте продукт ежедневно, включая сборку всех компонентов. При этом нужно следить, чтобы все компоненты проходили компиляцию, чтобы продукт не содержал серьезных ошибок, препятствующих запуску программы и работоспособности в основных режимах. В малых проектах этим занимается человек, ответственный за качество. В больших проектах сборкой занимается специальная группа. В Windows NT 3.0, например, этим занималось 4 человека.

Добавлять к основному проекту есть смысл только готовые части. Каждый программист добавляет изменения обычно раз в несколько дней. Перед включением он должен проводить отладку на своей собственной копии. Нет оправдания тому, чтобы не делать этого!

В Microsoft используются символические наказания для тех, кто испортил работающую программу, вплоть до обязанности носить бейсбольную кепку с надписью «Я люблю Sun» и тому подобных обидных детских шуточек. Но главное наказание в Microsoft – это обязанность ежедневной (точнее, еженощной) сборки и тестирования, длящаяся до тех пор, пока не провинится кто-нибудь другой. При завершении крупных проектов программистов обязывают носить пейджеры или сотовые телефоны. В случае порчи работающей программы программист обязан прийти на фирму в любое время дня и ночи и исправить ее. Поэтому компилировать программу лучше ночью, а тесты проводить утром, на свежую голову. Кроме того, при обнаружении ночных ошибок компиляции программист исправит их еще до прихода своих коллег.

Проводить эту процедуру надо даже тогда, когда нет времени – ибо на самом деле это экономит время. Windows NT 3.5 состояла из 5,5 млн. строк и 40000 модулей. Время компиляции на нескольких компьютерах занимало 19 часов. Однако именно ежедневными проверками система во многом обязана своим успехом. В меньших проектах нет оправдания тому, чтобы этого не делать, особенно при «вытягивании» безнадежного проекта.

Эволюционное прототипирование

Согласно этому методу, создание программы начинается с интерфейса, на который затем наращивается код. Лучше всего начать с рискованной идеи, которую, возможно, придется отбросить и начать сначала. Необходимо строить только «фасад», но не содержимое. В процесс должны вовлекаться пользователи, которые и являются критерием правильности «фасада».

Для экономии времени в компании Microsoft часто разрабатывают прототипы на Visual Basic или вовсе в PowerPoint и лишь в случае их пригодности прототипы переписываются на нужном средстве разработки. В случае успеха у пользователей прототип развивается до приемлемого состояния, завершается и выпускается. Для эволюционного прототипирования требуются опытные разработчики.

Этот прием имеет низкий риск. Лишь 6 проектов из 39 провалились из-за потери контроля над проектом, слабой помощи заказчика и пользователей, нереалистического графика и бюджета. Так, в одном из проектов руководство потребовало завершить двухлетний проект за 6 недель!

Самопожертвование

Дайте видение (vision statement), поставьте общую цель типа «сделать лучшую в мире программу того-то и для того-то». Разрешите разработчикам выбор в методах и средствах и свободу на уровне команды. Не указывайте команде, как решать задачу, и предоставьте ей полную свободу. Этот уже упоминавшийся прием плох тем, что люди «сгорают» на работе. Но именно так обычно работает Microsoft.

Прототипирование с выбрасыванием

В этом случае не следует бояться выбрасывать прототип, хотя в него и вложены деньги. Необходимо планировать выброс, считать данный этап научным экспериментом, в ходе которого прототипируются наименее понятные места.

Разработка в срок

Как разработать программу в срок? Лучше всего работает эволюционное прототипирование, которое выполняется до тех пор, пока хватает времени. Незадолго до конца разработки подводится черта, из технических требований исключаются пункты, которые невозможно реализовать в заданный срок. После этого дальнейшее наращивание программы прекращается и начинается сборка и тестирование. Таким образом, происходит своеобразное урезание постановки. Однако если нет обратной связи с пользователем, не используйте эту схему.

Внешняя закупка ПО

Закупка ПО, или подряд, или аутсорсинг – это очень важный процесс, определяющий успех предприятия. Не секрет, что большинство преуспевающих отечественных компаний по разработке ПО занимаются подрядным программированием, называемым также аутсорсингом, или – еще более модно – оффшорным программированием.

Наиболее общее понятие – подряд или найм. В мире используются следующие типовые виды подряда.

Консалтинг – это временное (от нескольких месяцев до нескольких лет) одалживание наемной рабочей силы у специализированных консалтинговых компаний. Тысячи небольших консалтинговых компаний США и Европы являются своеобразными поставщиками кадров, кадровыми депо. Группы одалживаемых сотрудников работают на объектах различных заказчиков, однако числятся в консалтинговой компании и получают в ней заработную плату. Заказчик же расплачивается с консалтинговой компанией по договору на основе почасовой ставки за каждый отработанный час. Условия работы специалистов каждый раз зависят от условий договора консалтинговой компании с очередным заказчиком. Когда нет работы, нет и зарплаты.

По окончании проекта консалтинговая компания подбирает сотруднику новую работу. Часто случается и так, что работник сам находит себе работу и лишь оформляет ее через консалтинговую компанию, чтобы не заниматься

предпринимательской деятельностью самому. Обычно консалтинговая компания оставляет себе лишь небольшой процент (обычно 2–3%, хотя в «русских» консалтингах случается всякое) за услуги и бухгалтерское обслуживание, выплачивая остальное работнику. Характерно, что никаких социальных пакетов, страховок, отпусков, больничных в консалтинговой компании не предусмотрено. Работник сам уплачивает страховку и совершает прочие выплаты из своего дохода. Таким образом, работа в консалтинге представляет собой нечто среднее между наемным трудом и предпринимательством. Следует отметить, что консалтинг обычно не имеет никакого отношения к консультированию.

Очевидное преимущество консалтинга – экономия. Нет необходимости держать человека, когда нет работы, что удобно для краткосрочных проектов. К тому же часто сотрудники, работающие в консалтинге, обладают более высокой квалификацией, обладают знаниями либо ноу-хау, нужными для выполнения конкретного проекта. Отрицательной стороной консалтинга является длительное присутствие на своей территории сотрудников другой организации с собственной корпоративной культурой. Часто «чужие» работники не разделяют ценностей окружающего коллектива и организации в целом. Кроме того, их присутствие разлагает дух собственных работников, которые, как правило, получают более низкую зарплату, чем консалтинг. «Они зарабатывают вдвое больше нас, вот пусть и работают», – говорят «свои» работники, забывая о собственных социальных пакетах и социальной защищенности. По причине отсутствия традиций консалтинг пока не получил распространения в нашей стране. Однако мы упоминаем здесь об этом потому, что, во-первых, многим предстоит работать в консалтинге или сталкиваться с ним. А во-вторых, консалтинг в будущем может прижиться и у нас как перспективный бизнес.

Аутсорсинг – это найм другой компании с целью делегирования (передачи) ей ряда функций своей компании. Концепцию аутсорсинга придумала американская компания EDS в 60-х годах. Делегировать можно все, что угодно – техническое обслуживание (компьютеров, сетей, системы водоснабжения и канализации, Web-хостинг), ремонт помещения «под ключ»

при помощи бригады «шабашников», рекламное обслуживание (найм рекламных агентств) и т.д.

Последний писк моды – аутсорсинг поддержки пользователей услуг. Скажем, «горячая линия» по обслуживанию клиентов банка или по поддержке программы передается в Индию, где английский язык – государственный, и его хорошо знают. Клиент, находясь в Англии и звоня по местному телефону, даже не знает, что его обслуживают в Индии. Известен случай, когда индийская фирма установила в центре Лондона видеокамеры для того, чтобы инженер по сопровождению мог сказать пользователю: «Какая сегодня ужасная погода, сэр!» или: «Как Вы добрались на работу, мадам, ведь сегодня такие жуткие пробки на Трафальгарской площади!»

Главный признак аутсорсинга, отличающий его от обыкновенного подряда – делегирование управленческих функций, а не просто объемов работ. Обычно устанавливается фиксированная цена за отчетный период (скажем, ежемесячная оплата) либо цены определяются фактическим объемом выполненных работ.

С каждым годом аутсорсинг получает все большее распространение в мире. Компании хотят делегировать все не свойственные им функции и сосредоточиться только на тех, которые составляют основной бизнес компании. Билл Гейтс пишет: «Один из важных принципов реинжиниринга состоит в том, что компании должны сосредоточить основное внимание на том, что составляет суть их деятельности, и вывести остальные функции за организационные рамки своего предприятия.

В компании Microsoft главными задачами являются создание крупномасштабных программных продуктов, сотрудничество с другими компаниями-производителями ПО, а также обслуживание и поддержка клиентов. Мы вывели за рамки компании целый ряд функций, не подпадающих под эти три категории, от консультативной технической помощи для наших служащих до физического изготовления наших пакетов программ. Такая политика помогла избежать быстрого разрастания штатов и привела к сокращению административных расходов (хотя это не означает, что мы совсем остановили рост численности персонала).

Например, мы прибегнули к услугам сторонней фирмы, скажем, для установки и обслуживания настольных компьютеров, потому что компании, специализирующиеся на этой работе, обычно используют самые передовые технологии, к тому же у нас появляется возможность выбора наиболее выгодного предложения среди борющихся за наши заказы подрядчиков».

Оффшорное программирование – это перемещение разработки программного обеспечения в регионы с более низким уровнем издержек. Оффшорное программирование представляет собой разновидность аутсорсинга. Главная цель оффшорного программирования заключается в экономии на затратах. Обычно удается сэкономить в 2–5 раз, хотя существует и более скромная оценка – 35%. Второстепенной целью является решение кадровых проблем.

Не следует путать оффшорное программирование с международными схемами ухода от налогообложения. Оффшорное программирование означает только то, о чем говорилось выше: заказчик программного обеспечения находится в богатой стране, а разработчик – в бедной.

Договор подряда

Цена договора может основываться на плановых либо фактических затратах времени и материальных затратах, с полной или частичной предоплатой или без нее (оплата по факту).

Как рассчитать цену? Для этого широко используется почасовая ставка. В целом по отрасли разработки программного обеспечения в Беларуси накладные расходы составляют порядка 250%. Это означает, что для обеспечения программисту заработной платы в 200–300 долл. «чистыми» на него должно приходиться хотя бы 900–1000 долл. безналичных денежных поступлений.

Не слушайте специалистов с высшим экономическим образованием о якобы чрезмерно большой величине накладных расходов в 250%: мол, таких высоких процентов не может быть. Спросите у них, каковы, по их мнению, должны быть расходы по разовым заказам. Ведь накладные расходы наших

государственных предприятий по изготовлению «твердых» изделий малыми партиями по разовым заказам нередко составляют 2000–2500%.

Как это объяснить? Просто экономисты привыкли иметь дело с другими (обычно материалоемкими) отраслями типового производства, а также с предприятиями производственного типа, в которых накладные расходы редко превышают 40–60% из-за высокой степени стандартизации и типизации производства, а также из-за высокой стоимости материалов в цене изделий. Проектный характер труда сам по себе предполагает высокий уровень накладных расходов.

Согласно классификации Международной организации труда, программирование – это не производство, а оказание услуг. В процессе программирования материалы практически не используются – только дискеты, матрицы компакт-дисков, бумага, порошок для ксерокса, услуги связи, Интернет и т.п. Вся компьютерная техника, офисное оборудование, мебель относятся не к материалам, а к основным средствам и автоматически попадают в накладные расходы.

Для оценки накладных расходов в Вашей компании необходимо вести статистический учет затрат. Скажем, можно фактические затраты за 3 последних месяца разделить на 3, на численность производственного персонала (программистов, постановщиков, дизайнеров и т.д.), на число рабочих дней в месяце (20) и на число часов в рабочем дне (8). Это и будет Ваша нижняя планка. Для получения прибыли необходимо заложить в контракт большую ставку, как минимум на 30–40% и более, с учетом возможных затяжек, простоев, больничных и т.д.

Типичные почасовые ставки белорусских компаний, работающих для зарубежных заказчиков, составляют от 10 до 20 долл. в час. Ставки московских компаний – 25–35 долл., американских – 70–200 и более долл. в час. Чтобы получить цену контракта, необходимо умножить плановую трудоемкость, рассчитанную одним из указанных выше методов, на расценку и не забыть прибавить материальные затраты, если они есть (командировочные расходы, оборудование, визы, страховки и т.д.).

Необходимо следить, чтобы в расчетах использовались одинаковые единицы измерения.

При фиксированной цене договора (fixed-cost contract) цена получается в среднем на 50% выше фактических затрат, однако гарантируется достижение нужного результата. При оплате фактических затрат платится за процесс, но нет гарантии достижения результата в рамках бюджета.

Существует еще один способ – выполнить предоплату в 20% и ждать первых результатов; если понравится, то можно платить дальше и т.д. При этом Вы все равно вынуждены участвовать в проекте. Но это даже лучше – ведь в любом случае Вам пришлось бы писать постановку задачи.

Если заказчик настаивает на фиксированной цене договора при недостаточно определенных требованиях, то максимальную цену определит наиболее компетентный постановщик, а минимальную – неопытный, который не знает рисков, связанных с неполными требованиями. Поэтому существует стандартная рекомендация по определению победителя тендера: при прочих равных обстоятельствах следует выбирать поставщика, предложившего максимальную цену. Однако на практике в силу боязни членов тендерной комиссии быть обвиненными в пренебрежении к необходимости экономии средств или быть заподозренными в лоббировании так поступают намного реже, чем следовало бы. Во всяком случае, никогда не выбирайте поставщика, предлагающего минимальную цену! Он-то и окажется самым некомпетентным.

Недостаточно определенный проект должен состоять из двух этапов, где первый, более короткий, отводится под определение требований ко второму этапу и предусматривает постановку, определение и стабилизацию требований, а уже во время второго этапа все остальные работы выполняются на основе фиксированной цены договора, определяемой по итогам первого этапа.

Как сделать подряд успешным

Коммуникация – ключевой момент для успеха подряда. Необходимо продумать способы общения (командировки, электронная почта, телефон и

т.д.) с учетом разницы в пояском времени, качества линий связи, языковых проблем, социально-культурных различий. Командировки планируйте в начале, в конце и каждые несколько месяцев в процессе работы. В договор необходимо обязательно включить требования по проекту, документацию, библиотеки, копирайт, тесты, дизайн, исходные тексты программ. Необходимо заранее определить, кто будет сопровождать программу, каким образом будут сохранены коммерческие секреты, обеспечена конфиденциальность и т.д. Необходимо также четко определить критерии приемки и сферы ответственности каждой стороны.

Главный риск заказчика при подряде заключается в потере видимости и контроля над проектом. Поэтому необходимо предусмотреть как можно больше контрольных точек. Мысли заказчика о том, что можно выдать задание и забыть о нем до момента сдачи работы являются опасной иллюзией, сгубившей не один проект. Если заказчик не участвует в процессе разработки, то можно дать полную гарантию, что в результате получится совершенно не то, что ему требуется.

Необходимо также учитывать возможность рассекречивания. Если заказываемая разработка – основной бизнес Вашей компании, то ее конкурентоспособность ослабляется, даже если возможен кратковременный положительный эффект. Убедитесь в том, что подряд – действительно в долгосрочных интересах Вашей компании.

Обычно в договор включается раздел NDA – Non-Disclosure Agreement, договор о неразглашении. Полезно на его основе разработать нечто вроде Подписки о неразглашении, которую должны подписать все участники проекта со стороны заказчика. По нашему (а тем более по западному) законодательству о коммерческой тайне подобная расписка может служить основанием для судебного разбирательства.

Идеальный тип проектов для подряда – переписывание существующей системы. Чем лучше определены требования к проекту, тем более вероятен успех подрядной разработки.

Во всех случаях следует избегать двойных стандартов при оценке собственной и чужой работы. Замечено также, что если появляется внешняя конкуренция, то внутренние разработчики начинают работать лучше. Любопытно, что при расположении заказчика и подрядчика в разных часовых поясах работа может оказаться более эффективной за счет использования электронной почты и общего сервера.

Ключи к успеху подряда со стороны заказчика:

1. *Управляйте внешним проектом тщательнее, чем внутренним.*
2. Сделайте приоритетом *общение* с разработчиком. Выделите специалистов для ответов на вопросы разработчика.
3. Тщательно выбирайте партнера и тщательно подготовьте договор.
4. Постановка должна быть как можно более подробной, если только разработчик не сильнее заказчика в данной задаче.

Главный приоритет с точки зрения разработчика – обеспечить видимость выполнения проекта заказчиком. Для этого обычно применяются следующие приемы:

1. Ведение разработки полностью или частично in-house, т.е. непосредственно на территории заказчика, по схеме консалтинга. Это позволяет полностью контролировать процесс разработки за счет повышения затрат.
2. Частая оценка заказчиком очередных релизов программного продукта (не реже чем один раз в 2-3 недели, а в необходимых случаях – еще чаще, даже ежедневно), чтобы заказчик непосредственно участвовал в процессе разработки, не допуская появления неадекватного продукта.
3. Регулярная отчетность разработчика перед заказчиком о ходе работ.

Многие компании применяют систему ежедневной отчетности разработчика перед заказчиком. Каждый программист, участвующий в проекте, составляет электронное письмо с кратким (2–3 строчки) отчетом за каждый день и направляет его менеджеру проекта. Менеджер проекта сводит

отчеты воедино и направляет их заказчику ежедневно либо еженедельно по желанию последнего. Практика показала, что заказчик знает о внешних проектах больше, чем о своих собственных. Он желает получать отчеты даже тогда, когда проекты ведутся на его собственной территории. Это снимает стресс заказчика, позволяет и ему, и разработчику раньше увидеть проблемы. Появляется возможность предпринять ранние корректирующие воздействия и не допустить срывов. Программисты быстро привыкают к подобной форме отчетности, которая отнимает у них несколько минут в день.

Таким образом, проблемы, возникающие при подрядной разработке, по своей сути являются проблемами не техническими, а организационными, и даже психологическими, культурными. Они требуют от обеих сторон доброй воли, доверия и постоянного общения.

Проблемы менталитета

Главная проблема в оффшорном программировании заключается в различиях в менталитете разработчика и заказчика. Причиной этого является фундаментальное культурологическое различие между народами – в ценностях, отношении к жизни, работе, уровне и сфере доверия и т.д.

Так, обычно западные партнеры предпочитают узнавать о проблемах в момент их появления и готовы встретить плохие известия с пониманием. Люди западного мира ужасно не любят неопределенности, а также не любят, когда проблемы «зависают». Своевременное оповещение партнера о проблемах – ключ к успеху работы с западным заказчиком, который готов терпеть любые неприятные известия, лишь бы он мог предвидеть развитие событий, контролировать ситуацию и управлять рисками.

В западном мире постоянно что-то случается – происходят катастрофы, крушения, тайфуны, наводнения. Вопреки расхожему мнению наших сограждан, там все время что-нибудь ломается. В центрах крупных городов среди бела дня затеваются перестрелки, гаснет свет, затапливает метро и т.п. Причем все это происходит гораздо чаще, чем в нашей стране, использующей на порядок меньше всевозможной техники. На фоне всевозможных

катаклизмов отставание разработки от графика с точки зрения рационального Запада – это не трагедия, а, можно сказать, сущая ерунда.

Для западного человека признание собственной вины не означает «потерю лица», как на Востоке. Наоборот, западная мораль поощряет признание даже там, где оно не важно, не нужно или даже вообще не является проступком. В то же время японцу или китайцу, которым в лицо открыто высказали недовольство или, не дай Бог, обвинили в чем-либо, впору кончать жизнь самоубийством или, в крайнем случае, жестоко отомстить обидчику. Причем месть обычно строится так, чтобы обидчик ни за что не догадался об источнике своих неприятностей, а считал их «наказанием свыше». Поэтому на Востоке всегда не договаривают, давая собеседнику самому осознать свою вину. А мы занимаем среднее положение между Востоком и Западом.

К сожалению, разработчики СНГ, Индии и некоторых других стран избегают сообщать западному партнеру плохие новости, считая это потерей лица. Они часто говорят, что «начальник заболел», «уехал и не сказал, когда будет», «сейчас никого нет», «мы не знаем» или просто не отвечать на письма и телефонные звонки. Это ведет к замалчиванию проблем и негативным последствиям для проекта. Если заказчик с большим опозданием узнает о невозможности получения продукта в срок – а ведь он связан обещаниями, – то это неминуемо ведет к разрыву партнерских отношений.

Переписка

На Западе у людей обостренное чувство времени. Поскольку поощряется индивидуализм и личные достижения (achievement), время является непреходящей ценностью, ибо только его жесточайшее планирование позволяет достичь поставленных целей. Напротив, на Востоке время не ценится потому, что обычно не ставятся никакие цели. Пытаться достичь чего-то бессмысленно, ибо все равно достигнутое отберет государство или кто-нибудь более сильный. Поэтому наши люди «часов не наблюдают», часто предпочитая спокойно плыть по течению.

На Западе Вы не можете опоздать даже на пять минут. Поступить так – значит, ни много, ни мало – посягнуть на чью-то личную свободу, так как эти пять минут Ваш собеседник мог посвятить важным делам, другому человеку или даже заработать деньги. Напротив, на Востоке чем выше руководитель, тем больше по негласному ритуалу он обязан опоздать. Большие начальники могут опоздать на несколько часов, соревнуясь между собой: кто позже придет – тот и главнее.

Обратите особое внимание на формальные моменты общения с западным партнером. Ответ на любое электронное письмо или факс должны следовать в течение не более 24 часов с момента получения, а лучше всего – немедленно. Если Вы не можете ответить в течение суток, то сразу напишите в ответ, когда Вы планируете ответить, и не забудьте это сделать в запланированный срок! Не вздумайте назначить срок, в который Вы не уложитесь. Если это все же произошло, немедленно сообщите об этом партнеру и назначьте новый срок. Главное – не держите его в неведении, не заставляйте его нервничать, полагая, что он «подождет», «он забыл» или «ничего страшного, и так сойдет». Не сойдет!

Даже уезжая на один день, не забывайте оставлять автоответчик, который автоматически отвечает на все письма стандартным письмом, чем-то вроде: «Я в командировке до такого-то числа. Если смогу – отвечу раньше. По таким-то вопросам обращайтесь к такому-то человеку по такому-то адресу или телефону». Ваша подпись.

Вообще, при работе с западным партнером самая ужасная ошибка – не реагировать даже по мельчайшему поводу. Отсутствие реакции или запоздалая реакция в отсутствие каких-либо объяснений по этому поводу напрочь убивает желание сотрудничать.

Даже если письмо не требует ответа, все равно необходимо дать знать, что Вы его получили – например, напишите в ответ: «спасибо» или «большое спасибо, с наилучшими пожеланиями, такой-то». Ни одно соображение партнера не должно остаться без ответа. Игнорирование вопроса или проблемы – непростительная оплошность в общении. Вы обязаны что-то ответить, даже если Вам это неловко. Попробуйте спрогнозировать, чего от

Вас ожидают, и вспомните, *что* именно Ваш партнер отвечал Вам в подобных случаях.

Западные люди преувеличенно, даже несколько ханжески вежливы, благодарят за любую мелочь, поощряют любое достижение партнера, поздравляют с любым успехом, будь он маленьким или большим. Поэтому отечественная «комиссарская» этика делового общения, когда деловые письма составляются без личного обращения, традиционных благодарностей и знаков внимания в конце, кажется им немыслимым хамством.

Западные люди в целом более прагматичны, рационалистичны и формальны, любят законы. Люди восточной ментальности склонны к романтизму, наивности. В системе ценностей Востока личность важнее схемы, а здравый смысл важнее закона и договора. Так, между японскими и китайскими фирмами письменные договоры зачастую вообще не оформляются, сделки заключаются устно. Если возникают конфликтные ситуации или существенные изменения условий, то они улаживаются в рабочем порядке путем взаимных переговоров и уступок.

Различия выражаются и в принципах подбора кадров. Западные менеджеры склонны подбирать кадры под конкретную задачу. Бывает, что при смене производственных задач одни работники увольняются для того, чтобы нанять других с такой же профессией, но другим опытом. В то же время для наших руководителей не менее важен потенциал работника, его способность развиваться и перестраиваться в соответствии с текущими нуждами предприятия. Для сравнения: в Японии важен только потенциал кандидата и не так важны конкретные знания, которыми он обладает. Все необходимые знания приобретаются при производственном обучении, на которое постоянно затрачивается не менее 20% рабочего времени. Поэтому при смене производственной задачи производится переобучение людей, но никого не увольняют.

Западные люди нетерпимы к любому мелкому обману, в частности, плагиату. Попавшегося на этом студента, например, в тот же день исключают из университета независимо от ранее полученных оценок. По этой причине западные студенты не заимствуют готовые письменные работы, а успешные у

нас интернет-ресурсы типа бесплатных коллекций рефератов там невозможны. Напротив, любой обман партнера на Востоке считается честью и доблестью, если дело окончилось к выгоде лжеца. В целом на Востоке лжи относятся терпимо. Скажем, в Китае правда относительна и вообще не относится к категории морали. Мы не можем это осуждать. Дело в том, что в конфуцианском обществе верность и служение своему делу ценится выше честности.

Еще одно важное различие состоит в отношении к власти. На Западе и близко нет такого страха и благоговения перед органами власти, какое имеет место, например, у нас. На Западе власть обычно воспринимается как союзник, как инструмент: раз человек платит налоги – значит, может требовать от власти, чтобы она его обслуживала. Если кто-то нарушает законы, то положено жаловаться властям. Поэтому на Западе поощряется доноительство, помогающее добиться торжества справедливости. Донос на конкурента или сослуживца считается совершенно нормальным явлением, никому даже в голову не приходит это осуждать. Государство на Западе поощряет подобный тип поведения не только публичной похвалой, но и денежными премиями. Особенно премии распространены в Германии, где общепринято доносить куда следует не только на сослуживцев, но и на соседей.

В то же время у нас и на Востоке государство на протяжении столетий неизменно уничтожало (или обезвреживало) ярких, талантливых личностей с возможностями и претензиями выше среднего. Поэтому наше государство традиционно воспринимается как враг, которому стараются не попадаться на глаза и к помощи которого обращаются лишь в самых крайних случаях. Любое сотрудничество с властью, а тем более доноительство в нашем обществе обычно имеет крайне негативную оценку. Донос на конкурентов осуждается, а открытое обсуждение способов ухода от налогов не считается зазорным перед лицом общего врага – государства.

Язык служит отражением менталитета. Так, в русском языке большинство предложений имеет безличную, пассивную форму, когда действующее лицо не известно. Например, у нас говорят: «его сняли с работы», причем не указывая, кто. В то же время в английском языке

предложений без подлежащего не существует, предложения имеют преимущественно активный характер: такой-то сделал то-то.

Никогда не говорите западным партнерам о том, что Ваша компания или Вы лично уклоняетесь от налогообложения. Вряд ли Вас «сдадут» компетентным органам, но о Вас и Вашем бизнесе сложится негативное мнение. Не следует делать того же по поводу конкурирующих или партнерских фирм, ибо негативное мнение будет соотнесено со страной в целом, а значит, и с Вами.

Любопытно, что 83% заказов американских компаний на оффшорную разработку в Индии сделаны этническими индийцами, занимающими в этих компаниях высокие должности. В то же время выходцы из бывшего СССР избегают заказывать что-либо у компаний, работающих на своей исторической родине. Частично это происходит по объективным причинам, частично, по-видимому, из-за боязни показаться нелояльными и желания дистанцироваться от своей прошлой жизни.

В чем-то они правы. Отношение иностранных заказчиков к нашим людям далеко не однозначное, а, точнее сказать, такое же, как в Москве к чеченцам. Среднестатистическое американское мнение гласит, что есть отдельные «хорошие» русские, но в целом русские постоянно ловчат, и с ними нужно всегда «держаться настороженно».

В качестве примера приведем поучительную историю карьеры вице-президента одного из крупнейших банков с Уолл-Стрит. Эта женщина проработала в банке 12 лет, пройдя путь от простого программиста до вице-президента по разработке программного обеспечения. Безупречная работа приносила ей похвалы руководства, хорошее вознаграждение и акции банка. Однако она стала преградой в осуществлении планов другого вице-президента, который собрался закупить один программный продукт за 8 млн. долл. Она доказала руководству, что аналогичный продукт можно приобрести всего за 800 тыс. долл. Таким образом, сверхдорогая закупка у неизвестной фирмы из провинции была сорвана.

За это вице-президент–мужчина жестоко отомстил, написав жалобу на свою неподатливую коллегу, заявив, что та потворствует русским при приеме на работу. Для того чтобы понять ситуацию, нужно знать отношения американцев к представителям различных наций. Если бы такое написали про негра, то заявителя немедленно уволили бы и подали в суд за расовую дискриминацию. Если бы дело возбудили в отношении китайца или итальянца, то все бы возмутились махровым национализмом, а автор жалобы подвергся бы публичному остракизму вплоть до того, что с ним перестали бы здороваться. Однако в отношении русского подобное обвинение – увы! – совершенно не показалось странным и выглядело искренней заботой о благе банка.

В результате разбирательства выяснилось, что за 12 лет русский вице-президент действительно взяла на работу 11 программистов по представлению небольшой русской консалтинговой фирмы. К тому же все эти программисты были русскими. Их квалификацию проверили, все они оказались превосходными специалистами.

Тогда проверили автобиографии этих 11 программистов на глубину в 6 лет назад до поступления на работу в банк. Дело в том, что для американцев отнюдь не секрет, что многие русские при поступлении на работу представляют фальшивые характеристики с предыдущих мест работы, которые покупают у созданных специально для этой цели фиктивных фирм. Оказалось, однако, что все принятые русские специалисты действительно работали по специальности в местах, указанных в их резюме.

Проведенное расследование так и не выявило никакого компромата. Однако страх перед русскими оказался настолько велик, особенно на фоне развернувшегося тогда (осенью 1999 года) скандала о хищении в России средств МВФ через Bank of New York, что обвиняемую на всякий случай уволили по статье «недоверие» (non-trust), лишив всех опционов на акции и отсроченных премий.

Зная это, легче понять, почему так трудно отечественным компаниям находить заказы в США и Европе. Опыт показывает, что выгодное сотрудничество возможно только по личным связям, когда разработчик тесно

связан с заказчиком личным знакомством – непосредственно либо через человека-посредника, тесно связанного с обеими сторонами.

Если личных связей нет, то сотрудничество обычно не носит выгодного характера. Цены часто опускаются до уровня даже более низкого, чем в нашей стране, а невыполнение обязательств со стороны западного партнера становится обычным явлением. На Западе тоже любят легкий хлеб, тем более что при выполнении небольших случайных оффшорных проектов малыми группами программистов-надомников договоры, как правило, не оформляются. Заказчик никаких обязательств на себя не берет, действуя по принципу: «вы делаете, а мы посмотрим». Особенно этим грешат бывшие «наши», но практика показывает, что и коренные жители западных стран тоже отнюдь не промах.

ПРИЛОЖЕНИЕ 1

Разработка резюме

Подача и отбор резюме – первый этап в процедуре приема на работу. Целью резюме является предварительное знакомство с кандидатом и возможность значительной экономии времени при отборе. В противном случае пришлось бы проводить устное собеседование со всеми кандидатами.

Сегодня письменное резюме специалиста – это обязательное условие рассмотрения кандидатуры при приеме в серьезную организацию с высоким уровнем заработной платы. Большинство кандидатов на должность отсеивается уже на этапе рассмотрения резюме. Вот почему так важно правильно составить его, избегая типичных ошибок.

Содержание

Какие данные о себе указывать?

Укажите Ф.И.О., год рождения, контактные адреса (в том числе электронный) и телефоны. Часто забывают указать отчество или домашний адрес. Не указывают, какой из телефонов рабочий, а какой домашний, и в какое время куда лучше звонить. Хорошо добавить маленькую строгую цветную фотографию.

В тексте довольно часто приводят не тот электронный адрес, с которого посылают электронное письмо с резюме. Если кандидат претендует на владение компьютером – это ошибка. Посылать резюме надо только с того адреса, который фигурирует в тексте.

Образование

В разделе «Образование» укажите, где, когда и какое учебное заведение окончили, а также (если недавно закончили вуз) какие были оценки – если, конечно, они являются Вашим «плюсом». Если Вы занимаетесь

наукой или занимались ею в вузе – укажите научную специализацию, ученую степень, ученое звание.

Необходимо перечислить всевозможные награды различных конкурсов, олимпиад, конференций, в том числе школьных, наличие «красных» дипломов и т.п. Пишите свое резюме подробно, указывайте факультет, кафедру, специализацию, тему диплома. Укажите номера соответствующих дипломов, свидетельств и сертификатов. Не забудьте также о курсах повышения квалификации и стажировках, даже если они были однодневными.

Опыт работы

Это самый важный и самый длинный раздел резюме, который по объему должен превышать все остальные разделы вместе взятые. Если Ваша работа носила проектный характер (программист, дизайнер), то перечислите все Ваши проекты, длившиеся свыше одного месяца. Если работа однородная или не связана с проектами (менеджер, продавец), то выделите в ней какие-то особенности и виды работы. Например, работа с определенным крупным заказчиком или поставщиком.

По каждому проекту (виду работы) необходимо кратко и точно указать:

- период времени с точностью до месяца (можно приблизительно);
- место работы;
- наименование проекта (вида работы, особенности);
- краткую характеристику проекта (вида работы, особенности) на 1–2 строчки;
- используемый инструментарий (например, компьютерные программы, специфические технологии, которые могут представлять интерес для работодателя);
- Вашу роль в выполняемой работе.

Скажем, если Вы – программист, то в последнем пункте необходимо указать, чем именно Вы занимались, кроме программирования – например,

постановкой, разработкой структуры базы данных, документации, тестированием, внедрением, сопровождением и т.д.

Если Вы – операционист в банке, то необходимо указать, с какими клиентами работали (особенно если были крупные клиенты) и сколько было клиентов; какими счетами (рублевыми, валютными, вкладами населения) оперировали; какой программой типа «операционный день банка» пользовались и т.д.

Если Вы Web-дизайнер – не забудьте привести ссылки на выполненные Вами работы.

В Ваших интересах сделать список как можно длиннее. Чем больше опыта – тем больше шансов устроиться на работу. Не пренебрегайте упоминанием курсовых, дипломных, научных работ, всевозможных приработков и т.д. Все это – большие проекты, малые ли – так или иначе составляет Ваш реальный опыт, который непременно нужно отразить в послужном списке.

Перечислять работы (проекты) и нанимателей необходимо в обратном хронологическом порядке, т.е. последняя работа должна быть записана первой, предпоследняя – второй и т.д.

Когда раздел с описанием опыта опущен, невозможно сделать вывод о качествах кандидата, и резюме неизбежно проиграет в сравнении с другими.

Нужно ли указывать желаемый уровень оплаты труда?

Это зависит от обстоятельств. Вообще говоря, всякий разговор о зарплате до окончания собеседования обычно ставит нанимателя в трудное положение. Ведь он еще не знает Ваших возможностей, не оценил, сколько Вы «стоите» в его шкале ценностей и возможностей, потому и не может заранее ответить, готов ли он дать Вам требуемую зарплату. Поэтому, если Вы вообще решились указывать конкретные цифры, то укажите только желаемый минимальный уровень, на который Вы согласны, и обязательно упомяните, что он минимальный. Писать лучше правду. Если Вы укажете слишком высокую минимальную зарплату, то никто не рискнет с Вами связываться. Если же Вам предложат запрашиваемый минимум, то всегда

можно сказать, что указанная цифра минимальная, а предстоящая работа сложная, ответственная, и она стоит намного больше.

Нужны ли личные подробности?

Следует как можно подробнее излагать факты производственной биографии и минимум – личной. Это позволит показать, что Ваша жизнь в большей степени ориентирована на работу, нежели на досуг. Не стоит излагать подробно автобиографию, детали, связанные с Вашим детством или историю Ваших браков. Не стоит указывать наличие прав на вождение автомобиля, если Вам предстоит работать в офисе.

Если для Вас настолько важны собственный автомобиль, дача, трое детей и увлечение легкой атлетикой, что Вы решились указать это в производственном резюме, то наниматель подумает, что Вы – неделовой человек. Ведь ему придется отпускать Вас в рабочее время на техосмотр или ремонт автомобиля, пристраиваться к графику Ваших соревнований или гастролей.

В конце резюме будет вполне достаточно перечислить 2–3 своих увлечения. Лучше указывать активные увлечения (например, спорт, туризм, танцы), чем бытовые (дача, рыбалка, охота).

Хорошо ли резюме в виде единой таблицы?

Широко распространенная ошибка – оформление резюме в виде единой безликой многостраничной таблицы-анкеты, состоящей из неких оценок кандидата, которые он выставил себе сам.

Главный недостаток этого решения состоит в том, что анкеты обычно берутся в Интернете с сайта какой-нибудь конкретной организации, у которой – свои особенности и специфические требования к кандидату. Например, многие программисты используют при рассылке резюме анкету СП ИВА, не обращая внимание на то, что она вообще формой резюме не является и служит для формирования компьютерной картотеки специалистов. Большинство пунктов этой анкеты касается знаний конкретных программных продуктов компании IBM. Это необходимо для поступления в ИВА, главным

заказчиком которой является IBM. Для других программистских организаций указанные знания и умения чаще всего бесполезны, зато требуются совершенно другие, которых в этой анкете нет.

При подмене резюме таблицей невозможно оценить профессиональный и человеческий потенциал кандидата. Только свободная форма и наличие послужного списка позволяют оценить индивидуальность и практические умения и отделить их от формализованной самооценки.

Если у нанимателя есть требования к оформлению резюме, строго следуйте им. Если же на сайте есть только таблица, то приложите к ней и обычное резюме, что значительно повысит Ваши шансы быть принятыми на работу.

Оформление

В какой форме отправлять резюме?

Достойный кандидат направляет резюме по электронной почте с короткой сопроводительной запиской и вложением в виде RTF- или DOC-файла с именем в виде собственной фамилии.

Дело в том, что менеджер после прочтения обычно размещает файл с резюме у себя на компьютере, как правило, под фамилией кандидата. Поэтому не стоит посылать файл с именами RESUME.RTF или RESUME.DOC – ведь такой файл обязательно придется переименовывать.

Это только для Вас этот файл, быть может, единственное резюме, зато для менеджера – он один из десятков или сотен подобных. Лишь очень немногие кандидаты догадываются сэкономить время менеджеров и назвать файл с резюме своей собственной фамилией. Это свидетельствует об учете человеческого фактора и заранее располагает менеджера к себе.

Не следует посылать резюме в Microsoft Excel – не все им пользуются. К тому же, как уже упоминалось, таблица, в том числе электронная, не является лучшей формой представления резюме. Иногда приходится получать резюме в виде таблицы, по ширине двукратно выходящей за

пределы экрана, что делает его почти нечитаемым. Сразу возникает вопрос: о чем думал кандидат, на что он рассчитывал?

В то же время отсутствие всякого форматирования может затруднить восприятие текста, поэтому форматы Microsoft Word несколько предпочтительнее, чем, например, формат обычного текста (txt или plain-text), который редко кому кандидату удастся оформить так, чтобы он легко воспринимался и хорошо выглядел. С помощью Word этого легче добиться.

Надо ли архивировать резюме?

Во-первых, оптимальный размер файла резюме с фотографией – 100–150 Кбайт. Не делайте файл с резюме больше чем 200–250 Кбайт. Не стоит включать в резюме большие фотографии или громоздкие элементы оформления – из-за них увеличивается размер файла.

Во-вторых, его не следует архивировать, заставляя менеджера тратить драгоценное время на распаковку. Не заставляйте нанимателя ждать или делать лишние усилия! Тем более, менеджер может вообще не распаковать Ваш файл, если он пользуется другим архиватором, чем Вы.

Как выбрать шрифт?

Хорошо, когда резюме легко читается. Плохо действуют на восприятие нагромождение различных элементов оформления, разнообразие шрифтов. Пишите шрифтом Arial кеглем не менее 10, который легко воспринимается глазом и обычно установлен на любом компьютере. Не надо пользоваться слишком мелкими шрифтами, ибо менеджеры – часто люди не с самым лучшим зрением. Не пользуйтесь другими, реже встречающимися шрифтами, ибо их может не оказаться на компьютере менеджера.

Что писать в сопроводительном письме?

Напишите что-нибудь вроде: «Уважаемый директор компании! (Варианты: руководитель студии, зав. отделом программирования, начальник цеха, управляющий банком, начальник отдела кадров) Ищу работу (должность) такую-то. Резюме прилагается. С уважением, такой-то».

Следует непременно указать, на какую работу или должность Вы претендуете – из Вашего резюме это может быть непонятно. Не пишите лишние слова вроде «Вот, случайно забрел на сайт, увидел, что Вам требуются сотрудники, и решил послать свое резюме».

О грамотности

Конечно, для многих работ стопроцентной грамотности не требуется. Опыт показывает, что наличие грамматических и речевых ошибок слабо связано с деловыми и волевыми качествами кандидата – практическими навыками, работоспособностью, целеустремленностью, организаторской жилкой, способностью добиться результатов. Зато отсутствие ошибок часто свидетельствует об уровне подготовки кандидата как специалиста – о его эрудиции, теоретической подготовке, логических и алгоритмических способностях, умении четко и связно излагать свои мысли.

В любом случае, грамматические и речевые ошибки еще никогда не способствовали хорошему восприятию текста и не улучшали качества резюме. Попросите знакомых с хорошим знанием русского языка прочитать Ваше резюме, прежде чем решите рассылать его.

То же касается резюме, составленного на иностранном языке. Ни в коем случае не пользуйтесь программами автоматического перевода! Они пока работают настолько плохо, что непригодны ни для чего серьезного. Результат вызовет смех и недоразумения, а резюме сочтут халтурой или отпиской.

Надо ли подавать резюме на иностранном языке?

Направление в русскоязычную компанию резюме только на иностранном (обычно английском) языке является распространенной ошибкой. Во-первых, не все хорошо знают английский. Во-вторых, английское резюме еще не означает, что Вы знаете английский – ведь перевод вполне мог сделать кто-то другой.

Если знание языка может быть плюсом, то лучше показать это на собеседовании, а в послужном списке при описании какой-нибудь работы

указать, что был, например, опыт переписки с зарубежным партнером, либо приходилось разрабатывать документы на иностранном языке. Если такого не было, можно просто указать, что «немецкий – 7 лет в школе, 5 в вузе, итоговая оценка в вузе – 5». Это гораздо лучше, чем приводить беспомощные самооценки типа: «чтение – 4, письменный – 3, разговорный – 2», которые никому ни о чем не говорят.

Исключение составляет случай, когда резюме потенциально может читаться человеком, не знающим русского языка. Тогда резюме на английском или другом иностранном языке обязательно. Но для белорусской фирмы даже в этом случае лучше направить оба варианта – русский и иностранный, предоставив менеджеру возможность выбрать предпочтительный для него язык.

Структура

Ни в коем случае не пишите связного текста. Не превращайте резюме в литературное сочинение! С ним очень неудобно работать.

Дело в том, что менеджеры – люди занятые и обычно читают резюме, что называется, по диагонали, выхватывая наметанным глазом нужные им места и ключевые слова. Поэтому резюме должно быть структурированным – так его легче воспринимать.

Каждый раздел нужно формализовать – либо свести в таблицу (послужной список, образование), либо каждый новый элемент (ФИО, адрес и т.д.) начать с новой строчки. Необязательно прорисовывать рамки и разделительные линии таблиц – достаточно, если текст будет расположен достаточно строго по колонкам и хорошо структурирован. Озаглавьте каждый раздел и напишите заголовки крупным шрифтом, чтобы глаз легко выхватывал нужные сведения.

Не злоупотребляйте мелкими элементами оформления – курсивом, жирным шрифтом, подчеркиванием, словами из заглавных букв. Не превращайте текст в лоскутное одеяло.

Общие рекомендации

Побольше фактов и никаких оценок

Следует подробно излагать только «голые» факты, избегая лестных самооценок наподобие «большой опыт работы», «умение работать в команде», «хорошие организаторские способности» и т.п. Резюме нужно составлять так, чтобы менеджер сам сделал нужные Вам выводы.

Пусть за Вас говорят факты. Так, из послужного списка должно быть видно, сколько времени Вам приходилось руководить группой, отделом, предприятием, сколько людей было в подчинении, какие именно сложные задачи приходилось выполнять, какие награды и вознаграждения Вы получали.

Факты бывают разные. Не занимайтесь очковтирательством! Нередко студенты-программисты претендуют на общий стаж программирования в 40–50 лет, указывая: «Visual C++ – 5 лет, Delphi – 5 лет, Pascal – 9 лет, Assembler – 7 лет, Basic – 8 лет». Менеджер сам увидит из послужного списка, сколько времени каким инструментом пользовался кандидат, и сделает соответствующие выводы.

Нет – ложной скромности!

Ни в коем случае нельзя опускать важные для работодателя положительные факты – например, о численности возглавляемой бригады или об участии в важных и крупных работах. Во-первых, это ухудшает качество резюме. Во-вторых, подрывает доверие к самому кандидату, если в ходе собеседования будет обнаружено сокрытие важных фактов. Не имеет значения, положительный факт скрыт или отрицательный.

Как отмечалось выше, настоящая скромность заключается в том, чтобы не давать оценок самому себе. Предоставьте сделать это другим на основе тщательно отобранных и правильно представленных фактов Вашей трудовой биографии.

Свойственная многим застенчивость и ложная скромность не украшает, а унижает всех участников процесса приема на работу. Не заставляйте нанимателя устраивать перекрестный допрос, чтобы клещами вытянуть из Вас важные сведения! Не вынуждайте его наводить справки о Вас через Ваших прошлых работодателей! Ваша задача – представить ему всю необходимую информацию о себе – так, чтобы у него не возникло никаких дополнительных вопросов.

Конечно, не надо врать и приписывать себе несуществующие заслуги. Все, что требуется – это не скрывать о себе положительных сведений. Ваше резюме не должно выглядеть как резюме специалиста второго сорта! Ведь есть другие кандидаты, быть может, менее достойные, но с лучше составленным резюме.

Поэтому сделайте так, чтобы Ваше резюме «заиграло»! Составив резюме, перечитайте его на следующий день. Если Вам покажется, что Вы читаете о ком-то другом, кто лучше Вас и кого Вы лично уважали бы – значит, резюме хорошее.

Адаптируйте резюме к требованиям работодателя

Важно, чтобы в тексте резюме учитывались будущая работа и должность. Если, например, Вы претендуете на должность специалиста по тестированию или документированию программного обеспечения, не стоит делать акцент на том, что Вы – программист высокого класса, даже если Вы им и являетесь. Ведь Вашу квалификацию могут счесть слишком высокой для этой работы (*overqualified*) и отказать, понимая, что Вы не сможете качественно выполнять работу, ведущую к дисквалификации. Да и какому менеджеру (особенно если он тоже бывший программист) захочется иметь в подчинении специалиста более высокой квалификации, чем он сам?..

Если Вам непременно нужна эта работа, следует преобразовать резюме так, чтобы Вашим основным занятием выглядело именно тестирование или соответственно документирование, а остальные виды деятельности (программирование, внедрение, сопровождение) – вспомогательными.

Аналогично следует поступать, если Вы претендуете на изменение рода выполняемой работы – следует показать, что Вы всю жизнь как будто только тем и занимались, что работали по Вашей новой специализации.

Многое также зависит от адресата резюме. Скажем, отечественный работодатель склонен воспринимать содержание резюме всерьез, в то время как западный рассматривает резюме лишь как способ продемонстрировать желание устроиться на данную работу (т.е. ожидает резюме, «заточенное» под будущую должность) и ни на секунду не сомневается, что 90% указанного в резюме – ложь или серьезное преувеличение.

Нет и не может быть единого резюме на все случаи жизни, которое можно было бы рассылать по всем фирмам без изменений. Каждый раз следует прежде подумать, какие качества будут цениться на новой работе, и в соответствии с ними модифицировать резюме.

ПРИЛОЖЕНИЕ 2

Собеседование при приеме на работу

Собеседование является наиболее важным этапом приема на работу. От качества его проведения зависит эффективность, а иногда и судьба целой организации.

Цели проведения собеседования:

1. Определить годность кандидата с точки зрения профессиональных качеств

Чтобы определить уровень компетентности кандидата, ему задаются специфические вопросы, которые позволяют выявить специальные знания. Вопросы должны быть двух типов: на общую эрудицию («Какие книги, газеты, журналы Вы читаете по предмету?», «Какие Вы знаете средства разработки под Windows?») и на конкретные знания, необходимые для работы на данном участке. Важно также узнать, как кандидат учился в вузе, вел ли научную работу, имел ли приработок во время учебы. Не стоит придираться к незнанию конкретных вещей: часто толковые практики обладают низкой эрудицией, а вот среди читавших все на свете и блестяще отвечающих на все вопросы кандидатов частенько попадаются пустозвоны, не умеющие применить свои энциклопедические познания на практике. Как говорит Ф.Брукс в своей книге «Мифический человеко-месяц», «редко встречаются люди, сочетающие в себе большие административные и технические способности. Мыслители встречаются редко, практики еще реже, но реже всего – мыслители-практики».

Не придирайтесь, если кандидат не знает о паре каких-либо особенных «фишек», о которых знаете Вы и Ваши коллеги. Для узких специалистов играют роль не только и не столько конкретные знания, сколько способность учиться и осваивать новое. Иными словами, крайне важно выявить потенциал специалиста, который тесно связан с его способностью самостоятельно учиться.

Ни в коем случае не следует доверяться так называемым references – письменным или устным рекомендациям. Эта вещь хорошо работает на Западе, но совершенно не срабатывает у нас. Дело в том, что из-за резко сниженного (по сравнению с Западом) понятия чести наши рекомендующие часто дают заведомо ложные рекомендации, не соответствующие действительности. Иногда встречаются факты лжесвидетельства даже в отношении совсем незнакомых людей (fake experience). Поэтому в наших условиях часто само наличие рекомендаций дает повод подозревать кандидата во лжи. Особенно это касается тех, кто никогда не сталкивался с практикой работы в США или других западных стран (это легко выяснить из резюме). Причем не имеет значения, исходит рекомендация от Вашего знакомого или от человека, знакомого кандидату. Опыт показывает, что даже самые лучшие друзья, родственники или коллеги рекомендуют кандидатов исходя из своих «добрых побуждений». Они могут руководствоваться собственными корыстными или бескорыстными интересами, совершенно не задумываясь о последствиях для Вас и для себя. Отнеситесь к рекомендациям с поправкой на того, кто рекомендует и почему.

2. Определить годность кандидата с точки зрения человеческих качеств

В этом случае оценке подлежит все: поведение кандидата, его манера одеваться и вести себя, его высказывания о предмете, о мире, об окружающих, о прошлых работодателях.

Необходимо отместить кандидатов, ведущих себя вызывающе, протягивающих руку первыми, спорящих с интервьюером, настойчиво пытающихся начать беседу с обсуждения будущей зарплаты. Не заслуживают доверия и кандидаты, уходящие от прямых ответов, отвечающие не на тот вопрос, который был задан, спокойно рассказывающие о том, как они обманывали кого-либо. Эти люди не только будут плохими работниками, но и ухудшат моральный климат в коллективе, а в критических ситуациях их поведение непредсказуемо.

Интервьюеру следует вести себя дружелюбно, любезно, слушать внимательно. Необходимо, чтобы претендент раскрепостился и был искренен. Хорошо предложить кофе, чай и т.п. В то же время нельзя никоим образом высказывать свое отношение к услышанному. Например, нельзя хвалить или тем более осуждать поступки претендента. Лучшая тактика интервьюера – побольше молчать, улыбаться и давать говорить кандидату.

Степень общительности претендента следует рассматривать с точки зрения его будущей должности. Скажем, человек с хорошо подвешенным языком будет хорошо справляться с работой снабженца, управленца, менеджера по продажам или специалиста по технической поддержке, но может оказаться совершенно беспомощным в программировании. Такой человек будет докучать соседям разговорами, отвлекая их от работы и не работая сам. Напротив, наиболее сильные программисты – «хакеры» – это, как правило, погруженные в собственный мир закомплексованные нелюдимы, не умеющие связать двух слов. Они обычно боятся интервьюера, незнакомой обстановки и т.п. Эти их недостатки не будут сказываться при работе в составе группы «нормальных» программистов. «Хакер» быстро привыкнет к ним и перестанет бояться. Окружающие будут его опекать. А производительность и качество работы этого трудяги может быть в разы выше, чем у остальных программистов.

Необходимо обратить особое внимание на способность кандидата добиваться поставленных целей даже вопреки нехватке средств или знаний. Оценить работоспособность кандидата в критических ситуациях. Эта способность хорошо соотносится с занятиями всеми видами спорта и хобби, в которых кандидату удалось добиться серьезных успехов (спортивный разряд, призер олимпиады по любому школьному или институтскому предмету).

Привычка хорошо трудиться связана с работой над собой и со способностью засиживаться за интересным делом или работой. Невинный вопрос «Часто ли приходится Вам засиживаться на работе допоздна?» позволяет отсеять людей, не привыкших напряженным трудом добиваться поставленных целей, а лишь отсиживающих на работе «от звонка до звонка».

Вообще, у каждого опытного интервьюера есть как минимум десяток подобных вопросов-«коньков» с подвохом, позволяющих выявить тот или иной аспект профессиональных или человеческих качеств кандидата. Список таких вопросов интервьюеру необходимо постоянно расширять.

3. Определить пригодность кандидата для данной конкретной работы

Чтобы узнать, будет ли человек соответствовать той конкретной работе, которую ему собираются поручить, необходимо определить, на каком жизненном этапе находится кандидат, какова система его жизненных ценностей и приоритетов.

Каждый человек находится на разных этапах своего жизненного пути. Скажем, молодому холостому выпускнику вуза более важно получить интересную работу, даже если она не очень хорошо оплачивается. Он готов сидеть на работе дни и ночи напролет. В то же время 30-летнему семейному человеку наряду с творческим аспектом крайне важен и материальный. 40-летнему может быть необходим достаточный статус в группе, и ему тяжело сидеть за работой по 12–14 часов ежедневно. Для 50-летней женщины или 55-летнего мужчины важны высокий официальный оклад и возможность спокойно без особых производственных перипетий доработать до пенсии.

Молодые люди, особенно программисты-мужчины, избегают работу, связанную с людьми (поддержка пользователей, управление и т.п.). Они любят заниматься конкретным проектом так, чтобы ни с кем не делить свою работу и иметь как можно меньше связей с другими участками производства. Напротив, женщины и более взрослые люди часто предпочитают работу в группе, и им не так важно сделать проект единолично от начала до конца.

Далее у каждого человека жизнь складывается по-разному. Одни достигают в юности больших успехов (победители школьных олимпиад, юные гении), после чего, не сумев «пройти медные трубы», как бы успокаиваются, думая, что для них нет ничего невозможного. Всю последующую жизнь они, лентясь, выезжают на старом багаже. Часто такие люди заканчивают вуз с тройками или даже бросают его, ибо «всё и так

знают, а в вузе все равно ничему не учат», и идут работать. Таких специалистов-любителей, которые с готовностью берутся за любую работу и неизменно проваливают ее, лучше избегать.

Другие, напротив, перебиваясь с троек на четверки в школе и на первых курсах вуза, постепенно приспособляются к жизни и с каждым годом достигают все больших успехов в учебе и работе, идя по жизни по восходящей линии.

Бывают жизненные ситуации и посложнее, когда человек резко меняет тип деятельности и статус, скажем, демобилизовавшись после многолетней службы в армии в расцвете своих способностей (иногда даже в 30-35 лет). Тут нужно быть особенно внимательным, попытаться разгадать тип личности, стать на его место и понять его мотивы.

4. Общие рекомендации

Никогда не начинайте интервью без предварительного знакомства с резюме. Это позволит Вам сэкономить массу времени во время интервью, а также оценить способность кандидата к грамотному письменному изложению фактов.

Примите кандидата как своего доброго знакомого, предложите ему чай или кофе, дайте ему раскрепоститься и раскрыться. Ведите себя непринужденно. Позвольте ему как можно больше говорить о себе, не прерывайте. Вы узнаете много интересного и наилучшим образом поймете кандидата.

Ни в коем случае не стоит цепляться к отдельным словам. Необходимо смотреть на кандидата «в целом» и выносить суждения только на основе множественных подтверждений своих предположений. Для этого интервью должно быть длительным – обычно не менее полутора-двух часов.

Психологические тесты полезны, однако не стоит ими увлекаться всерьез. Во-первых, по-настоящему полезные тесты (а не примитивные газетные анкеты-опросы) может правильно подобрать и интерпретировать только профессиональный психолог. Для того чтобы научить их проведению

Вас, психолог должен вместе с Вами протестировать хотя бы десяток кандидатов, иначе результаты будут искажены. Во-вторых, результаты психологических тестов являются вспомогательными и могут лишь подтвердить Вашу точку зрения, но ни в коем случае не могут подменить или опровергнуть ее. Это знает любой профессиональный психолог, но часто не подозревает психолог-любитель, отвергающий кандидата только из-за того, например, что «его тест на лживость плохой».

Если Вы решили, что кандидат Вам может подойти, то собеседование заканчивается словами «Позвоните мне завтра (послезавтра) в такое-то время». Не говорите сразу, что кандидат Вам подходит. Подумайте, наведите справки. Если Вы уверены, что не подходит – проводите словами «Спасибо, когда Вы нам понадобится, мы Вам позвоним». Как знать – а вдруг действительно понадобится?

Указатель имен

Брукс, Фредерик (Frederick Brooks, род. в 1931 г.) – выдающийся специалист в области управления крупными проектами по разработке программного обеспечения. Лауреат премии Тьюринга, визионер, первый руководитель знаменитого проекта операционной системы IBM OS/360 для мэйнфреймов корпорации IBM. Наиболее известен своей книгой «Мифический человеко-месяц». Эта книга, посвященная управлению программистскими проектами, вместе с более поздней статьей-докладом «Серебряной пули нет», – пожалуй, единственный труд в области программирования, который за 30 лет практически не устарел, особенно на фоне большинства книг по компьютерным наукам, теряющих актуальность чуть ли не в день выхода. С 1964 г. Ф.Брукс – профессор университета штата Северная Каролина.

Вирт, Никлаус (Nicklaus Wirth) – профессор Цюрихской высшей технической школы, автор-разработчик языков программирования Паскаль и Модула.

Гейтс, Билл (Bill Gates, sir William Henry Gates III, род. в 1956 г.) – американский программист на Basic, выдающийся менеджер-практик, основатель и руководитель корпорации Microsoft, самый богатый человек в мире на протяжении многих лет. Основал Microsoft в 1975 г., будучи на 4-м курсе престижного университета, из-за чего бросил учебу. Впоследствии он объяснял это тем, что не мог упустить однажды в жизни представившийся ему момент вхождения в зарождавшийся бизнес программного обеспечения персональных компьютеров. Обеспечил плавный рост численности работников компании примерно на 70% в год на протяжении всех лет ее существования. Внедрил электронную почту у себя в компании еще в 1977 году при численности персонала около 10 человек в целях повышения качества управления. На протяжении всей деятельности в корпорации большое внимание уделяется борьбе с конкурентами. Так, в 80-х годах корпорация Apple Computer Products решила постепенно избавиться от

монополизма Microsoft на платформе Macintosh, разработав собственный компилятор Basic. В ответ на это Гейтс пригрозил, что если Apple это сделает, то Microsoft прекратит поставки Microsoft Office для этой платформы. В результате Apple была вынуждена отказаться от своих планов. Но самая известная история такого рода – отход Microsoft от совместной деятельности с IBM по разработке операционной системы Windows, когда Microsoft выпустила собственный продукт Windows 95, являющийся прямым конкурентом OS/2 – совместного продукта Microsoft и IBM. Считается, что Билл Гейтс – визионер, т.е. человек, умеющий предугадывать развитие компьютерной отрасли на годы и десятилетия вперед. Но это правда лишь отчасти, ибо число неуспешных проектов Microsoft не меньше, чем у других компаний. На самом деле Microsoft и сам Гейтс искусно манипулируют потребителями, навязывая им через прессу и публичные выступления выгодное корпорации мнение задолго до выхода продуктов на рынок, делая это раньше и чаще конкурентов, что и обеспечивает потрясающий успех корпорации. Является прекрасным управленцем и маркетологом, умеющим мотивировать людей и понимающим человеческую психологию. В то же время Гейтс сух, молчалив и имеет устойчивую репутацию нелюдима и трудоголика (впервые женился в 38 лет на маркетологе собственной компании). «Большой Билл» не умеет проговариваться публично, он говорит вслух и пишет только то, что «нужно», из-за чего многие критикуют его книги и публичные высказывания за неискренность. Это не помешало ему создать одну из самых успешных компаний в мире.

Герцберг, Фредерик (Frederick Herzberg, род. в 1923 г.) – американский специалист в области клинической психологии, профессор менеджмента в университете штата Юта. Труды Герцберга посвящены в основном особенностям личности работающего человека, но пользуются популярностью у теоретиков и практиков менеджмента, поскольку расширяют знания менеджмента о персонале и позволяют оптимизировать труд работников. Герцберг создал собственную теорию мотивации, которую можно условно разделить на две части – гигиена и мотивация. Под гигиеной Герцберг подразумевает политику и методы управления в компании, условия

работы, жалование, степень защищенности; все эти факторы не служат мотивами для повышения производительности, но создают моральное удовлетворение. Вторая часть теории мотивации касается собственно работы, выполняя которую, работник достигает определенных результатов, получает признание окружающих, продвигается по служебной лестнице, повышает свой статус, имеет возможность заниматься любимым делом. Эта группа факторов обусловлена личными склонностями работника и служит прямым стимулом к труду. Менеджеры должны использовать одновременно оба фактора – фактор гигиены и фактор мотивации, создавая такие условия труда, чтобы работник не испытывал неудовлетворенности. Если работник сможет добиться результатов, получить признание, найти интерес, продвигаться по служебной лестнице, то он будет работать с максимальной отдачей. Правда, у Герцберга есть еще одна теория, которая называет КІТА (a kick in the ass – пинок под зад). Эта теория гласит: самый простой способ заставить человека трудиться – давать ему КІТА, потому что улучшение гигиены (повышение жалования, условий труда, предоставление дополнительных льгот – пенсий, оплаченных отпусков и др.), не дает длительного эффекта мотивации. Мотивация зависит от эффективности использования работников, а не от того, как с ними обращаются.

Дейкстра, Эдсгер (Edsger Dijkstra, род. в 1930 г.) – профессор-математик, лауреат премии Тьюринга, автор многих книг и статей в области компьютерных наук, написанных преимущественно в период его жизни на родине в Голландии и преподавания в Эйндховенской технической школе. Область интересов – правильность программ, алгоритмы и системы. Его самые известные научные результаты – алгоритм Дейкстры (поиск кратчайшего пути на графе), структурное программирование (без оператора GO TO), а также концептуальный труд «Дисциплина программирования», представляющая собой курс лекций по «правильному» программированию, посвященный методике и философии программирования методом пошагового уточнения 70-х гг. эмигрировал в США. С тех пор работает в крупных компаниях и преподает в университете штата Техас в Остине.

Деминг, Эдавардс (W. Edwards Deming, 1900–1993 гг.) – всемирно известный теоретик менеджмента и консультант, чья работа помогла японской промышленности усвоить новые принципы управления, совершив революцию в качестве и производительности. Успехи Японии после Второй мировой войны во многом связаны с внедрением в Японии американских принципов управления. В течение 40 лет д-р Деминг исследовал мировую практику менеджмента и в результате сформулировал 14 принципов качества, во многом созвучных с идеями Друкера, например следующие: организация должна иметь четко сформулированную миссию; качество не может быть обеспечено за счет проверок, оно должно быть результатом работы; необходимо совершенствовать производственный процесс, а не только его результаты; человек является решающим звеном повсюду, в каждом процессе, даже в полностью автоматизированном; постоянное повышение квалификации – это не лозунг, а насущная необходимость.

Друкер, Питер (Peter F. Drucker, 1909–2001 гг., иногда его фамилию ошибочно переводят как «Дракер») – считается старейшим и самым последовательным теоретиком менеджмента. Именно он превратил менеджмент – непопулярную и неуважаемую в 50-е годы специальность – в научную дисциплину. Питер Друкер родился в Вене. Он получил юридическое образование и работал журналистом. В 1937 году переехал в Соединенные Штаты. В 1943 году Друкер провел полный анализ структуры и управления General Motors Co. В течение шести десятилетий Друкер был самым желанным советником исполнительных директоров, среди которых были Уолтер Ристон из Citicorp, Давид Рокфеллер из [Chase Manhattan Corp.](#), Генри Льюс из [Time Inc.](#) и Марк Уиллес из Times Mirror Co. Он изобрел термин «управление посредством целей» и помог разработать объективные меры зарплаты и продвижения. Друкер ориентировал компании на клиентов, а не только на технологию. Его последняя известная работа, переведенная на русский язык – «Менеджмент в XXI веке» – посвящена значительному повышению роли человеческого фактора в управлении.

Питер Друкер известен не только как автор трудов по менеджменту, но и аналитической литературы, посвященной политике, экономике и обществу.

Дюпон, Пьер Самюэль (Pierre S. Du Pont, 1870–1954 гг.) – американский промышленник из семьи французских эмигрантов; разбогател на военных поставках во время Первой мировой войны. После окончания войны Дюпон поместил большую часть своих капиталов в General Motors, что позволило ему в 1920–1923 годах занимать должность президента компании, а в 1923–1929 – председателя Совета директоров. Будучи одновременно членом Советов директоров многочисленных банков и корпораций, Дюпон немалые средства вкладывал в образовательные учреждения и занимался филантропией.

Йордон, Эдвард (Edward Yourdon) – программист, разработчик ряда структурных и объектных моделей проектирования и конструирования программ, независимый консультант в области программирования, руководитель американской консалтинговой компании Yourdon, Inc, журналист, публичный лектор. Автор около 30 блестяще написанных книг, популяризирующих важные идеи, явления и события в области программирования, например: «Путь камикадзе» (есть русский перевод) – руководство по выживанию в безнадежных проектах, «Упадок и падение американского программиста» – о нашествии программистов-эмигрантов, книгу о «проблеме Y2K», книгу по объектно-ориентированному проектированию специально для тех, кто привык к структурным моделям (есть русский перевод), и т.д. Написанные с замечательным чувством юмора, книги Йордона - увлекательное, легкое и в то же время серьезное и чрезвычайно полезное чтение.

Кнут, Дональд (Donald Knuth, род. в 1938 г.) – профессор Стэнфордского университета (США), выдающийся специалист-теоретик в области computer science, создатель широко распространенного в научной среде многоплатформенного текстового редактора TeX. Широко известен специалистам своей многотомной монографией «Искусство программирования для ЭВМ», из 7 томов которой пока написано 3 и в настоящее время пишется 4-й. Не так давно ликвидировал свой электронный адрес, заявив, что больше не имеет физической возможности отвечать на письма. Большую часть своего времени работает над 4-м томом своей

«компьютерной Библии», изредка читая публичные лекции и давая редкие интервью, занимается богословскими исследованиями и играет на органе.

Кодд, Едгар (Edgar Codd) – профессор-математик, лауреат премии Тьюринга. В 1970 г. Кодд изобрел реляционную модель данных, лежащую в основе всех сегодняшних крупнейших коммерческих СУБД.

Макнили, Скотт (Scott McNealey) – основатель американской корпорации Sun Microsystems, производителя рабочих станций среднего класса SPARC на базе процессора MIPS и разнообразного программного обеспечения – языка программирования Java, операционной системы Solaris, протокола NFS и др. Харизматичная личность и энергия Скотта во многом обеспечивает успех всей компании. Публичные выступления Макнили пользуются фантастическим успехом, люди специально приходят его послушать. Иногда его приглашают выступать даже конкуренты. Скотт со своим почти двухметровым ростом в свое время успешно играл за известные американские хоккейные клубы, и клубы-конкуренты вздохнули с облегчением, когда он полностью ушел в бизнес.

Маслоу, Абрахам (Abraham Maslow, 1908–1970 гг.) – американский психолог-бихевиорист.

Абрахам Маслоу родился в Нью-Йорке, он был самым старшим из семи детей. Отношения Маслоу со своими родителями не отличались ни близостью, ни любовью. Среднее образование он получил в бесплатных нью-йоркских школах. Он сам говорил о том, что до двадцати лет был чрезвычайно стеснительным, нервным, подавленным, одиноким, погруженным в себя и страдающим от жестоких неврозов человеком. В школе он был в изоляции, семью свою не выносил, и поэтому его настоящим домом была библиотека. Он был одним из лучших учеников в школе. В конце двадцать восьмого года, когда ему было двадцать лет, он поступил в Мэдисонский университет (Висконсин), где стал бакалавром (1930), магистром (1931) и доктором (1934) психологии.

Преподавать Маслоу начал в колледже Бруклина, где работал с 1937 по 1951 г. За этот период эмигрировавшие из Германии психологи

превратили Нью-Йорк в интеллектуальную столицу Америки. Маслоу работал с Максом Вертхаймером, Эрихом Фроммом, Карен Хорни, Куртом Голдстейном и Рут Бенедикт.

Вскоре после того как Маслоу покинул Мэдисон, он пришел к убеждению, что большинство современных психологов, как исследователей, так и теоретиков, излишне полагаются на данные наблюдений за лицами, обратившимися к психологам по патологическим причинам. Образ человеческой природы, сложившийся в результате исследования этих пациентов, не мог не быть искаженным и пессимистическим. Пытаясь исправить это положение, Маслоу начал изучать тех людей, которых он считал наилучшими образцами здоровой личности. Он назвал этих людей "самоактуализующимися", поскольку их отличала потребность в деятельности, творчестве, ответственность, открытость и справедливость.

В написанной им в 1943 г. эпохальной статье "Теория человеческой мотивации" и (более подробно) в работе "Мотивация и личность" (1954) Маслоу утверждал, что у людей есть высшие и низшие потребности. И те, и другие являются "инстинктоидными" и расположены в следующем иерархическом порядке: физиологическое благополучие, безопасность, любовь, уважение и самоактуализация. Каждая высшая группа потребностей зависит от предварительного удовлетворения потребностей низшего уровня.

Таким образом, по мнению Маслоу, человеческая природа представляет собой непрерывное удовлетворение внутренних потребностей, начиная с основных физиологических потребностей и вплоть до метапотребностей. Он утверждал, что самореализующиеся личности – это люди, которые уже удовлетворили свои низшие потребности и стремятся осуществить высшие устремления человеческой природы, становясь всем, чем они способны стать.

Назначение иерархии потребностей Маслоу первоначально состояло в объяснении человеческого поведения, однако она была быстро взята на вооружение менеджерами, поскольку позволяла понять особенности мотивации работников. Маслоу стал одним из первых деятелей менеджмента, которые использовали гуманистический подход к персоналу взамен

административного. Учитывая, что именно персонал становится ключевым ресурсом преуспевающих компаний, модель Маслоу в качестве концепции менеджмента становится все более актуальной.

Слоун, Альфред (Alfred Sloan, 1875–1966 гг.) – менеджер и теоретик менеджмента, почти 50 лет работал в компании General Motors. Поступив в нее на работу в 1921 году, когда компания имела восемь моделей автомобилей и 12% рынка, значительно уступая компании Ford с ее 60% рынка и всего одной моделью, Слоун начал с разработки идеальной схемы бизнеса для американской автомобилестроительной компании. Он считал, что компания должна иметь пять моделей, охватывающих все сегменты рынка. Слоун не разделял точку зрения Форда, утверждавшего, что клиента интересует только цена автомобиля. Слоун был убежден, что автомобиль должен отражать продвижение по социальной лестнице, следовательно, каждая модель должна свидетельствовать о некоем социальном статусе. Компания должна максимально облегчить клиенту продажу старой машины и покупку новой. Слоун добился того, что клиент, купивший самую дешевую модель General Motors, мог легко накопить денег на самую дешевую машину следующей модели. В результате возник рынок подержанных автомобилей, прослуживших всего один-два года. При этом каждая из моделей General Motors конкурировала не только с автомобилями других компаний, но и с машинами General Motors других классов. Благодаря такой стратегии все подразделения компании находились в непрерывном поиске, что позволило General Motors оставаться лидером почти 30 лет. Слоун считал, что успех при отсутствии конкуренции опасен, так как ведет к стагнации. Вся концепция была разработана Слоуном всего за месяц при помощи небольшой группы менеджеров. Информацию он черпал из собственных наблюдений за рынком и бесед с работниками и автомобильными дилерами. Успех его достаточно простой схемы был обусловлен системным подходом к созданию идеального бизнеса и правильным пониманием маркетинга – создавать и производить товар, который нужен клиенту. Слоуна называют изобретателем современного менеджмента.

Уотсон, Томас (Tomas J. Watson, 1914–1993 гг.) – руководитель корпорации IBM, блестящий управленец-практик. Превратил IBM из небольшой фирмы по производству кассовых аппаратов и перфокарточных счетных машин в мирового лидера в области вычислительной техники и программирования.

Уотсон-младший, возглавивший компанию в 1956 г., сразу ринулся в совершенно новую для себя сферу, имевшую большие перспективы. Уже тогда компьютеры считались многообещающим бизнесом, и в эту сферу устремились и другие, более крупные в то время концерны – General Electric, Radio Corporation of America (RCA), Hewlett-Packard и др. Фирма Уотсона сумела не только обогнать их, но и в короткое время (к началу 70-х годов) превратилась в фактического монополиста по производству крупногабаритных, а с 80-х годов и лидера в производстве персональных компьютеров. В дальнейшем положение IBM было поколеблено распространением мини-компьютеров, однако сегодня IBM с большим отрывом лидирует в мире по продажам как компьютеров, так и программного обеспечения.

Уотсон увидел свой главный рынок сбыта в крупных компаниях, для которых компьютеры означали техническую революцию в отчетности, контроле над финансовыми потоками и производством. Среди ведущих покупателей были также банки, страховые компании и другие финансовые институты. Уотсон понимал, что от научных учреждений много прибыли не получишь, и потому не старался лидировать в сфере сверхсложных и сверхскоростных компьютеров. Иначе говоря, он увидел в компьютере средство совершенствования простых операций, т.е. замену арифмометра, а позже печатной машинки. Именно в этой области мог возникнуть массовый спрос на компьютеры. Его расчет оказался правильным. Но и решение такой, казалось бы, простой задачи, потребовало удвоения расходов IBM на научные исследования и опытно-конструкторские разработки (НИОКР). Были приглашены сотни инженеров – специалистов по электронике. Кадры фирмы пришлось радикально обновлять. Вместо традиционных менеджеров-чиновников Уотсон окружил себя техническими

талантами, которые не боялись перечить начальству и говорить правду в глаза. Гений Томаса Уотсона заключался в понимании, что ускоренный рост продаж и прибылей зависит от поддержания быстрой смены моделей, которые успевали бы за техническими достижениями, какими бы малыми они ни были. Он не стремился делать из каждого выпускаемого компьютера совершенство. Ему важно было оставаться впереди других. Скорость решения технических проблем стала для него первым критерием. «Не тяните, – требовал он от сотрудников, – решайте проблемы быстро, не важно – правильно или нет. Если сделаете ошибку, она быстро вас ударит по морде и заставит себя исправить».

Томас Уотсон вошел в четверку крупнейших предпринимателей XX века, опубликованную в конце 1999 г. журналом Fortune. За время своей работы Уотсон наглядно продемонстрировал, что современный технический прогресс осуществляется быстрее всего, если НИОКР становятся органичной частью производства. В последние десятилетия это стало правилом для большинства западных фирм.

Форд, Генри (Henry Ford, 1863–1947 гг.) – американский промышленник, один из основателей автомобилестроительной промышленности США. В 1892-1893 году под его руководством был создан первый автомобиль с четырехтактным двигателем (Ford T); в 1903 году основал компанию Ford Motors Company. Впервые применил конвейер и внедрил поточно-массовую организацию производства, основанную на разделении труда и принципах Тейлора, что привело к многократному повышению производительности труда, многократному падению цен на автомобили и – как следствие – к их общедоступности, что во многом определило облик современной Америки. Форд не считал принципы менеджмента универсальными; он полагал, что их применение и эффективность зависят целиком от личности владельца (генерального директора) предприятия.

Литература

1. Мескон М.Х., Альберт М., Хедоури Ф. Основы менеджмента: – М.: Дело ЛТД, 1994.
2. Брукс Фр. Мифический человеко-месяц, или Как создаются программные системы. СПб.: Символ-Плюс, 1999.
3. McConnell, Steve. Rapid development: taming wild software schedules. Microsoft Press, 1996.
4. Котлер Ф. Маркетинг и менеджмент. Сер. Теория и практика менеджмента – СПб: Питер, 2001.
5. Друкер П. Ф. Задачи менеджмента в XXI веке.: М.: Вильямс, 2001.
6. Гейтс Б. Дорога в будущее. М.: Русская редакция, 1996.
7. Роджерс (Бак) Ф. Дж. Путь успеха: Как работает корпорация IBM. СПб.: Терра, 1997.
8. Мир управления проектами / Под ред. Х. Решке, Х. Шелле. – М.: Аланс, 1993.
9. Гейтс Б. Бизнес со скоростью мысли. – М.: ЭКСМО-Пресс, 2001.
10. Вахрушев В. и др. Эти невероятные японцы. Принципы японского управления. Сер. Психотехнологии и эффективный менеджмент, кн.3. : Технологическая школа бизнеса, М., 1992.
11. Надежные программы, 2000, № 1(8). с.35-45.

Учебное издание

Зиссер Юрий Анатольевич
Мелешенко Александр Александрович

Управление разработкой и продвижением программного обеспечения

Учебное пособие
по курсу «Менеджмент производства программного продукта» для студентов
специальности 31 03 04 «Информатика»

В 2-х частях

Часть I
Менеджмент

Редактор Т.Н. Крюкова
Компьютерная верстка Е.Г. Реут

Подписано в печать 30.08.2002.

Формат 60х84 1/16.

Бумага офсетная.

Печать ризографическая.

Усл.-печ.л. 9,65.

Уч.-изд.л. 7,8.

Тираж 150 экз.

Заказ 293.

Издатель и полиграфическое исполнение:

Учреждение образования

"Белорусский государственный университет информатики и радиоэлектроники"

Лицензия ЛП № 156 от 05.02.2001.

Лицензия ЛВ № 509 от 03.08.2001.

220013, Минск, П. Бровки, 6.