**1. Процессы управления человеческими ресурсами: планирование, набор, развитие команды проекта, управление командой проекта**

Процессы управления человеческими ресурсами включают в себя:

1. планирование человеческих ресурсов - определение и документальное оформление ролей, ответственности и подотчетности, а также создание плана обеспечения проекта персоналом.

2. набор команды проекта - привлечение человеческих ресурсов, необходимых для выполнения проекта.

3. развитие команды проекта - повышение квалификации членов команды проекта и укрепление взаимодействия между ними с целью повышения эффективности исполнения проекта.

4. управление командой проекта - контроль за эффективностью членов команды проекта, обеспечение обратной связи, разрешение проблем и координация изменений, направленных на повышение эффективности проекта.

**1 процесс: Планирование человеческих ресурсов**

**Планирование человеческих ресурсов** проводится с целью определения ролей, ответственности и подотчетности в проекте, а одним из основных его результатов является разработка плана обеспечения проекта персоналом.

План обеспечения проекта персоналом, как правило, включает в себя определение сроков и способов набора членов команды проекта, критерии их освобождения от участия в проекте, рекомендации по проведению дополнительного обучения, схемы поощрения и награждения.

В качестве одного из основных инструментов, используемых при планировании человеческих ресурсов, используются **организационные диаграммы и назначения по проекту**.

Существуют различные форматы документирования распределения ролей и ответственности членов команды проекта. Большинство форматов относятся к одному из трех типов

1. иерархическому,

2. матричному

3. текстовому.

Независимо от того, какая комбинация методов используется, цель всегда одна - добиться того, чтобы для каждого пакета работ был назначен один ответственный за его исполнение, и чтобы каждый член команды четко понимал свою роль и ответственность.

**Иерархическая диаграмма** строится сверху вниз в соответствии с имеющейся структурой подразделений организации (отделов, групп или команд). Под каждым отделом указывается список операций проекта или пакет работ. Таким образом, можно увидеть всю ответственность в проекте для данного функционального отдела (например, отдела информационных технологий или отдела закупок) в одном месте рядом с названием отдела. Можно строить иерархическую диаграмму, взяв за основу разбиение по типам ресурсов. В этом случае можно отобразить в виде отдельного объекта, например, всех программистов, несмотря на то, что они могут быть закреплены за различными отделами.

**Матричные диаграммы. Матрица ответственности** используется для отображения связей между выполняемыми работами и членами команды проекта. Она позволяет увидеть все операции, назначенные к выполнению определенному человеку, или отобразить всех людей, принимающих участие в выполнении определенной операции. Количество видов ответственности может быть различным, в зависимости от специфики проекта и его организации

Пример типовой матрицы ответственности приведен на рис. 2.

Другие варианты функций исполнения при назначении могут быть например такими:

И -- исполнение,

СИ -- соисполнение,

ОИ -- ответственное исполнение,

ТО -- текущая ответственность,

У -- утверждение,

С -- согласование,

О -- обсуждение,

Э -- экспертиза,

К -- консультирование,

УС -- участие в совещании и т. д.

**Текстовые форматы** используются, если требуется более детально охарактеризовать роли в проекте, текстовые форматы. Обычно в таких документах в краткой форме содержится следующая информация: обязанности, полномочия и квалификация.

**2 процесс: Набор команды проекта**

**Набор команды проекта**- это процесс привлечения человеческих ресурсов, необходимых для выполнения проекта.

Набор членов команды проекта осуществляется из внутренних и внешних источников.

Назначение персонала во многих проектах является предметом переговоров. Если у исполняющей организации для выполнения проекта не хватает штатных специалистов, то требуемые услуги можно получить из сторонних источников. Это может выражаться в найме консультантов или передаче работ сторонним организациям на условиях субподряда.

Широкие возможности по привлечению новых членов в команду проекта открывает создание виртуальных команд. Виртуальные команды можно определить как группы людей, объединенных общей целью, причем каждый член группы выполняет работу при минимальном личном контакте или при полном его отсутствии. Работа таких команд стала возможной благодаря электронным средствам коммуникации (например, электронная почта и видеоконференции).

При работе в условиях виртуальных команд все большее значение приобретает планирование коммуникаций. Возможно, потребуется дополнительное время для четкого определения ожиданий участников, разработки протоколов для разрешения конфликтов, вовлечения сотрудников в процесс принятия решений и поощрения за участие в общем успехе проекта.

К основным **факторам**, определяющим принципы формирования команды проекта, относятся:

1. Специфика проекта.

2. Организационно-культурная среда. Организационно-культурная среда команды проекта делится на внешнюю и внутреннюю. Внешняя среда включает окружение проекта во всех аспектах. Внутренняя среда или организационная культура самой команды определяется способами организации и протекания процессов командного взаимодействия (такими как координация, коммуникация, деятельность по разрешению конфликтов и принятию решений, налаживание внешних связей).

3. Особенности личного стиля взаимодействия руководителя с другими членами команды.

При формировании команды проекта стоит ещё обратить внимание на личные качества ее членов. Они должны быть сбалансированы. Чем более разноплановые люди присутствуют в команде, тем лучше для проекта. Не менее важен и тип темперамента человека. Никогда не сажайте холерика за бумажную работу, он очень быстро устанет и провалит вам всю работу. Меняйте деятельность внутри своей команды примерно раз в час. Это необходимо для эмоциональной разгрузки и позволяет ускорить процесс в несколько раз.

**3 процесс: Развитие команды проекта**

**Развитие команды проекта** предусматривает повышение квалификации членов команды проекта и укрепление взаимодействия между ними для повышения эффективности реализации проекта. Целями развития команды проекты являются:

- повышение навыков членов команды для повышения их способности выполнять операции проекта;

- укрепление чувства доверия и сплоченности среди членов команды для повышения продуктивности ее работы.

Среди основных мероприятий развития команды могут быть:

**1. Обучение**. Оновключает в себя все действия, направленные на повышение квалификации членов команды проекта. Обучение необходимо, когда нельзя найти нужных специалистов или когда необходимо сформировать общую терминологическую и методологическую базу. Обучение может носить как официальный, так и неофициальный характер.

2. Организация собраний и встреч, полностью или частично посвященных вопросам функционирования команды. Такие встречи могут проводиться в режиме мозгового штурма, где обсуждаются не вопросы проекта, а функционирование команды как таковой.

3. Проведение **действий по сплочению команды** - **ТИМБИЛДИНГ -**(тренинги, неформальные мероприятия), командному решению проблем, осуществление действий по поддержке чувства успеха проекта и команды.

**4. Предупреждение конфликтов** или работа с ними. Для избегания конфликтов необходимо заранее определить список приемлемых и неприемлемых моделей поведения, принятых командой проекта для улучшения рабочих взаимоотношений, эффективности и коммуникаций.

**5. Со-расположене.**Совместное размещение подразумевает размещение всех или большинства активных членов команды проекта в одном месте, чтобы укрепить их способность работать в единой команде.

**6. Действия по мотивированию участников**. Решения о премировании принимаются официально или неофициально в процессе управления командой проекта на основании результатов оценок эффективности. Премированию подлежит только желаемое поведение членов команды.

**4 процесс: Управление командой проекта**

Управление командой проекта включает в себя:

- контроль за деятельностью членов команды проекта,

- обеспечение обратной связи,

- разрешение проблем и координацию изменений, направленных на повышение эффективности реализации проекта.

Для эффективного управления командой проекта рекомендуется:

Обеспечить правильное руководство проектом и демонстрировать его реальное присутствие.

Обеспечить и демонстрировать поддержку проекту со стороны руководства материнской компании ивлиятельных внешних заинтересованных лиц.

Создать положительный имидж проекту в компании во внешней к проекту среде.

Осуществлять эффективное планирование проекта, обсуждать рабочие задания.

Осуществлять правильное комплектование команды иорганизацию работы, в том числе взаимосвязи и отношения подотчетности.

Спланировать правильный процесс работы, вызвать трудовой энтузиазм, рабочий настрой.

Способствовать эффективному обмену информацией, укреплять обязательность сотрудников.

Проводить мероприятия, содействующие сплочению команды.

Стремиться к разрешению конфликтов и проблем.

10. Обеспечить мотивацию, создать надлежащую систему вознаграждения команды.

Огромное значение имеет подержание правильного микроклимата внутри команды и особенно личность самого менеджера проекта и функционального руководителя. Если между этими двумя лидерами начинают возникать проблемы, то с достаточной долей вероятности можно говорить, что проект будет провален.

К корректирующим действиям по управлению человеческими ресурсами относятся:

- кадровые перестановки,

- проведение дополнительных тренингов

- меры дисциплинарного воздействия.

Если команда управления проектом обнаруживает потенциальные или возникающие проблемы, относящиеся к человеческим ресурсам, то для того, чтобы снизить вероятность возникновения и/или последствия, вызванные такими проблемами, необходимо предпринять предупреждающие действия еще до того, как они возникли. К предупреждающим действиям могут относиться:

- тренинги по взаимозаменяемости, целью которых является снижение проблем, возникающих в случае отсутствия некоторых членов команды,

- дополнительное разъяснение отдельных ролей, чтобы убедиться, что выполняются все должностные обязанности,

- выделение дополнительного времени отдельным сотрудникам на случай возникновения необходимости сверхурочной работы в ближайшем будущем для того, чтобы выполнить требуемый объем работ в назначенные сроки.

**2. Подходы к формированию команды ИT-проекта: подход Центра объектно-ориентированной технологии компании IBM. Команда ХР проекта. Проектная группа: подход MSF**

управление человеческий планирование организационный

По результатам множества опросов менеджеров проектов в России и за рубежом, до 80% успеха при реализации проектов обусловлено слаженной работой проектной команды, которая, в свою очередь, обеспечивается верным распределением ролей среди участников.

Поэтому многие руководители IT-проектов стремятся максимально ограничить число людей, работающих над системой, тем более что при таком подходе проще обеспечить концептуальное единство проекта и выработать согласованную позицию участников относительно стратегии его реализации. С другой стороны, концепция маленькой энергичной группы имеет один существенный недостаток - это слишком медленно для действительно больших систем (есть риск, что растянув процесс разработки программной системы на годы, вы создадите продукт, потребность в котором давно уже отпала).

В ходе развития проекта командой выполняются те или иные функции. Распределение функций между исполнителями является ничем иным как распределением ролей в команде, выполняющей проект. Значимость ролей различается в зависимости от того, какой проект выполняет IT-команда.

1. **Подход Центра объектно-ориентированной технологии компании IBM (Функциональные роли в коллективе разработчиков)**

В качестве достаточно полного перечня ролей можно указать на список, предлагаемый в рамках подхода Центра объектно-ориентированной технологии фирмы IBM:

1. Заказчик -- реально существующий (в организации, которой подчинена команда, или вне ее) инициатор разработки или кто-либо иной, уполномоченный принимать результаты (как текущие, так и окончательные) разработки;

2. Планировщик ресурсов -- выдвигает и координирует требования к проектам в организации, осуществляющей данную разработку, а также развивает и направляет план выполнения проекта с точки зрения организации;

3. Менеджер проекта -- отвечает за развитие проекта в целом, несет ответственность за распределение заданий и ресурсов, за соответствие результатов установленным требованиям. В рамках этих функций менеджер проекта взаимодействует с заказчиком и планировщиком ресурсов;

4. Руководитель команды -- производит техническое руководство командой в процессе выполнения проекта. Для больших проектов возможно привлечение нескольких руководителей подкоманд, отвечающих за решение частных задач;

5. Архитектор -- отвечает за проектирование архитектуры системы, согласовывает развитие работ, связанных с проектом;

6. Проектировщик подсистемы -- отвечает за проектирование подсистемы или категории классов, определяет реализацию и интерфейсы с другими подсистемами;

7. Эксперт предметной области -- изучает сферу приложения, поддерживает направленность проекта на решение задач данной области;

8. Разработчик -- реализует проектируемые компоненты, владеет и создает специфичные классы и методы, осуществляет кодирование и автономное тестирование, строит продукт. Это широкий термин, который может подразделяться на более узкие роли (например, разработчик классов). В зависимости от сложности проекта команда может включать различное число разработчиков;

9. Разработчик информационной поддержки -- создает документацию, сопровождающую продукт, когда выпускается версия. Включаемые в нее инсталляционные материалы, равно как ссылочные и учебные, а также материалы помощи предоставляются на бумажных и машинных носителях. Для сложных проектов возможно распределение этих задач между несколькими разработчиками информационной поддержки;

10. Специалист по пользовательскому интерфейсу -- отвечает за удобство применения системы. Работает с заказчиком, чтобы удостовериться, что пользовательский интерфейс удовлетворяет требованиям;

11. Тестировщик -- проверяет функциональность, качество и эффективность продукта. Строит и исполняет тесты для каждой фазы развития проекта;

12. Библиотекарь -- отвечает за создание и ведение общей библиотеки проекта, которая содержит все проектные рабочие продукты. Он также отвечает за соответствие рабочих продуктов стандартам.

Отдельные роли в проекте могут совмещаться и, наоборот, одна и та же роль может быть разделена между несколькими исполнителями.

**Желательное или часто встречающееся совмещение ролей**:

1. менеджер проекта + архитектор

2. руководитель команды + архитектор

3. руководитель команды + менеджер

4. разработчик + разработчик (с различными функциональными направлениями)

5. библиотекарь + разработчик

6. специалист по пользовательскому интерфейсу + менеджер

7. разработчик + тестировщик (при условии «перекрестного» совмещения ролей, когда разработчики независимых компонентов проекта тестируют функциональность друг друга)

**Нежелательное или невозможное совмещение ролей**:

1. заказчик + планировщик

2. руководитель команды + проектировщик

3. менеджер проекта + разработчик

4. разработчик + тестировщик

**2. Команда ХР проекта - роли для людей**

В экстремальном программировании чётко описаны все роли. Каждая роль предусматривает характерный набор прав и обязанностей. Здесь существуют две ключевые роли: заказчик и разработчик.

**Заказчик -**человек или группа людей, заинтересованных в создании конкретного программного продукта.

**Разработчик -**один или группа от 2 до 10 человек, занимающихся непосредственно программированием и сопутствующими инженерными вопросами. Разработчик наделён следующими правами и обязанностями:

Каждая из базовых ролей экстремального программирования может быть уточнена более мелкими ролями. В ХР разрешено совмещение ролей в рамках одного человека.

**Сторона заказчика**

1. Составитель историй - специалист предметной области, обладающий способностями доступно изложить и описать требования к разрабатываемой системе. Этот человек или группа людей ответственны за написание историй пользователя и прояснения недопонимания со стороны программистов.

2. Приёмщик - человек, контролирующий правильность функционирования системы. Хорошо владеет предметной областью. В обязанности входит написание приёмочных тестов.

3. Большой босс - следит за работой всех звеньев, от разработчиков до конечных пользователей. Он контролирует внедрение системы и сопутствующие организационные моменты. Может быть также инвестором проекта.

**Сторона разработчика**

1. Программист - человек, занимающийся кодированием и проектированием на низком уровне. Он достаточно компетентен для решения текущих задач разработки и предоставления правдивых оценок запланированных задач. Но при этом он должен обладать умением работать в паре, привычкой к простоте, умением и желанием постоянно переделывать код и рассматривать систему как общую собственность, которая принадлежит всей команде.

2. Тестер - помогает заказчику в подборе и написании функциональных тестов, отвечает за регулярный их запуск и оповещает команду о результатах тестирования. Кроме того, он следит за правильностью работы тестирующих инструментов.

3. Инструктор - отвечает за весь процесс разработки. Это опытный разработчик, хорошо владеющий всем процессом разработки и его методиками. Несёт ответственность за обучение команды аспектам процесса разработки. Внедряет и контролирует правильность выполнения методик используемого процесса. Обращает внимание команды на важные, но по каким-то причинам упущенные моменты разработки. Вмешательство инструктора в процесс разработки должно быть по-возможности минимальным.

4. Наблюдатель (ревизор) - член команды разработчиков, пользующийся доверием всей группы (совесть команды), который следит за прогрессом разработки. Он сравнивает предварительные оценки трудозатрат и реально затраченные усилия, выводя количественные показатели работы команды, такие как средняя скорость и процентное соотношение выполненных и запланированных задач. Данная информация предоставляется заказчику для своевременного контроля над ситуацией.

5. Дипломат - коммуникабельная личность, инициирующая общение между членами команды. Так как документооборот минимизирован, важно постоянное общение и передача опыта внутри команды, лучшее понимание требований к системе. Дипломат регулирует и упрощает общение между заказчиками и разработчиками. Является важным звеном на собраниях. Он препятствует недомолвкам, разгару страстей и ненужным ссорам. Дипломат не может навязывать своего мнения дискутирующим сторонам.

**3**. **Проектная группа: подход MSF**

При использовании подхода MSF все задачи, выполняемые в рамках проекта, распределяются по 6 ролям:

**1. Менеджер продукта.**

Цель этой роли -- удовлетворение требований заказчика. Его главная задача -- сформировать общее представление о поставленной задаче и о том, как ее решать. Как представитель заказчика, менеджер продукта отвечает за создание бизнес-сценариев, он помогает заказчику определить функциональные требования к программной системе и расставить приоритеты их реализации, проверяет, удовлетворяют ли решения группы потребностям заказчика.

**2. Менеджер программы**

Задача менеджера программы -- вести процесс разработки с учетом всех ограничений. Главная обязанность менеджера программы -- выполнить все стадии разработки так, чтобы нужный продукт был выпущен в нужное время. Кроме того, менеджер программы дает заказчику рекомендации, касающиеся дальнейшего развития продукта. Отдел программного менеджмента отвечает за функциональные возможности, изложенные в спецификациях (в котором описано, что будет создано) и в главном плане проекта (который определяет, как это будет сделано). Он также координирует работу над этими документами. Тем не менее каждый участник проекта вносит свой вклад в функциональные спецификации и в план проекта.

**Внимание!** Важно, чтобы менеджер программы понимал, что каждый член группы разбирается в своих обязанностях намного лучше его. Менеджер программы должен полностью положиться на опыт остальных сотрудников и только следить за соблюдением всех условий и ограничений.

В ходе реализации проекта менеджер программы контролирует реальные затраты, сравнивая их с запланированными. Он регулярно сообщает о состоянии работы всем основным участникам проекта.

Как правило, группа менеджмента программы управляет ресурсами, используемыми другими ролями, а также составляет и координирует расписания совещаний. Менеджер программы отвечает и за бюджет проекта, объединяя требования к ресурсам всех членов группы в единый план расходов. Важно, чтобы еще до начала проекта менеджер программы имел непререкаемый авторитет у всех его участников, включая бизнес-отделы и руководство организации.

**3. Разработчик**

Разработчики знакомят остальных членов группы с применяемыми технологиями и собственно создают продукт. В качестве консультантов они предоставляют исходные данные для проектирования, проводят оценку технологий, а также разрабатывают прототипы и тестовые системы, необходимые для проверки решений и сокращения рисков на ранних стадиях процесса разработки. Руководителю группы разработки не рекомендуется совмещать несколько ролей.

**4. Тестер**

Задача тестеров -- испытание продукта в реальных условиях, дабы определить, что в продукте работает и что не работает. Тестеры разрабатывают стратегию, планы, графики и сценарии тестирования, которые позволяют убедиться, что все ошибки выявлены и исправлены до выпуска приложения. Ошибкой называют любую проблему, из-за которой продукт не выполняет свои функции.

Нельзя совмещать должности тестера и разработчика. Разделение этих обязанностей:

- гарантирует независимую проверку того, что продукт действительно выполняет все требования;

- повышает качество продукта за счет конкуренции между группами.

**5. Инструктор**

Цель группы обучения -- повысить эффективность труда пользователей. Группа обучения отвечает за выпуск удобного, полезного продукта, которому практически не нужна поддержка.

**6. Логистик**

Логистик представляет интересы служб поддержки и сопровождения, справочных служб и других служб канала доставки. Он занимается развертыванием продукта и его сопровождением и контролирует продукт с этой точки зрения в процессе проектирования. Логистик руководит всеми сотрудниками, устанавливающими и настраивающими пользовательские системы. Кроме того, он должен уметь координировать установку программного обеспечения и оценивать ее результаты.

После развертывания логистики в течение некоторого времени консультируют группу сопровождения. Конечно, сложно предсказать все трудности, которые могут возникнуть в процессе эксплуатации приложения, поэтому во всех планах следует предусматривать и порядок действий в экстренном случае.

6 ролей модели проектной группы связаны с 6 целями проектной группы (табл. 2). Все цели важны для успеха проекта в целом, и поэтому все роли равноправны. В этой модели нет руководителя всего проекта -- есть группа людей, знающих, что нужно делать, и делающих это.

**Табл. 2**. Цели и роли

|  |  |
| --- | --- |
| **Цель** | **Роль** |
| Удовлетворение требований заказчика | Менеджер продукта |
| Соблюдение ограничений проекта | Менеджер программы |
| Соответствие спецификациям | Разработчик |
| Выпуск только после выявления и устранения проблем | Тестер |
| Повышение эффективности труда пользователя | Инструктор |
| Простота развертывания и постоянное сопровождение | Логистик |

Обычно каждую роль исполняют несколько человек, один из которых считается руководителем.

**3. Организационная схема управления IT-проектом**

**Под организационной структурой понимается совокупность элементов организации (должностей и структурных подразделений) и связей между ними.**

К жестким механистическим структурам относится линейно-функциональная структура.

**Линейно-функциональная ОСУ** является наиболее часто встречающаяся и традиционной, характеризуется обособлением отдельных функций (финансы, маркетинг, сбыт и др.) и назначением на каждую из них линейного руководителя. В такой структуре менеджер проекта не обладает большими полномочиями, осуществляет лишь общую координацию работ и подотчетен функциональному руководителю. Чаще его называют координатором проекта.

К **адаптивным ОСУ** относятся проектная и матричная. Они позволяют оперативно реагировать на изменения внешней среды, внедрять новые технологии, корректировать требования к конечному результату в зависимости от условий реализации проекта.

**Проектная ОСУ** - это временная структура, создаваемая для решения конкретной задачи. Ее цель состоит в том, чтобы собрать в одну команду самых квалифицированных сотрудников организации для осуществления конкретного проекта в установленные сроки с заданным уровнем качества, не выходя за пределы установленной сметы. Когда проект завершен, команда распускается. Ее члены переходят в новый проект, возвращаются к своей основной работе или уходят из организации. Проектно-целевая структура возникает в случае, когда вся деятельность организации концентрируется на выполнении определенного проекта (или программы как совокупности проектов) достижении определенной цели. При этом все другие структурные образования либо отсутствуют, либо имеют вспомогательное значение - штабные и обслуживающие подразделения, комитеты и пр.

**Матричная структура** получается при наложении проектной структуры на функциональную. В матричной организации члены проектной группы подчиняются как руководителю проекта, так и руководителю того функционального подразделения, в котором они работают постоянно. Руководитель проекта обладает так называемыми проектными полномочиями, объем которых варьируется в зависимости от тех прав, которые делегирует ему высшее руководство компании.

Руководитель проекта в матричной организации отвечает в целом за интеграцию всех видов деятельности и ресурсов, относящихся к данному проекту. К его компетенции относится также планирование проекта, контроль за соблюдением графика, оценка количественных, качественных и временных показателей. Руководители функциональных отделов решают, где и как должна быть выполнена та или иная работа, контролируют ход выполнения задач.

**Разновидности матричных оргструктур:**

**1. Слабая матрица,** больше похожая на функциональную структуру. Проект в этом случае может иметь только одного постоянного сотрудника -- координатора проекта. Он отвечает за координацию задач по проекту, но имеет ограниченную власть над ресурсами. Приоритет проектов, чаще всего, ниже текущих функциональных задач предприятия.

**2. Сбалансированная матрица** -- менеджер проекта координирует все работы и разделяет ответственность за достижение цели с руководителями функциональных подразделений. При существовании нескольких проектов в компании часть из них может управляться по схеме: «приоритет проекта ниже приоритета функциональных областей», и это характерно для проектов развития или коммерчески затратных проектов; другая часть - по противоположной схеме: «приоритет проекта выше приоритета функциональных областей», что присуще инвестиционным или коммерчески привлекательным проектам.

**3. Сильная (жесткая) матричная структура** характеризуется тем, что руководитель проекта имеет большие права и полномочия по управлению проектом; в проекты привлекается от 50 до 95% всех организационных ресурсов предприятия. Деятельность по проекту имеет явный приоритет над функциональной.Преимущества и недостатки ОСУ:

|  |  |
| --- | --- |
| **Преимущества** | **Недостатки** |
| **Функциональная ОСУ** |  |
| 1. возможность гибкого формирования команды  2. отсутствует дублирование функций | 1. слабая мотивация  2. трудности координации ипроблемы с результатами;  3. медленные коммуникации и реакция на требования клиента;  4. проект не управляется как единое целое;  5. нет ответственности за конечные продукты |
| **Матричная ОСУ** |  |
| 1. гибкость  2. снижается дублирование;  3. существует баланс ресурсов при осуществлении нескольких проектов; | 1. сложность  2. дорога в использовании;  3. невозможна без качественной корпоративной информационной системы  4. нарушается принцип единоначалия и формируется двойной фокус принятия решений;  5. присутствует конфликт приоритетов. |
| **Проектная ОСУ** |  |
| 1. концентрирует все усилия на достижении целей одного конкретного проекта.  2. прямая ответственность за результат;  3. короткие линии коммуникации, высокая скорость коммуникаций;  4. высокая заинтересованность в проекте;  5. мотивация членов команды высока и способствует нацеленности на выполнение задачи;  6. четко осуществляются контроль и коммуникации | 1. дублирование функций;  2. нерациональный объем запасов (в сторону увеличения);  3. противоречивость в исполнении процедур и правил;  4. внутренняя конкуренция между проектами |
|  | |

Рис. 3. иллюстрирует три возможные схемы организации менеджмента проекта. Здесь стрелки обозначают связи участников реализации проекта, обусловленные их взаимодействием с менеджером, жирность контура значков отражает менеджерские обязанности сотрудников.

В небольшом проекте все нити управления, как правило, концентрируются в руках одного человека - менеджера проекта, что позволяет исключить проблемы внутреннего согласования противоречий, обеспечивает централизованную ответственность за проект перед заказчиком (этот вариант иллюстрируется схемой с одним менеджером).

Однако по мере перехода к более масштабным проектам менеджерские функции становится все сложнее удерживать в одних руках. Один из возможных способов решения проблемы -- это образование службы менеджера, состоящей из его помощников, которым он может поручать различные задания, освобождая себя от рутины постоянного контроля.

Третий вариант, который применяется в еще более сложных проектах, - это образование группы менеджеров, ответственных за разные разграниченные сферы проекта.

Как видно из рисунка, в схеме со службой менеджера помощники по своему статусу являются обычными работниками, тогда как при делегировании полномочий менеджеры по направлениям получают соответствующие полномочия в своих сферах ответственности (обозначены овалами).

Для небольших групп возможна следующая организация работ: обязанности главного менеджера распределяются по команде разработчиков, которая за счет внутренних механизмов решает, как планировать работы, как их распределять и контролировать. Такой подход характерен для «легких» методологий разработки программного обеспечения, отличительным признаком которых является отказ от многих принципов традиционного менеджмента. Наиболее ярким примером такого подхода является экстремальное программирование (ХР).

**4. Эффективность команды проекта**

Различают эффективность с позиций профессиональной деятельности по проекту и организационно-психологического климата деятельности.

В **профессиональном отношении эффективность** -- это прежде всего нацеленность всей команды на конечный результат, инициатива и творческий подход к решению задач. Высокая производительность и ориентированность на лучший вариант решения, активное и заинтересованное обсуждение возникающих проблем дополняют ее характеристику.

**С позиций организационно-психологического климата эффективной** можно назвать такую команду, в которой:

- неформальная атмосфера;

- задача хорошо понята и принимается;

- члены команды прислушиваются друг к другу;

- задачи обсуждаются коллективно с участием всех членов команды;

- все члены команды свободно выражают как свои идеи, так и чувства;

- конфликты и разногласия присутствуют, но выражаются и центрируются вокруг идей и методов, а не личностей;

- группа осознает, что делает, решение основывается на согласии, а не на голосовании большинства.

При соблюдении таких условий команда не только успешно выполняет свою миссию, но и удовлетворяет личные и межличностные потребности своих членов

В общем случае, *человеческие ресурсы проекта* − это совокупность профессиональных, деловых, личностных качеств участников проекта и членов команды проекта и их возможностей (влияния, «веса», связей и т.п.), которые могут быть использованы при осуществлении проекта. Трудовые ресурсы являются частью человеческих ресурсов, рассматриваемых как измеряемый ресурс в проекте.

*Персонал*− это конкретные индивидуумы, частью которых является их квалификация, исполнение функционально-должностных обязанностей и прочее, что описывается в рамках штатного расписания проекта.

Когда говорим о современном менеджменте человеческими ресурсами и персоналом, то имеем в виду управление не измеряемыми величинами.

Планировать можно то, что поддается измерению. В рамках организационного планирования проекта проводится расчет требующихся для его осуществления трудовых ресурсов.

В этом случае, определяются и планируются сроки и длительность использования (загрузка, трудозатраты) менеджеров и специалистов.

Также определяется стоимость привлекаемых трудовых ресурсов, исходя из их квалификации, потребностей и возможностей проекта, видов работ (пакетов работ), соответствующей рыночной или нормативной стоимости их работ (услуг) и прочее. Производится также назначение персонала на определенные работы (пакеты работ) и/или направления работ.

Во всех специализированных программных продуктах (ПП) по управлению проектами используются блоки по управлению штатом и трудовыми ресурсами с хорошими коммуникационными возможностями для совместной работы персонала по проекту.

К достаточно распространенным и поддерживаемым на российском рынке ПП относятся как «большие» мультипроектные профессиональные системы − PrimaVera Planner 3.0, Open Plan Professional, так и менее «большие» − Microsoft Project 2000, Sure Track Manager 2.0. В рамках подсистем по управлению персоналом в этих ПП имеются хорошие возможности для планирования и использования привлекаемых к проекту специалистов, как его измеряемого ресурса. Но этого недостаточно.

Для проекта важно понимание границ применимости программных продуктов в области управления человеческими ресурсами: там, где речь идет об «управлении человеком», как измеряемым ресурсом, то использование специализированных программных продуктов по управлению проектами достаточно.

В проектах, в которых человеческий фактор имеет решающее значение, ориентация только на управление «трудовыми ресурсами» и «штатом» без учета организационной и профессиональной культур, индивидуальных особенностей членов команд и других плохо идентифицируемых и измеряемых характеристик команд, часто приводит к конфликтам, трудностям «на ровном месте» и неуспеху всего проекта.