**Модель группы и иерархическая модель**

**Существует две основные модели организации коллектива при разработке ПО:**

**1) иерархическая модель**

**2) модель группы**

Работа в коллективе отличается определенными сложностями. Прежде всего, коллектив хочет знать «кто начальник». Ответ на этот вопрос дает иерархическая модель организации, определяющая начальников и подчиненных. Однако, если в современных производственных средах один менеджер проекта отвечает за все тонкости разработки и принимает все важные решения, возникает множество проблем, ведущих к провалу проекта. **Иерархическая модель грешит множеством недостатков:**

·    нехваткой информации;

·    невозможностью учесть все особенности проекта;

·    отсутствием полноценной связи между всеми участниками проекта, так как вся информация идет в одном направлении — вверх по иерархии, к главному менеджеру;

·    трудностью освоения новых технологий, необходимых при создании кроссплатформенных приложений;

·    сложностью расстановки приоритетов.

Кроме того, опыта одного человека чаще всего недостаточно для быстрого решения задачи и для интеграции приложения в существующуюинфраструктуру.

В организациях, построенных на основе иерархической модели, затруднен обмен информацией — в этой модели он, по определению, осуществляется через посредников. Вся информация иерархических групп «фильтруется» тремя или четырьмя менеджерами, что значительно повышает вероятность утери самого важного. Часто такое отсеивание идей происходит при прохождении сообщения от разработ­чика, непосредственно занимающегося проектом, к высшему руководству. Естественно, некоторые участники «выпадают» из процесса, что снижает эффективность их труда и повышает вероятность провала проекта.

Дабы сгладить недостатки иерархической модели, в проектной группе предусматривается распределение обязанностей руководителя между членами коллектива. При этом за проект отвечает не один человек, а все члены группы — каждый за свой участок.

Модель группы не определяет структуру коллектива с точки зрения отдела кадров. Ведь в такую разностороннюю группу привлечены ресурсы из разных отделов организации. Задача модели проектной группы — определить цели проекта и распределить обязанности. Руководители каждого направления с помощью выделенных им ресурсов выполняют возложенную на них часть работы. Обязанности ролей определяются работой над проектом, а не деятельностью «штатной единицы». При этом руководители направлений выполняют свои обычные функции: составляют график выплаты премий, распределяют отпуска и контролируют эффективность работы сотрудников. Начальник может оценить степень участия и эффективность работы сотрудников в проектной группе, но это — прерогатива менеджера конкретного сотрудника, а не проектной группы.

**И у коллективного подхода есть недостатки:**

·    разрозненная связь с внешними источниками информации;

·    несогласованное представление о разных сторонах проекта;

·    несогласованность личных планов членов группы;

·    отсутствие опыта, снижающее эффективность коллективной работы.

**Обязанности членов группы**

MSF(модель проектной группы) — не готовое решение, а каркас, который можно адаптировать для нужд любой организации. Один из элементов этого каркаса - *модель проектной группы.*Она описывает структуру группы и принципы, которым надо следовать для успешного выполнения проекта.

Хотя модель группы разработчиков весьма конкретна, при знаков знакомстве с MSF ее нужно рассматривать в качестве отправной точки. Разные коллективы реализуют этот каркас по-разному, в зависимости от масштаба проекта, размеров группы и уровня подготовки ее членов.

Чтобы проект считался удачным, следует решить определенные  
задачи:           I

·  **удовлетворить требования заказчика**— проект должен выполнить  
требования заказчиков и пользователей, иначе ни о каком успехе  
не может быть и речи, возможна ситуация, когда бюджет и график  
соблюдены, но проект провалился, так как не выполнены требования заказчика;

·  **соблюсти ограничения**— разработчики проекта должны уложиться  
в финансовые и временные рамки;

·  **выполнить спецификации, основанные на требованиях пользователей**— спецификации — это подробное описание продукта, создаваемое группой для заказчика; они представляют собой соглашение между проектной группой и клиентом и регулируют вопросы, касающиеся приложения, в основе этого требования лежит принцип «сделать все, что обещано»;

·  **выпустить продукт только после выявления и устранения всех проблем**— не существует программ без дефектов, однако группа должна найти и устранить их до выпуска продукта в свет, причем устранением ошибки считается не только ее исправление, но и, например, занесение в документацию способа ее обхода; даже такой способ устранения проблем предпочтительнее, чем выпуск приложения, содержащего невыявленные ошибки, которые в любой момент могут преподнести неприятный сюрприз пользователям и разработчикам;

·  **повысить эффективность труда пользователей**— новый продукт должен упрощать работу пользователей и делать ее более эффективной. Поэтому приложение, обладающее массой возможностей применять которые сложно или неудобно, считается провальным;

·  **гарантировать простоту развертывания и управления —**эффективность развертывания непосредственно влияет на оценку пользователем качества продукта, например, ошибка в программе установки может создать у пользователей впечатление, что и само приложение небезгрешно, от проектной группы требуется не только подготовить продукт к развертыванию и гладко провести его, но и обеспечить пользователей поддержкой, организовав сопровождение приложения.

*Примечание. На проект влияет множество факторов, некоторые из которых создают дополнительные ограничения. Когда проектная группа рассматривает на совещаниях цели проекта и график его создания и выпуска, следует убедиться, что проект окупится, то есть удастся выпустить нужный продукт, не превышая запланированных расходов.*

Для достижения этих целей в модели проектной группы выполняемые задачи распределяются по шести ролям: менеджмент продукта, менеджмент программы, разработка, тестирование, обучение пользователей и логистика. Люди, выполняющие конкретную роль, должны рассматривать проект со своей «колокольни» и обладать необходимой для этого квалификацией.

Шесть ролей модели проектной группы, подробно описанные в этой главе, связаны с шестью целями проектной группы, что проиллюстрировано в таблице. Все эти 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.

**Таблица. Цели и роли**

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

Как же начать работу над проектом, не зная, сколько времени на это потребуется, сколько проект будет стоить и каких результатов ожидать? Ответить на эти вопросы поможет модель проектной группы MSF, обсуждаемая в следующих разделах этой главы, и модель процесса разработки, описанная в главе 4. Обе эти модели предлагают составлять расписания и графики «снизу — вверх», в проектировании придерживаться подхода «сверху — вниз» и, кроме этого, распределять ответственность за результаты работы между членами группы. На первый взгляд может показаться, что эти рекомендации не отвечают на поставленный вопрос. Однако на самом деле они позволяют сократить риски на ранних стадиях проекта. Приняв на этих стадиях правильное решение, вы избежите серьезных изменений в дальнейшем, что намного сократит время выполнения проекта и затраты на него.

**Модель проектной группы**

Работа над проектом включает множество разных видов деятельности и изучение требований с разных точек зрения, поэтому распределение основных задач по нескольким ролям повышает шансы на успех проекта. Как видно из рисунка 8.1, в MSF определены шесть таких ролей, которые и составляют модель проектной группы. У каждой из ролей свои обязанности, выполнение которых и обуславливает удачу проекта.

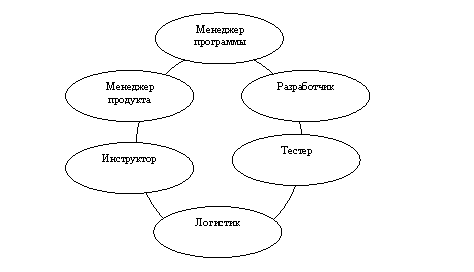


Рис. 8.1 Роли в модели проектной группы

Обычно каждую роль исполняют несколько человек, один из которых считается руководителем, именно он на совещаниях с другими членами группы представляет свое направление.

*Внимание! Коллективная ответственность не должна выливаться в коллективную безответственность. Это особенно важно в модели проектной группы, поскольку выполнение каждой ролью всех своих обязанностей — обязательное условие успеха проекта.*

**В проектную группу должны входить**:

·  опытные руководители;

·  инициативные сотрудники, способные принимать решения и нести ответственность за свое направление работы.

**Их задача:**

·    сконцентрироваться на выпуске продукта;

·    выработать общее представление о проекте.

Для эффективной работы проектной группы требуется соблюдение следующих правил в отношениях между людьми:

·             **доверие**— делает действия людей согласованными, при этом все

·             следуют принципу «мы делаем то, что обещали сделать»;

·             **уважение**— люди признают способности других, следуя правилу

·             «каждый из нас необходим нашей группе»;

·             **согласие**— все должны знать и поддерживать цели проекта и верить в его успех — «мы завершим проект, и точка»;

·             **ответственность**— люди должны ясно понимать цели проекта, свои обязанности и чего от них ожидают — «я сделаю свою работу, вы — свою, и к четвергу мы построим наш дом».

Подытожив эти характеристики, мы получаем главный принцип — обязанность каждого — выпустить нужный продукт в нужное время».

**Менеджер продукта**

Менеджер продукта должен вовремя реагировать на потребности заказчика. Его главная задача — сформировать общее представление о поставленной задаче и о том, как ее решать. Он должен ответить на вопрос «Зачем мы делаем все это?» и убедиться, что все члены группы знают и понимают ответ на него.

Основная цель этой роли — удовлетворение требований заказчика. Для этого менеджер продукта выступает представителем заказчика в группе разработчиков и представителем группы у заказчика. (На этом этапе важно понимать разницу между заказчиком и пользователем: заказчик платит за создание продукта, а пользователи с ним работают.) Кроме того, менеджер продукта вместе с менеджером про­граммы должны прийти к компромиссному решению относительно функциональных возможностей продукта, сроков его разработки и финансирования проекта.

Как представитель заказчика, менеджер продукта отвечает за выполнение требований заказчика, создание бизнес-сценариев, формирование общего представления о проекте у группы и клиента, а также проверяет, удовлетворяют ли решения группы потребностям заказчика.

Как представитель группы, менеджер продукта отвечает за взаимодействие с заказчиком и управление его ожиданиями. При этом он проводит брифинги с клиентом и главными менеджерами, устраивает встречи с пользователями и демонстрации продукта.

Менеджер продукта сообщает группе предварительную дату выпуска продукта, устанавливаемую заказчиком. Однако в действительно успешных проектах графики составляются «снизу — вверх». При этом группа сама определяет возможную дату выхода приложения, которую менеджер продукта сообщает клиенту.

*Примечание Заказчик (лицо, финансирующее проект) всегда ожидает от проекта чего-то большего. Естественно, ему хочется, чтобы все его пожелания воплотились в жизнь. И здесь основная роль отводится менеджеру продукта, следящему за ожиданиями клиента. Таким образом, работа менеджера часто определяет успех или провал проекта.*

Менеджер продукта должен постоянно общаться с заказчиком, выясняя: «Мы сделали то, что обещали?» Допустим, в рамках проекта предполагалось реализовать пять операций за две недели и 10 000 долларов. Однако за полторы недели было потрачено 7 000 долларов, реализованы только три функции. Как вы думаете, покажется ли этот  
проект заказчику успешным? Однако квалифицированный менеджер  
продукта мог бы ответить: «Был проведен второй этап переговоров, и мы договорились с заказчиком относительно самых важных операций, которые можно реализовать, не превышая финансирование и не выходя из графика. То, что мы обещали, мы выполнили. Сейчас же мы начинаем новый проект, в рамках которого реализуем две оставшиеся  
функции и еще две дополнительные». Таким образом, менеджер продукта пришел с заказчиком к компромиссу, продемонстрировав процесс принятия решений и проинформировав клиента об изменившихся рисках и проблемах. (Обычно трудности возникают при определении затрат на проект, даты его выполнения и функциональных возможностей, а также при выделении ресурсов.) Если все участники проекта смогут сказать: «Да, мы сделали то, что обещали», между заказчиком и проектной группой воцарится доверие и уважение.

*Примечание Основная причина неудачи многих проектов — непонимание ожиданий заказчика. При любых — запланированных или нет — изменениях проекта, его ресурсов или даты выхода продукта ожидания заказчика, пользователей и проектной группы должны быть скорректированы.*

Группа менеджмента продукта представляет интересы заказчика и помогает ему определить необходимые функции приложения и их приоритеты. Отложить работу над какой-либо функцией вправе только заказчик, а переговоры по этим вопросам ведет менеджер продукта. За выполнение и составление графиков отвечает менеджер программы, поэтому в его власти изъять определенные функциональные возможности, чтобы текущая версия появилась в срок. Если менеджерпрограммы решит сократить набор функциональных возможностей, чтобы не выбиться из графика, он обязан сообщить об этом менеджеру продукта и заказчику, которые вправе согласиться или нет. В последнем случае заказчик и менеджер продукта должны согласовать изменение даты выпуска продукта. Кроме того, они могут увеличить финансирование с целью расширения состава группы. Затем менеджер программы определяет, как и на что потратить дополнительные деньги и поможет ли это выполнить план. Окончательное решение о финансировании принимают заказчик и менеджер продукта.

***Внимание!****Дополнительные ресурсы не всегда ускоряют разработку, так как возрастают издержки, связанные с обменом информацией и управлением.*

Руководителя группы менеджмента часто называют «борцом за продукт». Как правило, это один из топ-менеджеров организации. Остальные члены группы менеджмента продукта должны хорошо разбираться в структуре организации, ее стратегии и бизнес-целях.

Примечание Если проектная группа состоит из консультантов или приглашенных из других организаций людей, менеджер продукта должен учитывать интересы как заказчика, финансирующего проект, так и заказчика, нанявшего консультантов и оплачивающего их услуги. Как правило, консультационная группа требуется для ознакомления организации с MSF и обеспечения взаимодействия заказчика и проектной группы.

**Менеджер программы**

Задача менеджера программы — вести процесс разработки с учетом всех ограничений. Руководитель этого направления должен понимать разницу между понятиями «руководитель» и «начальник». В своей книге «Dynamics of Software Development» бывший директор отдела программного менеджмента Microsoft Джим Маккарти пишет:

*«Запомните, что ваша цель — повысить авторитет каждого члена группы, а не подавить его своим».*

Главная обязанность менеджера программы — выполнить все стадии разработки так, чтобы нужный продукт был выпущен в нужное время. Он координирует деятельность других членов группы. И хотя иногда ему придется подгонять своих сотрудников, он не должен и помышлять о диктаторском стиле управления.

Главный менеджер программы составляет график проекта на основе информации, полученной от остальных членов группы. Он координирует этот график с руководителями всех подгрупп. Буферным временем проекта также управляет менеджер программы. Если отдельные части работы выполняются раньше графика или, наоборот, задерживаются, именно менеджер программы должен выяснить, как это скажется на проекте, и изменить график.

Для выполнения своих обязанностей менеджер программы должен отлично разбираться в деловой стороне проекта и иметь ясное представление о технологиях, необходимых для его выполнения. Естественно, что от руководителя отдела программного менеджмента требуется коммуникабельность и талант организатора.

Программный менеджер компании, где только начинают применять MSF, должен полностью понимать все модели и процессы, повышающие эффективность труда членов группы. И хотя умение руководить нужно всем ролям, особенно это качество важно для менеджера программы. Его авторитет должен стать непререкаемым еще до начала проекта у всех его участников, включая бизнес-отделы и руководство организации. Как правило, группа менеджмента программы управляет ресурсами, используемыми другими ролями, а также составляет и координирует расписания совещаний.

Так как менеджер программы отвечает за набор функциональных возможностей приложения, одна из его обязанностей — определить их набор, необходимый для выполнения требований заказчика. Критичные для проекта требования выявляет менеджер продукта, но набор реализующих их функций определяет менеджер программы. Кроме того, менеджер программы дает заказчику рекомендации, касающиеся дальнейшего развития продукта. Отдел программного менеджмента отвечает за функциональные возможности, изложенные в спецификациях (здесь описано, что будет создано) и в главном плане проекта (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 следите за соблюдением всех условий и ограничений.*

Менеджер программы отвечает и за бюджет проекта, объединяя требования к ресурсам всех членов группы в единый план расходов. Естественно, его задача — не только разобраться в этих требованиях, но и контролировать реальные затраты, сравнивая их с запланированными. Кроме того, менеджер программы должен регулярно сооб­щать о состоянии работы всем основным участникам проекта. Ведь одним из симптомов неудачного продукта является отсутствие информации о состоянии проекта, пока деньги на него не кончились. Важно внимать, что решение о дополнительном финансировании может требовать изменения других сторон проекта, а следовательно, его будет принимать менеджер программы совместно с заказчиком.

**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. Чтобы создать продукт определенного качества, разработчикам не следует замыкаться на создании кода, они должны участвовать и в решении прикладной задачи. Они творят не ради творчества, а для реализации требований за­казчика. Часто, чтобы полностью разобраться в проекте, приходится создавать прототипы, а чтобы протестировать новую технологию, — испытательные системы, помогающие принять окончательное решение относительно архитектуры приложения. Этим также занимаются разработчики.

Как программисты разработчики отвечают за низкоуровневое проектирование и оценку затрат на реализацию продукта. В большинстве организаций несколько основных разработчиков занимаются и архитектурой приложения. Как правило, это требуется на ранних ста­диях проекта, когда уточняются детали функциональных спецификаций и описывается взаимодействие продукта с внешними системами.

Разработчики сами оценивают сроки своей работы. Такая концепция MSF — создание графиков ответственными за выполнение конкретного участка членами группы — называется *составлением расписания* *«снизу — вверх».*Она позволяет выпустить нужный продукт в нужное время за счет уточнения графиков и повышения ответственности за выполнение работы в запланированные сроки.

Разработчики отвечают и за техническую реализацию проекта — в основном на фазах создания логической и физической модели, обсуждавшейся в главе 2. На этих стадиях их задача — определить методы реализации функциональных возможностей и заданной архитектуры, атакже оценить сроки выполнения этой работы. Заметим, что разработчики не выбирают функции — они только решают, как их реализовать.

Кроме того, на стадии «Планирование» разработчики решают, какое влияние окажет на проект добавление или удаление некоторых функций. Разработчики не участвуют в заключительной стадии проекта — развертывании продукта, однако они должны тесно сотрудничать с логистиками на стадии установки приложения.

*Внимание! Руководителю группы разработки не рекомендуется совмещать несколько ролей. Будучи программистом, он должен делать то, что у него лучше всего получается — писать качественный код.*

**Тестер**

Задача тестеров — испытание продукта в реальных условиях, дабы  
определить, что в продукте работает и что не работает, и нарисовать таким образом точный «портрет» приложения. Естественно, для проведения тестов нужно отлично разбираться и в требованиях пользователей, и в том, как их удовлетворить.

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. Ею может оказаться и ошибка в коде, называемая «жучком», и отклонение от спецификаций, заданных менеджером программы, и недоработки в документации, подготовленной группой обучения пользователей.

*Примечание Важно различать тестирование и общую оценку качества, (Total Quality Assurance, TQA). Тестирование касается только проек­та — точнее, его технической стороны. Проверку качества организует руководитель, ответственный за качество, — он выясняет соответствие продукта корпоративным, правительственным и другим стандартам.*

**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.

Хотя проверяют качества продукта только тестеры, за выпуск хорошего продукта отвечают все члены проектной группы.

*Внимание! За качество кода отвечают разработчики. Отделение тестирования от разработки не снимает с них эту обязанность*.

**Контроль изменений**

При работе над проектом необходимо контролировать изменения, им должны заниматься все участники группы, но чаще всего в полном объеме этим приходится заниматься именно группе тестирования. Разработчики уже не один год применяют системы, подобные Microsoft Visual SourceSafe, но они осуществляют только контроль версий. Такие системы пригодны для отслеживания версий любых документов, функциональных спецификаций, исходного и скомпилированного кода, но они только частично регистрируют изменения, в качестве примера можно привести сохранение версий документа в Microsoft Word — эта функция не гарантирует, что вы внесете верные правки и представите в качестве результата правильную версию. Таким образом, для управления изменениями необходимо:

·  создать эталонный документ;

·  определить изменяемые элементы;

·  определить влияние изменений на существующие системы, •процессы или документы;

·  определить метод реализации изменений;

·  назначить человека, который внесет изменения;

·  определить влияние изменения на условия выполнения проекта, •его бюджет, график и политику;

·  получить одобрение изменений (скажем, у руководителя проекта);

·  внести изменения;

·  сообщать новый документ, в котором изменение учтено.

***Внимание!****Изменения лучше анализировать и принимать порциями, иначе на это уходит слишком много времени.*

**Прочие обязанности**

Некоторые важные обязанности тестеров часто упускают из виду. К ним относятся:

·  **уведомление об ошибках и их отслеживание**— тестовая группа отвечает не только за управление изменениями, но и за систему вывления ошибок и информирования о них;

·  **сборка продукта**— в группе должен быть человек, ответственный за сборку (компиляцию) продукта, и часто такой «главный сборщик» является тестером, он может использовать только код, xранящийся в системе управления версиями; эту рутинную работе  
удается автоматизировать с помощью сценариев, однако необходимо проверять правильность сборки;

·  **выявление и контроль рисков**— это обязанность всех членов группы, менеджер программы должен разработать метод контроля - например, с помощью электронных таблиц Microsoft Excel; тестеры отвечают за работу программы в реальных условиях, поэтому ил задача — представить группе анализ рисков с этой точки зрения.

*Внимание! План контроля рисков, созданный или пересмотренный за пять минут до начала совещания, совершенно бесполезен. Процесс контроля рисков должен быть непрерывным и постоянным — только тогда гарантируется качество продукта и соблюдение сроков выпуска*

**Инструктор**

Цель группы обучения — повысить эффективность труда пользователей. Поэтому инструкторы «принимают сторону» пользователей подобно тому, как менеджеры продукта представляют интересы заказчика. Однако перед пользователями инструкторы выступают в роли представителей проектной группы.

В этом последнем качестве группа обучения отвечает за выпуск удобного, полезного продукта, которому практически не нужна поддержка. Персонал группы тестирует удобство использования продукт та, выявляет проблемы в этой области и проверяет проект пользова­тельского интерфейса.

Активно участвуя в создании пользовательского интерфейса, инструкторы сокращают затраты на сопровождение продукта и поддержку пользователей. Часто же бывает, что изучению этих расходов практически не уделяется внимания, хотя все расчеты очень просты; чем легче работать с приложением, тем меньше затраты на поддержку.

*Внимание! Не попадайтесь в ловушку: не завышайте планируемый прирост эффективности труда пользователей — это приводит к переоценке прибыли от инвестиций в проект.*

Довольно часто приложение сдается в эксплуатацию без плана обучения, а проектные группы даже не имеют представления о том как пользователи будут изучать приложение. Задача инструкторов - не допустить этого, заранее спроектировав, составив и протестиро­вав все необходимые материалы, включая краткие памятки, руководства пользователей, системы, онлайновой помощи, Web-страницы и даже — если понадобится — целый учебный курс. Когда материалы подготовлены, группа координирует обучение пользователей.

Управлять ожиданиями пользователей так же важно, как и ожиданиями заказчика. Однако не следует думать, что пользователи будут разбираться в функциональных спецификациях. Они более благосклонно отнесутся к прототипам системы, демонстрация которых значительно повысит шансы на успех проекта. Кроме того, пользователям полезно показать действующие модули программы. Хотя маркетинг входит в обязанности менеджеров продукта, также важно вовремя информировать и пользователей. Для этого следует периодически рассылать электронные сообщения о новых функциях программы и ее бета-версиях.

*Примечание. Роль обучения важна и в случае разработки приложения для другой организации. Ведь при этом вам неизвестно, насколько хорошо ее сотрудники информируют пользователей. Так же, как отдел менеджмента продукта управляет ожиданиями заказчика, отдел обучения должен управлять ожиданиями пользователей. Настоящий успех дает продукт, если пользователи узнают о его возможностях не сразу, а постепенно. При этом важны обучение и подготовка.*

**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. Кроме того, его задача — составление графиков развертывания приложения. Логистики, менеджеры продукта и менеджеры программы совместно определяют порядок передачи продукта пользователям и организации, после чего логистики готовят их к развертыванию приложения.

Логистик, участвующий в крупном проекте, должен обладать опытом развертывания крупномасштабных приложений на нескольких сотнях компьютеров. Именно поэтому от него требуется коммуникабельность и хорошая техническая подготовка. Логистик руководит всеми сотрудниками, устанавливающими и настраивающими пользовательские системы. Кроме того, он должен уметь координировать установку программного обеспечения и оценивать ее результаты.

Логистик обязан разбираться в инфраструктуре продукта и требованиях к его сопровождению. Его задача — проверить, чтобы все серверы развертывания и рабочие станции пользователей удовлетворяли требованиям. Обычно данные вопросы учитываются в планах выпуска, развертывания и сопровождения продукта. Заметим, что  
развертывание приложений значительно упрощается при использовании таких средств Microsoft Windows 2000, как Active Directory и System Management Server.

Хотя основная задача логистика заключается в плавном развертывании продукта, обязанность руководителя этого направления — составить план сопровождения и эксплуатации и убедиться, что существующие группы способны с этим справиться. Важно обучить не только пользователей, но и персонал справочной службы. Причем последних надо начать знакомить с продуктом еще на ранних стадиях разработки — скажем, на этапе бета-тестирования.

Перед сдачей приложения в эксплуатацию следует составить документацию, определить требования к резервному копированию данных и разработать  
план восстановления на случай отказа систем. 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.

Помимо перечисленных выше обязанностей, логистик занимается сопровождением промежуточных версий продукта в процессе разработки. Его знания требуются и при изучении поведения приложения в тестовой среде. Кроме того, помощь логистиков необходима при создании тестовых, сертификационных и производственных систем. Они должны также сопровождать приложение в процессе моделирования эксплуатации на этапе бета-тестирования.

Многие организации используют системы мониторинга производительности и стабильности. Поэтому логистики обязательно должны проинформировать разработчиков о наличии таких средств. Например, интеграция средств управления продуктом в среду административной консоли Microsoft Management Console (ММС) упростит интеграцию продукта с другими системами Microsoft.

Размер группы логистики определяется графиком развертывания, уровнем автоматизации установки, наличием средств автоматическо­го развертывания приложений и другими характеристиками этого процесса.

**Размеры группы и масштаб проекта**

В проектной группе за каждое направление должен отвечать как минимум один человек. При реализации крупного проекта возникает затруднение, связанное с эффективным обменом информацией. В небольших организациях или при работе над мелкими проектами роли можно совмещать. Однако в этом случае существует другая проблема — как «усидеть на нескольких стульях» одновременно, не упустив из виду ни одной существенной детали проекта с точки зрения каждой роли.

**Крупные проекты**

В своей книге «Rapid Development: Taming Wild Software Schedules» бывший сотрудник Microsoft Стив Макконнелл пишет: *«Для реализации крупного проекта необходимо, чтобы в организации был наработан опыт формализованного и непрерывного обмена информацией. ...А это возможно при наличии иерархической структуры, то есть небольших групп, в каждой из которых есть сотрудник, отвечающий за взаимодействие с другими группами и менеджерами».*

Таким образом, чтобы справиться с крупным проектом, приходится и делить проектную группу на тематические и функциональные подгруппы.

**Тематические группы**

Это небольшие подгруппы из одного или нескольких человек, роли которых различны. Каждой из таких групп выделяется некий набор функциональных возможностей приложения, за все стороны проектирования и разработки которого она и отвечает (включая составление проекта и графика реализации). Например, какой-либо группе нужно предоставить решать задачу вывода данных на печать.

По этому поводу Стив Маконнелл замечает: *«Достоинства тематических групп — эффективность, сбалансированность и ответственность. Возможности каждой такой группы велики, ведь в ее состав входят представители... всех заинтересованных сторон. Ее члены принимают решения, учитывая все возможные мнения, а, следовательно, нет никаких оснований их изменять.*

*Но этой же причине в такой группе высока ответственность. Ее* *члены могут обратиться ко всем людям, опыт которых необходим в их работе. Поэтому, если они так и не найдут оптимального решения, в этом они смогут винить только себя. Такая группа сбалансирована. Вряд ли вы захотите, чтобы окончательные спецификации*соз*давал только отдел разработки, маркетинга или контроля качества. Только решение, принятое группой представителей каждого из этих отделов, будет по-настоящему сбалансировано».*

**Функциональные группы**

Функциональные группы формируются в рамках одной роли. Они нужны в очень крупных проектных группах или при работе над крупномасштабными проектами, когда отдельные роли нуждаются в дополнительном подразделении. Например, в Microsoft отдел менеджмента продукта обычно состоит из групп планирования и маркетинга. Обе они занимаются менеджментом продукта, но первая отвечает за определение действительно необходимых заказчику функций приложения, а вторая — за информирование потенциальных клиентов о достоинствах продукта.

Приведем еще один пример. Иногда группу разработчиков требуется разделить на три подгруппы, каждая из которых занимается одним уровнем архитектуры (пользовательским, прикладным или уровнем данных). Довольно часто в крупных отделах разработки группу  
программистов подразделяют на группу решений и компонентную группу. Первые создают собственно приложение, «склеивая» вместе его компоненты, разработанные вторыми.  Кстати, это разделение часто оказывается и разделением по языкам программирования: разработчики решений, как правило, работают с языками, позволяющими быстро собирать приложения из готовых компонентов, тогда как выбор языка для разработки повторно используемых компонентов, пригодных для многих проектов, определяется прежде всего соображениями эффективности.

**Небольшие проекты**

Хотя в модели группы разработчиков предусмотрено шесть направлений деятельности, необязательно включать в проектную групп шесть человек. Другими словами, некоторые должности можно совмещать. Конечно, основной смысл такого разделения в том, чтобы каждую из шести задач решал один из членов группы Однако не во всех проектах это возможно.

В небольших группах один человек может играть несколько ролей. При этом мы рекомендуем соблюдать следующие принципы разделения должностей.

·  **Нельзя совмещать разработку с другими видами деятельности —**ее создателей приложения не стоит отвлекать от основной задачи. Если «повесить» на разработчиков дополнительные обязанности, то скорее всего график работ будет нарушен, а дату выпуска продукта придется отодвинуть.

·  **Конфликт интересов**— нельзя совмещать роли, интересы которых противоположны. Пример — менеджер продукта и менеджер программы. Первый хочет выполнить все требования заказчика, второму же надо уложиться в график и бюджет. Если совместить эти роли, возникает опасность упустить просьбу заказчика о внесении изменений в проект либо, напротив, принять их без должного анализа влияния на график работ. Таким образом, назначение на эти роли разных людей позволяет соблюсти интересы всех участников проекта.

На рис. показаны комбинации ролей, оказывающие позитивное и негативное влияние на проект. Роли, отмеченные буквой «3» — Запрещено— нельзя совмещать из-за конфликтов интересов. Вероятность совмещения ролей, отмеченных буквой «Н» — Нежелательно — и из-за сильного различия в необходимой квалификации Например, знания и опыт менеджера продукта и логистика сильно отличаются. Сочетания ролей, помеченные буквой «Д» — Допустимо — возможны, так как их интересы совпадают. Например, как тестеры, так и инструкторы отвечают за выполнение требований пользователей.

Естественно, успех совмещения ролей зависит от конкретных членов группы, их навыков и опыта. В некоторых проектных группах возможно успешное совмещение ролей, комбинация которых согласно нашей таблице опасна. Главное — помнить цели каждого направления и контролировать совмещение должностей в соответствии с ними, предотвращая конфликты. В противном случае некоторые обязанности какой-то роли будут не выполнены, что увеличивает число неконтролируемых рисков.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Менеджер продукта | Менеджер программы | Разра-ботка | Тестиро-вание | Инструктор | Логис-тика |
| Менеджер продукта |  | **З** | **З** | **Д** | **Д** | **Н** |
| Менеджер программы | **З** |  | **З** | **Н** | **Н** | **Д** |
| Разработка | **З** | **З** |  | **З** | **З** | **З** |
| Тестирование | **Д** | **Н** | **З** |  | **Д** | **Д** |
| Инструктор | **Д** | **Н** | **З** | **Д** |  | **Н** |
| Логистика | **Н** | **Д** | **З** | **Д** | **Н** |  |

**Д -**Допустимо **Н -**Нежелательно**З -**Запрещено

**Рис. Деструктивные и созидательные сочетания ролей**

**Создание группы**

Модель проектной группы описывает структуру группы для работы над проектом создания приложений масштаба предприятия. Однако одной структуры недостаточно — важным фактором успеха является квалификация членов группы. В этой главе мы опишем методы проверки их квалификации.

**Поиск руководителей**

Главная задача человека, ответственного за создание проектной группы, — подобрать квалифицированных исполнителей. Эта кажущаяся простой (но на самом деле сложная) задача имеет огромное значение для успеха всего проекта.

Найти лидеров — несложная проблема; в любой организации они всем известны. Важно понимать, что мы говорим именно о лидерах, а не начальниках. Конечно, в любой организации есть менеджеры директора и так далее, но положение в иерархической структуре далеко не всегда гарантирует наличие качеств лидера. Лидеров определяют действия и качества, а не должности.

Зачем нужны именно лидеры? Потому что исполнителей и так избытке. Важно понимать, что деление сотрудников на лидеров и исполнителей не умаляет деловых качеств и квалификации последних. Чтобы лидер добился удачи, он должен набрать отличных исполнителей. Однако между лидерами и подчиненными должно быть взаимопонимание. Группа обсуждает, что нужно делать, а затем выполняет принятое решение, поэтому все, и руководители, и подчиненные, одинаково важны для проекта — в отсутствие кого-либо и них успех проекта невозможен.

Руководители должны обладать:

·  умением понимать и помогать;

·  коммуникабельностью;

·  авторитетом внутри организации и за ее пределами;

·  чувством ответственности за поставленные цели;

·  умением принимать конструктивные решения;

·  уверенностью в своих силах;

·  достаточной для решения поставленных задач квалификацией;

·  способностью помочь другим развить свои таланты и приобрести опыт.

Многие прочтут этот список и скажут: «Я обладаю всеми этим качествами». Еще больше людей подумают: «Я могу обладать этим качествами». Однако настоящего лидера отличают именно эти *способности, а*не *намерения.*Это относится и к качествам руководителя: нужно оценивать оказываемое им влияние, а не его намерения. Ведь давно известно, что «реальность — это то, что видят другие».