

image not found or type unknown



В настоящее время сложность промышленных приложений и систем такова, что процесс их разработки стал практически неуправляемым. Кроме того, их развертывание на сотнях компьютеров, расположенных в разных местах, значительно раздвигает границы процесса разработки.

Один человек не способен создать приложение масштаба предприятия. Ни один разработчик просто не удержит в голове все требования к системе и варианты проекта. Поэтому сегодня разработкой промышленных систем занимаются проектные группы, и все обязанности распределяются среди членов группы.

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 существующую инфраструктуру.

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 — 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 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:

- 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 не выявленные 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.

Как же начать работу над проектом, не зная, сколько времени на это потребуется, сколько проект будет стоить и каких результатов ожидать? 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, в MSF определены шесть таких ролей, которые и составляют модель проектной группы. У каждой из ролей свои обязанности, выполнение которых и обуславливает удачу проекта.

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

В проектную группу должны входить:

- опытные руководители;
- инициативные сотрудники, способные принимать решения и нести ответственность за свое направление работы.

Группа равных

В группе равных важна каждая роль. Такой подход, и только он, делает возможным неограниченный обмен информацией между членами группы, повышает ответственность за выполнение работы и усиливает понимание того, что все шесть целей проекта одинаково важны. Естественно, в таких группах надо проводить проверку качества продукта; это делает представитель заказчика, разбирающийся в решаемой задаче.

Подчеркнем, что равенство — это не анархия. Равенство существует лишь в отношении ролей; в рамках каждой роли следует придерживаться обычной иерархической модели. В каждой группе необходима должностная иерархия и наличие руководителя, ответственного за управление и координирование работы своего направления. Задача остальных членов группы, исполняющих ту же роль, — выполнять поставленные перед ними задачи.

Ориентация на продукт

Ориентация на продукт — это не призыв работать над коммерческим программным обеспечением одним способом, а над приложениями для внутреннего пользования — иначе. Это требование — относиться к результату

работы, как к продукту.

Прежде всего нужно выяснить, является проект самостоятельным или частью более крупного проекта. MSF рекомендует идентифицировать проекты — это позволяет людям чувствовать себя членами команды. Скажем, в Microsoft принята практика присвоения проектам кодовых имен (например «Чикаго» для Windows 95 и «Мемфис» для Windows 98). Кодовые имена четко идентифицируют проект и работающую над ним группу, позволяют людям четче ощущать причастность к проекту и ответственность за него. Создать и усилить значимость группы, поднять ее боевой дух можно разными способами — скажем, помещая название проекта на футболки, кружки и прочие сувениры.

Сформированное отношение к продукту позволяет сконцентрироваться на результате проекта, а не на процессе его достижения вовсе не означает, что процесс не важен для группы. Мы просто иначе расставляем приоритеты — процесс должен служить достижению цели, а сам по себе он не имеет ценности. Увлеченность процессом не должна стоять на пути к результату. Необходимо прежде всего, чтобы все, чувствовали ответственность за выпуск продукта.

Коллективная игра

Главная и основная цель - поставить заказчику работающую систему. Именно так будет оцениваться успех или неуспех проекта. Уже впоследствии люди будут спрашивать, насколько интересным был этот проект, хорошо ли им руководили, красивой ли получилась программа, и легко ли будет поддерживать ее в будущем. Если команда разработчиков не поставит заказчику никакой программы, все сочтут, что проект окончился неудачей. Иногда есть возможность выйти из игры, когда становится понятно, что цель того не стоит, и этой возможностью не стоит пренебрегать. Обратите внимание: это справедливо и для альпинизма, и для разработки ПО.

Правда, между альпинизмом и разработкой программного обеспечения есть одна существенная разница: разработка ПО продолжается от одного проекта к другому, в то время как альпинисты совершают всего одно восхождение. Закончив создание одной программы, разработчики тут же начинают работать над следующей ее версией, над другой программой, которая продолжает или дополняет первую, и т.д.

Таким образом, в отличие от альпинистов, программисты ставят перед собой две цели:

- Поставить заказчику систему;
- Создать основу для следующей игры.

Оценка завершившегося проекта тоже производится двояко: во-первых, оценивается, была ли вообще поставлена программа и, во-вторых, насколько выгодную позицию занимает теперь команда для начала следующей игры.

Обе эти цели конкурируют между собой и претендуют на ограниченные ресурсы команды. Разработчики могут поставить заказчику продукт гораздо быстрее, если в будущем этот продукт не надо будет изменять и дополнять. (А насколько быстрее это можно было бы сделать, если не заниматься исправлением ошибок в программе!) С другой стороны, разработчики могут потратить время на создание более продуманной архитектуры приложения, написание документации и передачу системы команде преемников - но все это отложит срок поставки готовой системы (а иногда может и вообще помешать успешному завершению проекта).

При всем этом команда ограничена во временных, финансовых и человеческих ресурсах - их просто не хватит на то, чтобы идеальным образом справиться с обеими задачами. Большинство разработчиков остаются довольны тем, что сумели достичь первой цели и сделать хоть что-то для достижения второй. Даже те компании, которые обладают достаточным количеством ресурсов, едва ли могут позволить себе уделять второй цели большое внимание, так как она воистину безгранична. В целом же, лучшим из реалистичных вариантов будет поставить заказчику систему приемлемого качества в оговоренное время и сделать хоть какие-то приготовления к следующим играм.

Ориентация на отсутствие дефектов

Ориентация на отсутствие дефектов — это требование соблюдения качества продукта. Оно вовсе не подразумевает, что продукт должен быть выпущен без дефектов; достаточно, чтобы дефектов было не больше нормы, установленной группой на ранних стадиях разработки. Кроме того, это означает, что группе придется постоянно контролировать качество на всех стадиях: если завтра потребуются выпустить готовый продукт, группа должна быть готова выполнить это обязательно. Таким образом, с точки зрения качества продукт всегда должен быть практически готов.

В успешных группах все отвечают за качество продукта — эту обязанность нельзя переложить на других. В этом смысле все члены группы являются представителями заказчика.

Понимание целей бизнеса

Чтобы выпустить успешный продукт, группе недостаточно разбираться только в технической стороне проекта. Создано множество технически совершенных и безупречно написанных продуктов, не способных выполнять задачи, поставленные заказчиком. Чтобы избежать такой ситуации, члены группы должны ясно представлять себе решаемую задачу.

И здесь необходимо активное участие заказчика в процессе разработки. Его обязательно нужно привлекать к созданию концепции проекта, к принятию решений по продукту и его использованию и к анализу промежуточных версий.

Ответственность в равной мере

Чтобы добиться согласованной работы всех членов группы, вы должны сделать их роли взаимозависимыми и перекрывающимися, они все в одинаковой мере должны разделять ответственность за выпуск нужного продукта в запланированные сроки. Таким образом разрушается узкая специализация членов группы, которая ведет к изолированности, а не к совместной работе.

Совместное проектирование

В создании функциональных спецификаций должны участвовать представители всех направлений, так как у каждого из них свое представление о проекте. Они должны убедиться, что в проекте учтены не только требования всей группы, но и требования каждой роли.

Обучение на опыте других проектов

Чтобы не полагаться на удачу, стоит воспользоваться опытом работы над другими проектами, как успешными, так и провалившимися. Изучение опыта — основа совершенствования. Один из методов структурирования и накопления опыта — подведение итогов каждого этапа, зафиксированное в обзоре. Обзор, основа которого — сопоставление планов с результатами, помогает скорректировать дальнейший ход работы и избежать ошибок в дальнейшем. Кроме того, при этом можно выявить хорошо зарекомендовавшие себя методы, чтобы применять их в будущем.

Обучение группы

Работа группы эффективна, если все ее члены понимают, что они делают. Однако во многих случаях это простое правило реализуется недостаточно полно. Выпустить же нужный продукт в срок удастся только при наличии формального и неформального планов обучения проектной группы.

Изучение методологии

Разработка программного обеспечения — это не только создание кода. Поэтому изучение методологии разработки руководителями групп и основными участниками проекта сократит затраты и время выполнения проекта.

Целью любого проекта должно стать улучшение обмена информацией. Для этого в MSF и других моделях разработки применяется общий язык передачи информации о состоянии продукта. Основные элементы этого процесса существуют не один год, и многие разработчики применяют их на практике. Используя универсальный язык MSF, опытные разработчики смогут поделиться своим опытом с другими членами группы.

Изучение технологий

Создать хороший продукт очень сложно. Поэтому группам разработчиков, логистиков и, по возможности, менеджеров программы надо изучать существующие технологии. Каждый день появляются новые формы обучения современным технологиям, однако для успешной реализации таких технологий недостаточно только знания их основ, необходимо понимать способы их применения.

Часто при обучении упускают из вида сопровождение приложения в существующей информационной инфраструктуре. Однако успех проекта зависит не только от качества созданного кода. Проектная группа должна исследовать взаимное влияние инфраструктуры и приложения друг на друга. Как скажется на приложении появление новых операционных систем, новых версий существующих операционных систем или новых базовых приложений, используемых продуктом?

Кроме того, проектной группе придется учесть то обстоятельство, что во время работы над проектом могут появиться новые технологии или новые версии приложений. Нужно уметь оценивать влияние новых функциональных возможностей таких продуктов на жизненный цикл проекта. Поэтому время на

обучение следует включать в график проекта еще до определения даты выпуска приложения.

Координация работы с внешними группами

Проектная группа, рассчитывающая на успех, должна взаимодействовать с внешними группами — как с заказчиком и пользователями, так и с другими разработчиками. Этим занимаются менеджер программы, менеджер продукта, инструктор и логистик. В обязанности этих ролей входят как внутренние, так и внешние контакты, в то время как разработчики и тестеры изолированы от общения с внешним миром. (Такая изоляция приводит к повышению эффективности труда этих двух групп.)

Важно, чтобы взаимодействие с внешними группами было явным и понятным. Проектным группам, как правило, приходится взаимодействовать с большим количеством внешних организаций, включая финансовые и юридические подразделения и отделы контроля качества продукта.