

Тема: «Коллективная разработка ПО»

# Модели коллективов разработчиков

---

Разработки программы в зависимости от количества участников и типов взаимоотношений между участниками делятся на виды:

- авторская разработка
- коллективная разработка
- общинная разработка

# Авторская разработка

**Авторская разработка** – это создание программных продуктов, при котором весь жизненный цикл разработки поддерживается одним человеком. Был распространён в 60-70 г.г. и сейчас применяется редко из-за сложности, объема, требуемого качества, сопровождения.

С появлением ПК программное обеспечение стало продуктом массового применения, поэтому стали доминировать крупные компьютерные компании с развитой структурой менеджмента и рекламной компанией. Авторские разработки применяются в области наукоемких приложений. Для таких приложений характерна необходимость многолетнего изучения предметной области, практически полное отсутствие начального финансирования проекта, малая рентабельность, которая определяется узким кругом области.



Авторская разработка выигрывает по производительности 30 и более раз коллективной разработки за счёт:

- исключения меж личных коммуникаций, связанных с необходимостью порождения и изучения большого количества документации
- исключения работ по разбиению проекта на составляющие, по распределению работ между исполнителями
- координации действия между исполнителями и контролем за их работой

Авторская разработка предполагает достижение профессионального успеха. В принципе это реально, но необходимо найти свою нишу. В настоящее время авторская разработка применяется при создании условно бесплатных программных продуктов.

# Коллективная разработка

**Коллективная разработка** – это создание программного продукта коллективом разработчиков. Один из основных вопросов коллективной разработки является разделение труда.

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** (начальник -> подчинённые)
- 2) **модель группы** (не определяет структуру коллектива с точки зрения отдела кадров)

Основной моделью разделения труда в наше время является иерархическая структура.

# Иерархическая модель

Если в современных производственных средах один менеджер проекта (начальник) отвечает за все тонкости разработки и принимает все важные решения, возникает множество проблем, ведущих к провалу проекта. **Иерархическая модель грешит множеством недостатков:**

- нехватка информации
- невозможность учесть все особенности проекта
- отсутствие полноценной связи между всеми участниками проекта
- трудность освоения новых технологий
- сложность расстановки приоритета (чтобы справиться, нужно быть гениальным)
- структура равноправности соисполнителей (доверительность-каждый отвечает за своё)

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.

# Модель группы

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. 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»..

## **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.

# Примерный состав бригады:

---

1. разработчики
  - специалисты по инженерии и программированию (приёмы и методы программирования, знания инструментальных систем)
  - программисты
2. технические писатели
3. специалисты по качеству
4. специалисты по сопровождению
5. специалисты по продаже

# Разделение по типам работы:

1. разработка приложений требует:
  - программистов
  - специалистов по инженерии и программировании
  - специалистов по инженерии и знаниям
2. работа с приложениями - эксплуатация требует:
  - специалистов по приложениям
  - администраторов данных
  - администраторов баз данных
3. техническая поддержка
  - системный администратор
  - сетевой администратор
  - администратор коммуникации
4. обеспечение качества программы
  - технический писатель
  - специалист по тестированию
  - специалист по качеству
5. маркетинг
  - специалист по сопровождению
  - специалист по продаже продукта
6. системное интегрирование
  - системный интегратор (основные задачи: предложить заказчику варианты решения проблемы, выбрать приемлемый вариант по цене, технике и реализовать)

# Модель бригады главного программиста

## 1. **Главный программист:**

- анализ
- проектирование
- создаёт, реализует, пишет документацию.
- большой опыт работы и существенными знаниями с большим талантов

## 2. **Дублёр**

- Может выполнять любую работу главного программиста, но с небольшим опытом работы. Не несёт ответственность.

## 3. **Администратор (менеджер)**

- контроль денег, людей, помещений, машинных ресурсов, контактов с другими группами и руководством.

## 4. **Редактор технический**

- Критически перерабатывает документацию. Перерабатывает языки документации, которые разрабатывает программист.

## 5. **Языковед**

- Эксперт в тонкостях языка программирования.
- Может найти эффективные способы программирования. Обычно работает с несколькими бригадами программистов.

## 6. **Инструментальщик**

- Разработчик специализированных инструментальных утилит, скриптов, поддерживает основной инструментарий и оказывает по нему консультации. Может осуществлять администратор. Работает с несколькими бригадами.

## 7. **Отладчик**

- Разработчик тестов и организатор тестирования ПП (тестировщик, тестер)

## 8. **Делопроизводитель**

- Отвечает за регистрацию всех данных в бригаде, благодаря активным программистам освобождается от рутинных работ. В настоящее время функции делопроизводителя автоматизированы и выключены из проекта.

# Модели коллективной разработки компании Microsoft (MSF)

---

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 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.

# 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.

- Менеджер продук
- Менеджер програи
- Разработчики
- Тестеры
- Инструктор
- Логистик



# Цели и роли

Шесть ролей модели проектной группы 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.

<b>Цель</b>	<b>Роль</b>
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

## 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.

Менеджер продукта сообщает группе предварительную дату выпуска продукта, устанавливаемую заказчиком.

**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.

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.

# Менеджер программы

*Менеджер программ* – его задача вести процесс разработки с учетом всех ограничений. Является руководителем разработки. Его цель повысить авторитет каждого члена группы, а не подавить их своим авторитетом. Главная обязанность выполнить все стадии разработки так, чтобы нужный продукт выпущен был в назначенное время. Он координирует деятельность других членов группы.

Главный менеджер программы составляет график проекта на основе информации, полученной от остальных членов группы.

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

Менеджер программы отвечает и за бюджет проекта, объединяя требования к ресурсам всех членов группы в единый план расходов.

# 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.

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.

# 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.

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.

Некоторые важные обязанности тестеров :

- **уведомление об ошибках и их отслеживание;**
- **контроль сборки продукта;**
- **выявление и контроль рисков.**

# 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.

Персонал группы тестирует удобство использования продукта, выявляет проблемы в этой области и проверяет проект пользовательского интерфейса.

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, онлайн-помощи, Web-страницы и даже — если понадобится — целый учебный курс. Когда материалы подготовлены, группа координирует обучение пользователей.

**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.

# 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.

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.

# 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.

Чтобы справиться с крупным проектом, приходится 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.

## **Функциональные грппы**

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.

# Небольшие проекты

Хотя в модели группы разработчиков предусмотрено шесть направлений деятельности, необязательно включать в проектную групп шесть человек. Некоторые должности можно совмещать.

---

В небольших группах один человек может играть несколько ролей. При этом мы нужно 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.

# Деструктивные и созидательные сочетания ролей

На рис. показаны комбинации ролей, оказывающие позитивное и негативное влияние на проект. Роли, отмеченные буквой «З» — **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.

	Менеджер продукта	Менеджер программы	Разработка	Тестирование	Инструктор	Логистика
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	

# Повышение эффективности коллективной работы

---

- заинтересованность;
- надежда, оптимизм, готовность к работе;
- определение задач и решений;
- проявление взаимопомощи;
- доверительные уважительные отношения;
- единение.

Для успеха проекта недостаточно только распределить роли и обязанности. Помимо структуры, следует придерживаться определенных принципов и методов. Ниже обсуждаются «лучшие методы и принципы», которые успешно применялись не только внутри Microsoft, но также партнерами и заказчиками компании.

# Общее представление о проекте

Важнейший фактор для успеха проекта — единое понимание целей и задач проекта всеми участниками. Каждый из них изначально имеет **свое** мнение, касающееся приложения. В процессе формирования общего представления о проекте все эти мнения обсуждаются, что позволяет добиться единства целей всех членов проектной группы и заказчика.

- На основе выработанного представления о проекте создается документ «Концепция проекта», который:
- описывает не только то, что делает продукт, но и то, чего он не делает;
  - конкретизирует продукт (например, позволяет включать и исключать определенные функциональные возможности из данной версии);
  - побуждает группу достичь сформулированной цели;
  - содержит описание путей реализации проекта, благодаря чему проектная группа и заказчик могут начать работу.

# 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.

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.

# Принципы и методы эффективной коллективной работы

- **Общее представление о проекте**
- **Группа равных**
- **Ориентация на продукт**
- **Ориентация на отсутствие дефектов**
- **Понимание целей бизнеса**
- **Ответственность в равной мере**
- **Совместное проектирование**
- **Обучение на опыте других проектов**
- **Обучение группы**
- **Изучение методологии**
- **Изучение технологий**

# Координация работы с внешними группами

