Contents

[Основные роли, встречающиеся в управлении и разработке IT-проекта 1](#_Toc429916155)

[Для чего нужно разделение ролей 1](#_Toc429916156)

[Распределение функций между исполнителями 1](#_Toc429916157)

[Подход компании Microsoft к распределению ролей 3](#_Toc429916158)

[Основные роли, встречающиеся на проекте и их обязанности 4](#_Toc429916159)

[Возможность совмещения ролей 7](#_Toc429916160)

## Основные роли, встречающиеся в управлении и разработке IT-проекта

### Для чего нужно разделение ролей

Каждый ИТ специалист идет по пути наименьшего сопротивления, как правило, пытаясь сбросить часть работы на коллегу. Для избегания возможных конфликтов в команде нужно четко разграничить участок работ каждого участника, в том числе заказчика. Для целенаправленного выполнения проекта должен быть выполнен ряд работ, различных как по своему назначению, так и по квалификационным требованиям, предъявляемым к разработчикам. Иными словами, в ходе развития проекта командой разработчиков выполняются те или иные функции.

### Распределение функций между исполнителями

Функции, выполняемые разработчиками, — понятие неформализованное. В разных проектах оно может обретать свое содержание. Тем не менее типовые функции, которые предполагают практически все программные проекты, можно перечислить. Так, в любом программном проекте есть функции кодирования, т.е. записи программы на алгоритмическом языке по имеющимся спецификациям, анализа требований, т.е. выявления истинной потребности в создаваемой программе, тестирования и отладки. Далее мы рассмотрим эти и другие функции разработчиков, связанные с проектом, а пока лишь заметим, что в рамках деятельности менеджера любого проекта необходимо организовать распределение функций проекта между исполнителями. Вполне естественно считать эти действия одной из функций менеджера. В результате ее выполнения члены команды, выполняющей проект, начинают играть соответствующие роли.  
Обычно роль объединяет родственные функции. Также принято обозначать роли их главными функциями. Продолжая только что приведенную иллюстрацию функций, выполняемых разработчиками проекта, укажем на следующие роли: кодировщик — действующее лицо, главной функцией которого является кодирование, аналитик — тот, кто занимается анализом требований. Подобную характеристику можно дать и тестировщику. Что же касается функции отладки, то в реальных проектах она обычно подразделяется на несколько видов: отладка компонентов, которой занимаются разработчики компонентов (например, те же кодировщики) и комплексная отладка, которая может поручаться специально выделенным сотрудникам или рассматриваться в качестве задачи группы разработчиков. Часто выполняются такие работы, как отладка спецификаций, декомпозиция проекта и др. Иными словами, функция отладки обычно не рассматривается как образующая роль, а распределяется по нескольким ролям в соответствии с принятой стратегией развития проекта.

Не следует путать функции, которые предписано выполнять разработчику как исполнителю определенной роли, с поручениями в проекте. Поручения — это разовые или систематические задания, из которых обычно и складываются действия, необходимые для выполнения той или иной функции. Если для функции определен регламент выполнения поручений, т. е. последовательность выполнения составляющих ее поручений не требует дополнительных разъяснений для исполнителя, то такая функция называется технологической. В случае нетехнологической функции сотруднику, который выполняет соответствующую роль, приходится самому выстраивать нужную последовательность. Иными словами, технология здесь уступает место ремеслу.

При разработке любых проектов естественно стремление к повышению их технологичности, к установлению регламента для как можно большего числа функций. И одним из способов достижения этого для менеджера является использование работников с нужной квалификацией, для которых поручаемые им роли оказываются технологичными, т.е. состоящими только из технологических функций. К сожалению, реальность такова, что менеджеру приходится работать в условиях ограниченных возможностей в подборе кадров, а потому уровень технологичности выполнения проекта снижается по сравнению с идеальной ситуацией.  
Таким образом, в рамках любого проекта возникает задача повышения квалификации сотрудников. Для различных схем ведения проектов эта задача решается по-разному. Крайнюю точку зрения на проблему соответствия квалификации работников требованиям проекта отражает идея **экстремального программирования**, когда используется неформальная организация группы исполнителей проекта без четкого распределения ролей, а значит, и обязательств сотрудников. В этом случае провозглашается принцип, когда каждый в группе делает то, что он умеет делать лучше всего. И хотя все функции, которые должны выполняться, остаются, создается впечатление, что в группе исполнителей проекта исчезают роли. В результате возможны пробелы в разработке, в частности при анализе и декомпозиции проектируемой системы. Чтобы этого избежать, разработчики должны понимать, какую абстрактную роль они исполняют в каждый конкретный момент, выполнение каких проектных функций необходимо сейчас, как связаны между собой работы, как должны распределяться ресурсы. Словом, они должны обращать внимание на выполнение распределенных по группе менеджерских функций. И даже в этом случае схему экстремального программирования можно рекомендовать лишь для слаженных групп исполнителей с высоким уровнем коллективной ответственности.

Функции, выполняемые разработчиками в проекте, подразделяются на организационные и производственные. Первые создают условия для выполнения проектных заданий, вторые непосредственно связаны с этими заданиями. Часто неудачи проекта возникают из-за того, что менеджер не учитывает важность выполнения организационных функций. Так, обычно проектное задание фиксирует лишь то, что нужно предъявлять заказчику в качестве результатов. С точки зрения результатов просто не требуется знать, например, как организована передача проектных материалов между разработчиками, какая процедура отчетности предусматривается, но игнорирование задач реализации информационных потоков в проекте может привести к хаосу, что в конечном итоге отразится и на результатах.

Понятно, что как состав, так и значимость ролей разработчиков и тех, кто с ними связан, различаются в зависимости от выполняемого проекта, от коллектива исполнителей, принятой технологии, от других факторов. Неоднозначность выбора ролей приводит к тому, что их список часто становится непомерно велик. Чрезмерное увеличение числа есть следствие отождествления роли и функции и, соответственно, игнорирования понятия родственности функций. В то же время, если ролей выбрано недостаточно, есть опасность, что не все нужные в проекте функции будут охвачены планированием и управлением.

### Подход компании Microsoft к распределению ролей

Как конкретный разработчик может получать одновременно несколько ролей, так и роль может быть распределена между несколькими исполнителями. Когда менеджеру в конкретных условиях руководства коллективом придется распределять роли, он неизбежно столкнется с тем, что эта задача зависит и от специфики проекта, и от контингента исполнителей. В связи с этим уместно упомянуть об одном из ее решений, которое предлагается специалистами Microsoft в качестве универсального подхода.

Предлагается образовывать небольшие мобильные коллективы как атомарные производственные единицы с общей ответственностью за выполняемые задания — так называемые проектные группы. Такие группы строятся как многопрофильные команды, члены которых распределяют между собой ответственность и дополняют области компетентности друг друга. Группа состоит не более чем из 10 человек. Все они считаются обладающими сходным уровнем профессиональной подготовки, хотя и в разных областях индивидуальной специализации. Провозглашается равноправие членов группы и коллективная ответственность за выполняемые задания: проектная группа — команда равных. Все это позволяет сохранять внутри группы неформальные отношения.  
Вместо понятия роли для группы в целом определяются **ролевые кластеры**, которые заполняются точно так же, как происходит распределение ролей. В то время как за успех проекта ответственна вся команда, каждый из ее ролевых кластеров, определяемых моделью, ассоциирован с одной из проектных целей и работает над ее достижением. В данной модели именно эти цели задают роли разработчиков, которые определяются кластерами. В терминологии используется понятие области компетенции, или области функциональной специализации (functional area), обозначающее ту или иную роль, которую выполняет кластер группы в проекте. Принципиальное отличие распределения исполнителей по ролевым кластерам от распределения ролей заключается лишь в том, что ответственность за это несет не менеджер проекта, а сама группа. Менеджер проекта выдает задания и контролирует их выполнение лишь в целом для группы, не вмешиваясь в ее работу.

Определено шесть ролевых кластеров, которые соответствующим образом структурируют проектные функции разработчиков:

* **Управление продуктом (product management)**. Ключевая цель кластера — обеспечивать удовлетворение интересов заказчика. Для ее достижения кластер должен содержать следующие области компетенции:
  + планирование продукта;
  + планирование доходов;
  + представление интересов заказчика;
  + маркетинг.
* **Управление программой (program management)**. Задача — обеспечить реализацию решения в рамках ограничений проекта, что может рассматриваться как удовлетворение требований к бюджету проекта и к его результату. Области компетенции кластера:
  + управление проектом;
  + выработка архитектуры решения;
  + контроль производственного процесса;
  + административные службы.
* **Разработка (development)**. Первостепенной задачей кластера является построение решения в соответствии со спецификацией. Области компетенции кластера:
  + технологическое консультирование;
  + проектирование и осуществление реализации;
  + разработка приложений;
  + разработка инфраструктуры.
* **Тестирование (test)**. Задача кластера — одобрение выпуска продукта только после того, как все дефекты выявлены и устранены. Области компетенции кластера:
  + разработка тестов;
  + отчетность о тестах;
  + планирование тестов.
* **Удовлетворение потребителя (user experience)**. Цель кластера — повышение эффективности использования продукта. Области компетенции кластера:
  + общедоступность (возможности работы для людей с недостатками зрения, слуха и др.);
  + интернационализация (эксплуатация в иноязычных средах);
  + обеспечение технической поддержки;
  + обучение пользователей;
  + удобство эксплуатации (эргономика);
  + графический дизайн.
* **Управление выпуском (release management)**. Задача кластера — беспрепятственное внедрение и сопровождение продукта. Области компетенции кластера:
  + инфраструктура (infrastructure);
  + сопровождение (support);
  + бизнес-процессы (operations);
  + управление выпуском готового продукта (commercial release management).

### Основные роли, встречающиеся на проекте и их обязанности

Существуют следующие роли на IT-проектах:

* **Заказчик (Customer)** — отвечает за:
  + своевременный просмотр спецификаций и других присылаемых документов (с целью утвердить документ, дать комментарии, исправить неточности и т.п.);
  + внесение замечаний, дефектов, пожеланий в багтрекинговую систему;
  + своевременный просмотр каждого деливери после его поставки и предоставление комментариев.
* **Планировщик ресурсов (Planner)**:
  + выдвигает и координирует требования к проектам в организации, осуществляющей данную разработку;
  + развивает и направляет план выполнения проекта с точки зрения организации.
* **Менеджер проекта (Project Manager)** — отвечает за:
  + организацию процесса разработки;
  + координацию и контроль всех видов деятельности в проекте;
  + разработку плана проекта (Project Plan);
  + проведение регулярных статус митингов в проектной группе;
  + контроль готовности передачи и нового билда для QA;
  + предоставление заказчику документации и промежуточных версий для просмотра и утверждения или комментирования;
  + регулярное общение с заказчиком, выяснение требований;
  + предоставление отчетов о статусе проекта.
* **Руководитель команды (Team Leader)**

TeamLead, как следует из самого слова – это лидер команды разработчиков. Это человек, который отвечает за все, что касается разработки в проекте. У него очень богатый функционал – это снятие запроса клиента, оценка объема и сроков работы с учетом рисков и особенностей своей команды, распределение задач внутри команды с максимальной эффективностью, планирование и выпуск релизов, и даже мотивация команды на работу и участие в презентациях ключевых релизов заказчику.

Такой человек очень важен, ведь менеджер, как правило не является техническим специалистом, поэтому у него могут возникнуть сложности, как на этапе оценки задачи, так и на этапе организации разработки и презентации релизов. Руководитель также пишет код, но на его написание он тратит не более 20-30% рабочего времени, так как в приоритете у него управление командой.

Компетенции, необходимые человеку, претендующему на такую позицию в проекте, можно разделить на две группы: технические и менеджерские.

*Технические компетенции*

Team Leader должен, в первую очередь, уметь преобразовать бизнес-задачу в задачу техническую, понятную для разработчиков. Это не так просто, как кажется, ведь бизнес-задачи порой формулируются очень и очень пространно. Более того, ему необходимо уметь объяснить разработчикам не только то, что нужно сделать, но и зачем это нужно.  
Лидер команды разработчиков должен безусловно иметь понимание процесса промышленной разработки с точки зрения разделения ответственности в команде, процесса управления требованиями, качеством и технологий, необходимых для обслуживания процесса. Руководитель проекта должен обладать солидным опытом разработки – от 5 лет и более.

*Менеджерские компетенции*

В данную группу можно отнести практически все стандартные компетенции менеджера, однако стоит выделить самые важные:

* мотивация команды. Те специалисты, которые научатся делать это правильно, получат огромное преимущество перед остальными. Человеку, работающему со своей командой бок о бок гораздо проще мотивировать людей, чем формальному начальнику;
* наставник для членов своей команды. Поэтому ему нужно уметь правильно развивать людей — распределять интересные и сложные задачи, правильно   
  давать развивающую обратную связь, составлять планы развития.

Конечно, это далеко не полный список навыков, необходимых руководителю. Не стоит также забывать об универсальных личностных качествах, которые являются столь же важными, как и профессиональные компетенции:

* + ответственность;
  + умение грамотно излагать мысли письменно и устно. Лидер, конечно, не должен быть профессиональным оратором и писателем, но обладать элементарными навыками письма и устного изложения своих мыслей просто жизненно необходимо;
  + честность — самое главное качество. Умение быть честным в первую очередь перед самим собой, способность не закрывать глаза на моменты, когда что-то идет не так. Честность перед членами команды, менеджерами проекта и заказчиком.

Даже если в команде есть очень сильный тимлид, у менеджера все равно останется целый ряд задач. Это и административная работа по ведению договоров и прочих документов, и планирование бюджета и сроков. При отсутствии аккаунт-директора, менеджер принимает на себя его обязанности и занимается поддержанием и развитием отношений с клиентом.

* **Системный аналитик (Technical Leader) —** отвечает за:
  + координацию и контроль деятельности по дизайну, архитектуре и кодированию;
  + поддержку контроля версий;
  + настройку скрипта для авто-билдера и своевременную сборку версий.
* **Архитектор (Architect)** — отвечает за:
  + проектирование архитектуры системы;
  + согласование развития работ, связанных с проектом.
* **Проектировщик подсистемы (Designer)** — отвечает за:
  + проектирование подсистемы или категории классов;
  + определение реализации и интерфейсов с другими подсистемами.
* **Эксперт предметной области (Domain Expert)** — отвечает за:
  + изучение сферы приложения;
  + поддержку направленности проекта на решение задач данной области.
* **Разработчик (Developer)**:
  + разработку качественного кода;
  + проведение модульного тестирования;
  + поддержку контроля версий;
  + написание пользовательской документации, относящейся к инсталляции и администрированию.

Это широкое понятие, которое может подразделяться на специальные роли (например, разработчик классов). В зависимости от сложности проекта команда может включать различное число разработчиков.

* **Бизнес аналитик (Business Analyst)** отвечает за:
  + выяснение и анализ всех требований заказчика;
  + фиксирование всех требований заказчика (в багтрекинговой системе и в функциональных спецификациях), прослеживание всех изменений в требованиях;
  + написание и поддержка спецификаций.
* **Разработчик информационной поддержки (Information Developer**):
  + 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.

* **Специалист по пользовательскому интерфейсу (Human Factors Engineer)**:
  + отвечает за удобство применения системы;
  + работает с заказчиком, чтобы удостовериться, что пользовательский интерфейс удовлетворяет требованиям.
* **QA менеджер (Quality Assurance manager) —** отвечает за:
  + организацию и контроль процесса тестирования в проекте;
  + планирование тестирования;
  + участие в адаптации процесса разработки под проект, анализ его качества;
  + анализ результатов тестирования и качества продукта;
  + участие в управлении требованиями;
  + участие в настройке багтрекинговой системы, полное прослеживание багов;
  + контроль готовности деливери и нового билда для QA.
* **QA аналитик (QA Analyst)** — отвечает за:
  + подготовку тест дизайна;
  + написание тест кейс спецификаций;
  + проведение тестирования;
  + регистрацию багов;
  + прослеживание и проверку багов;
  + написание документации пользователя.
* **Библиотекарь (Librarian**) — отвечает за:
  + создание и ведение общей библиотеки проекта, которая содержит все проектные рабочие продукты, а также за соответствие рабочих продуктов стандартам.

### Возможность совмещения ролей

|  |  |
| --- | --- |
| **Роли действующих лиц проекта и совмещение ролей** | |
| **Роли** | **Характеристика совмещения ролей** |
| Менеджер и архитектор | Желательно |
| Менеджер и руководитель команды | Противоречиво |
| Руководитель команды и архитектор | Возможно |
| Руководитель команды и проектировщик подсистемы | Нежелательно |
| Менеджер и разработчик | Не допускается |
| Для различных разработчиков | Эффективно с ограничениями(обычное дело) |
| Создание документации (все сотрудники) | Успешно распределяется |
| Специалист по интерфейсу и менеджер | Разумно |
| Эксперт предметной области и менеджер | Зачастую разумно |
| Специалист по интерфейсу и эксперт предметной области | Редко бывает эффективно |
| Эксперт предметной области и разработчик | Бывает полезно |
| Специалист по интерфейсу и разработчик | Часто полезно |
| Библиотекарь и один из разработчиков | Допустимо |
| Тестировщики и другие члены команды | Перекрестно |
| Эксперт предметной области, тестировщик | Оправданно |