Лекция 6

Тема. Модель проектной группы и модель процесса разработки.

План лекции

1. Модель проектной группы
2. Характеристики модели процессов MSF

Модель проектной группы

Модель проектной группы определяет роли и обязанности участников на разных стадиях выполнения проекта. Эти роли взаимосвязаны и перекрываются — таким способом обеспечивается постоянное взаимодействие участников.

Данная модель определяет шесть основных ролей участников проектной группы:

- менеджера продукта, отвечающего за удовлетворение требований заказчика;
- менеджера программы, ответственного за своевременное выполнение проекта в рамках бюджета;
- разработчика, реализующего систему (а также, например, разрабатывающего комплект проектной документации на сеть);
- тестера, испытывающего не только продукт, но и его спецификации и документацию к нему;
- логистика, отвечающего за разворачивание продукта и легкость его поддержки;
- инструктора, деятельность которого направлена на повышение эффективности труда пользователей (не производительности системы!). Кстати, инструктор организует и обучение разработчиков, если в проекте предстоит применить новую технологию.

Участники проектной группы взаимодействуют не только между собой, но и с участниками проекта, не входящими в проектную группу. Обязанность взаимодействия с

Рис. 6.1. Роли в модели проектной группы

каждым внешним контрагентом возложена на определенного участника проектной группы (рис. 6.1).

Представления MSF о том, какие роли и при каких условиях совместимы (см. таблицу), со временем меняются, однако и прежде, и сейчас MSF считает совершенно недопустимым поручать одному и тому же человеку роль разработчика и любую другую роль, ведь именно разработчик создает продукт, и отвлекать его неразумно. Считается также недопустимым или по крайней мере не рекомендуется совмещение ролей менеджера продукта и менеджера программы, потому что первый ратует за характеристики решения, пусть ценой задержки и перерасхода средств, а второй борется за своевременное выполнение проекта в рамках бюджета и готов пожертвовать возможностями программы или сети.

Каждая роль выполняет определённые функции. За каждой ролью может находится как один сотрудник, так и целый отдел (в больших проектах). Допустимо также и совмещение ролей в малых проектах. При этом рекомендуется соблюдать следующие принципы совмещения должностей:

	Менеджер продукта	Менеджер программы	Разработ-чик	Тестер	Инструк-тор	Логистик
Менеджер продукта		З	З	Д	Д	Н
Менеджер программы	З		З	Н	Н	Д
Разработ-чик	З	З		З	З	З
Тестер	Д	Н	З		Д	Д
Инструк-тор	Д	Н	З	Д		Н
Логистик	Н	Д	З	Д	Н	

В данной таблице:
З – запрещается совмещать.
Н – нежелательно совмещать.
Д – допускается совмещать.

Менеджер продукта

Менеджер продукта должен вовремя реагировать на потребности заказчика. Его главная задача – сформировать общее представление о поставленной задаче и о том, как её решить. Он должен ответить на вопрос «Зачем мы всё это делаем?» 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 разработчиков у заказчика. (Важно понимать разницу между заказчиком и пользователями. 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 программы

Задача менеджера программы — вести процесс разработки с учетом всех ограничений. Руководитель этого направления должен понимать разницу между понятиями «руководитель» и «начальник».

Главная обязанность менеджера программы — выполнить все стадии разработки так, чтобы нужный продукт был выпущен в нужное время. Он координирует деятельность других членов группы.

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. Он должен полностью положиться на опыт остальных сотрудников и только следить 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.

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. Как программисты разработчики отвечают за низкоуровневое проектирование и оценку затрат на реализацию продукта.

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. На этих стадиях их задача — определить методы реализации функциональных возможностей и заданной архитектуры, а также 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.

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.

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.

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 продукта, 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.

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 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 этим требованиям. Обычно данные вопросы учитываются в планах выпуска, развертывания и сопровождения продукта

Модель проектной группы не только описывает цели, требуемые умения и действия участников в рамках их ролей. Эта модель включает также ряд принципов, тесно увязанных с принципами модели процесса: описывается, какие принципы одной модели способствуют соблюдению каждого принципа другой модели. Так, следование принципу тесного взаимодействия участников обеспечивается проведением периодических собраний проектной группы.

Применение модели проектной группы позволяет точнее распределять обязанности исполнителей; благодаря этому можно избежать некоторые конфликты интересов и повысить качество разработок. Знание ролей участников позволяет легче подключать к проекту персонал заказчика, что всегда является источником проблем. Очень полезными оказались рекомендации по масштабированию группы: в малых проектах на одного человека возлагается несколько ролей (с учетом их совместимости); в больших, наоборот, создаются функциональные группы, реализующие одну роль, или тематические группы, полностью отвечающие за часть решения.

Характеристики модели процессов MSF

Тремя особенностями модели процессов MSF являются:

- Подход, основанный на фазах и вехах.
- Итеративный подход.
- Интегрированный подход к созданию и внедрению решений.

Подход, основанный на вехах

Характеристики подхода, основанного на вехах. Занимая центральное место в методологии MSF, вехи используются как опорные точки для планирования и мониторинга хода проекта. MSF вводит два типа вех: главные и промежуточные. Они имеют следующие особенности:

- Главные вехи служат точками перехода от одной фазы к другой. Они также определяют изменения в текущих задачах ролевых кластеров.
- В MSF используются обобщенные главные вехи, большинство из которых применимо к любому типу IT-проектов.
- Промежуточные вехи показывают достижение в ходе проекта определенного прогресса и расчленяют большие сегменты работы на меньшие, обозримые участки.
- Промежуточные вехи могут варьироваться от проекта к проекту. MSF рекомендует использовать определенный набор промежуточных вех, но на практике проектная группа может сама устанавливать их в соответствии с особенностями своей работы.

Главные вехи – это моменты жизненного цикла проекта, когда полученные на той или иной фазе результаты синхронизируются членами проектной группы друг с другом и с ожиданиями заказчика. В этот момент заказчиком, заинтересованными сторонами и проектной группой производится формальный анализ достигнутого прогресса. Успешное прохождение главной вехи знаменует согласие проектной группы и заказчика продолжать далее работу над проектом. Хотя в принципе возможно отодвинуть время окончания проекта на неограниченно долгий срок, минимизировав тем самым всю имеющуюся в нем неопределенность, такое решение дорого стоит и не отвечает реальным бизнес-нуждам. Вехи позволяют заказчику и проектной группе проверить соответствие рамок проекта потенциально изменяющимся требованиям, и если необходимо, скорректировать их, а также отреагировать на возникающие риски. Хотя ролевой кластер “Управление программой” организует работу над проектом в целом, на каждой из фаз определенные ролевые кластеры имеют ведущее значение. Объем работы, выполняемой различными ролевыми кластерами, меняется в процессе перехода проекта из одной фазы в другую. Использование проектных вех помогает должным образом организовать эти переходные процессы.

Ведущие роли различных фаз

- Между ролевыми кластерами и главными вехами существует определенное соответствие. Оно указывает, какие именно роли несут первоочередную ответственность за достижение каждой из вех. Переход от одной фазы к другой включает в себя также перенос основной ответственности от одних ролевых кластеров к другим.
- Нижеследующая таблица показывает, какие роли являются первостепенными в достижении каждой из главных вех. Однако следует отметить, что ведущее положение одних ролей никоим образом не исключает участие в работе остальных ролевых кластеров.

Веха	Ведущие ролевые кластеры
Концепция утверждена	Управление продуктом
Планы проекта утверждены	Управление программой
Разработка завершена	Разработка, Удовлетворение потребителя
Готовность решения утверждена	Тестирование, Управление выпуском
Внедрение завершено	Управление выпуском

Анализ пройденных вех

Каждая главная веха предоставляет возможность осмыслить и извлечь уроки из только что завершившейся фазы. Анализ пройденных вех во время специально проводимых собраний помогает повысить отдачу от такого осмысления. MSF отдельно рассматривает собрания, на которых результаты фазы обсуждаются вместе с заказчиком и другими заинтересованными сторонами, и последующее излечение уроков внутри коллектива. Окончательные собрания такого рода проводятся уже после завершения проекта. В некоторых организациях они носят название постмортемов.

Итеративный подход

Итеративный подход к процессу разработки широко используется в MSF. Программный код, документация, дизайн, планы и другие рабочие материалы создаются, как правило, итеративными методами. MSF рекомендует начинать разработку решения с построения, тестирования и внедрения его базовой функциональности. Затем к решению добавляются все новые и новые возможности. Такая стратегия именуется стратегией версионирования. Несмотря на то, что для малых проектов может быть достаточным выпуск одной версии, рекомендуется не упускать возможности создания для одного решения ряда версий. Рис. 6.2 показывает, как с созданием новых версий эволюционирует функциональность решения. Версии решения не обязательно следуют одна за другой. Зрелые программные продукты обычно развиваются по нескольким направлениям параллельно. Временные интервалы между выпусками версий зависят как от размера и типа проекта, так и от нужд и стратегии заказчика.

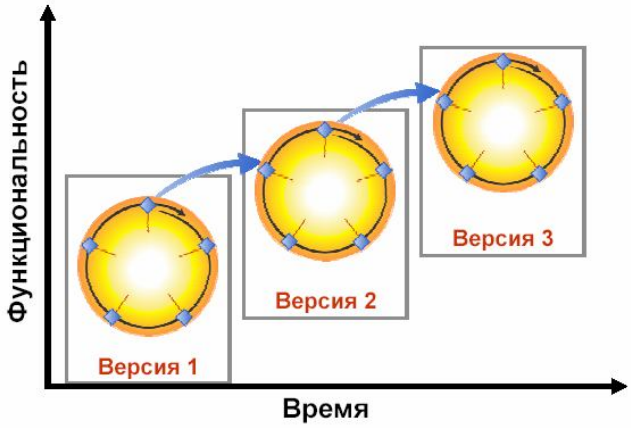

Рис.6.2. Версионирование

Создание “живой” документации

Итеративный подход к процессу разработки требует использования гибкого способа ведения документации. “Живые” документы должны изменяться по мере эволюции проекта. Такой подход существенно отличается от принципов ведения документации в каскадной модели, где процесс разработки начинается лишь после того, как готовы и зафиксированы все требования и спецификации.

Документация проектов MSF, также как и программный код, разрабатывается итеративно. Зачастую, планы первоначально имеют форму описания высокоуровневых “подходов”. На фазе выработки концепции они распространяются среди членов проектной группы и других заинтересованных лиц для получения отзывов. После перехода к фазе

планирования эти документы постепенно дорабатываются. Разработанные детальные планы снова поступают на проверку всем заинтересованным сторонам, и описанный процесс повторяется итеративно. Типы планов и общее количество описывающих их документов могут варьироваться от проекта к проекту.

Для избежания путаницы документы планов, разработка которых начинается на стадии выработки концепции, называются “подходами”. К примеру, подход к тестированию может быть кратко сформулирован во время фазы выработки концепции, а его превращение в план тестирования происходит на более поздних фазах.

Ранние базовые версии, отложенные итоговые версии

Создание базовых (предварительных) версий проектных документов на самых ранних этапах дает членам проектной группы возможность начать работу над своими задачами с минимальными задержками. Аналогично, отложенная на максимально долгий срок окончательная фиксация документов позволяет вносить в них жизненно важные изменения на протяжении всего этапа разработки. Но такая гибкость требует очень внимательного отношения к контролю за изменениями. Очень важно обеспечить их надлежащий мониторинг и исключить возможность несанкционированного изменения проектной документации.

Ежедневные билды

MSF рекомендует как можно чаще собирать текущие версии всех компонент решения для проведения тестирования и анализа. Этот подход применим как к разработке программного кода, так и к созданию компонент аппаратного и программного обеспечения. Его достоинство заключается в обеспечении стабильности решения на широком спектре тестовых данных еще до выхода решения в свет.

Большие, сложные проекты часто разделяются на меньшие подпроекты, каждый из которых разрабатывается и тестируется отдельной командой и затем вливается в общее решение. Для таких проектов, типичных в практике разработки продуктов Майкрософт, ежедневные билды являются фундаментальной, неотъемлемой составляющей рабочего процесса. В первую очередь разрабатывается базовая функциональность решения, и лишь затем создаются дополнительные его возможности. Разработка и тестирование ведутся непрерывно и одновременно, представляя собой параллельные процессы. Ежедневные билды служат верификацией совместимости всего разработанного программного кода и позволяют группам направлений продвигаться в своей работе к следующим итерациям разработки и тестирования.

Заметим, что ежедневные билды решения не попадают в реальную производственную среду. Лишь после того, как решение хорошо оттестировано и стабилизировано, осуществляется его пилотный (или бета) выпуск, частично устанавливаемый в производственную среду.

Для надлежащей синхронизации билдов необходимо иметь в проекте четко организованный механизм управления конфигурациями.

Управление конфигурациями проекта

Управление конфигурациями проекта – это *формализованный* процесс мониторинга и контроля за состояниями (версиями) различных элементов, таких как программный код, документация, руководства пользователей, файлы помощи, планы и календарные графики. Процесс управления конфигурациями также включает в себя мониторинг состояния аппаратного обеспечения, сетей и программных настроек решения. Проектная группа должна иметь возможность в случае необходимости осуществить возврат к более ранней конфигурации решения.

Управление конфигурациями часто путают с управлением изменениями в проекте, обсуждаемым ниже. В действительности эти две задачи взаимосвязаны, но не идентичны. Управление конфигурациями – это протоколирование и контроль состояний элементов проекта. Управление же изменениями – это процесс рассмотрения и одобрения проектных

изменений. Управление конфигурациями обеспечивает проектную группу инструментами, необходимыми для эффективного управления изменениями.

Например, проектная группа работает над электронной системой вызовов медицинской помощи, связывающей сеть больниц. Она сохраняет настройки, выбранные для сервера Microsoft BizTalk, и отслеживает изменения, производимые в ходе разработки и тестирования. Это пример управления конфигурацией. Затем, следуя недавно принятому правительственному нормативному акту, вносится предложение дополнить систему новой схемой электронного обмена данными. Руководители проектной группы встречаются со спонсором и членами команды сопровождения, чтобы проанализировать предложенные изменения, оценить их технологические риски и влияние на стоимость и календарный график проекта. Это пример контроля за изменениями.

В организациях, использующих MSF, для управления конфигурацией проекта может быть задействован широкий спектр процессов управления, применяемых в сопровождении решений.

Рекомендации для выпуска версий решения

Выпуск версий решения помогает укрепить взаимоотношения между проектной командой и заказчиком и обеспечить реализацию в решении всех лучших идей. Заказчику будет легче согласиться с переносом реализации определенных возможностей на более поздние версии, если у него есть доверие к проектной группе, основанное на своевременной поставке начальной и последующих версий решения. Вот общие рекомендации, облегчающие выпуск промежуточных версий решения:

- Создавая планы, предусматривайте версионирование.
- Прежде всего, поставляйте базовую функциональность.
- Выбирайте приоритеты, учитывая риски.
- Осуществляйте частые итерации разработки.
- Институциируйте процедуры контроля изменений в проекте
- Не создавайте новых версий, если они не увеличивают ценность решения.

Заблаговременное планирование последующих версий позволяет проектной группе более четко прояснить, что должно быть создано сейчас, и с чем можно повременить. Создание долгосрочного графика введения в решение новой функциональности дает возможность использовать доступные ресурсы и время наилучшим образом и избежать нежелательного расползания рамок проекта.

Предоставление заказчику простейшего, компактного, но отвечающего основным нуждам решения имеет большую ценность, чем разработка наиполнейшей версии, которая будет изготавливаться недели, месяцы, годы... Внедрив базовую функциональность, разработчики получают прочный фундамент, на основе которого может быть продолжено развитие решения. Проектная группа также получает отзывы от заказчика, помогающие выбирать наиболее адекватный путь эволюционирования решения.

Проведение проектной группой оценивания рисков позволяет выявить, реализация какой функциональности сопряжена с наибольшими из них. Подробная информация о рисках может быть получена из “Белой книги” дисциплины управления рисками MSF.

В первую очередь планируйте реализовать наиболее рискованные нововведения и изменения и только после них – менее рискованные. Например, задачи, связанные с большими изменениями в архитектуре, имеет смысл выполнить на ранних этапах проекта, минимизируя тем самым влияние этих изменений на бюджет и календарный график.

Существенный выигрыш от версионирования решения заключается в предоставлении заказчику в адекватные сроки пригодного к эксплуатации решения, которое затем последовательно улучшается. Если в этом процессе происходят задержки, то надежды заказчика на постоянное улучшение продукта оказываются неоправданными.

Поэтому ограничивайте рамки каждой итерации решения, чтобы создание новой версии происходило в приемлемые сроки.

Как только завершено создание базовой версии спецификации, должен быть незамедлительно введен контроль за изменениями планируемой функциональности решения. Принципиально важно, чтобы вся проектная группа и заказчик четко понимали значение такого контроля и его процедуры.

MSF не предписывает конкретный набор правил контроля за изменениями. В зависимости от размера и типа проекта, эти процедуры могут быть простыми или сложными. Однако для обеспечения эффективности контроля необходимо наличие следующих элементов:

- Без анализа и одобрения как со стороны проектной группы, так и со стороны заказчика запланированная функциональность не расширяется и не изменяется.
- В целях облегчения проведения анализа запросы на изменение функциональности подаются в письменном виде. Это позволяет группировать подобные запросы. В Майкрософт они называются запросами на изменение дизайна.
- Должны быть проанализированы влияние, осуществимость и приоритет каждого запроса на введение новой функциональности. При этом должны быть учтены взаимосвязи с другими составляющими решения, включая пользовательскую документацию, документацию сопровождения, учебные материалы и производственную среду.
- Для каждого запрашиваемого изменения должно быть определено его влияние на бюджет и календарный график проекта (детали см. в разделе “Оценка снизу вверх”).
- Должны быть определены члены группы контроля за изменениями, утверждающей или отвергающей запрошенные изменения. В эту группу должны входить представители заказчика, ролевого кластера “Управление программой” и прочие представители проектной группы и других заинтересованных сторон. Группа контроля за изменениями может формироваться различным образом, но, в любом случае, она должна иметь полномочия вносить изменения в бюджет, календарный график и функциональность решения.
- Должен вестись мониторинг изменений, и все его результаты должны быть легко доступны. Например, хорошей практикой является создание в функциональной спецификации и других важных проектных документах секции для протоколирования изменений.
- Эффективное управление изменениями возможно только при эффективном управлении конфигурациями.

Как уже было отмечено ранее, решение не представляет бизнес-ценности, пока оно не внедрено. Именно по этой причине модель процессов MSF содержит весь жизненный цикл создания решения, включая его внедрение – вплоть до момента, когда решение начинает давать отдачу.

Преимущества интегрированной модели процессов

Модель процессов, интегрирующая разработку и внедрение решения, имеет следующие преимущества:

Сосредоточение на нуждах предприятия

Предприятия (и в особенности руководители предприятий) обычно рассматривают разработку и внедрение решения как нечто неразделимое. Даже если разработка решения прошла удачно, заказчики не увидят отдачи до тех пор, пока оно не внедрено на предприятии.

Улучшенная поддержка разработки веб-приложений

Для сегодняшних команд разработчиков веб-приложений создание и внедрение веб-сайта является единым, скоординированным процессом.

Улучшенная поддержка веб-сервисов

Неотъемлемой частью проектирования и разработки веб-сервисов является их немедленная установка на веб-серверах. Так как веб-сервисы все чаще становятся основными каналами поставки программного обеспечения, разработчики коммерческого программного обеспечения находят необходимым включение процесса установки в жизненный цикл продукта.

Улучшение взаимодействия с командой сопровождения

Зачастую команда разработчиков создает решение, не уделяя должного внимания вопросам его эксплуатации. Это приводит к низким показателям производительности, доступности и управляемости решения. Интегрированная модель процессов MSF обеспечивает процесс передачи ответственности от команды разработчиков к команде сопровождения сквозь ряд последовательных вех, а не как одномоментный перенос нагрузки.

Замечания об использовании интегрированной модели процессов

Длительность фаз не одинакова

Хотя на диаграммах все фазы проекта схематически изображаются секторами одинаковых размеров, это не означает, что все они занимают одно и то же время. На практике длительность каждой фазы может значительно меняться в зависимости от проекта.

Деятельность может выходить за границы одной фазы

Новички в MSF иногда полагают, что всякая проектная деятельность привязана к определенной фазе и осуществляется лишь в ее рамках. На самом деле это не так. Например, планирование осуществляется не только на фазе планирования, тестирование выходит за рамки фазы стабилизации, разработка может продолжаться после завершения фазы разработки. Фазы характеризуются, в первую очередь, своими целями и результатами и, в меньшей степени, видами деятельности, на которых фокусируется проектная группа.

Создание, обновление и уточнение планов происходит на протяжении всего жизненного цикла проекта. Однако пик деятельности по планированию проекта приходится на фазу планирования, и главные документы планов создаются во время этой фазы.

Проекты, ограниченные разработкой приложения или внедрением инфраструктуры

Некоторые проекты состоят лишь из задач разработки, другие – только из задач внедрения. Поставщики коммерческого программного обеспечения изготовляют “коробочные” программные продукты, установку которых они не осуществляют (хотя они должны хорошо представлять себе все её аспекты). Аналогично, команды проектов по развертыванию инфраструктуры не разрабатывают внедряемые ими технологии, хотя некоторое программирование, конечно же, имеет место и в этом случае (например, написание скриптов автоматической установки).

Поэтому команды, работающие над проектами, ограниченными лишь разработкой или внедрением, могут просто опускать те фазы и вехи, которые не встречаются в их проекте.

Вопросы для самопроверки.

1. Что определяет модель проектной группы?
2. Какие возможны совмещения ролей?
3. Назовите цели задачи ролей проектной группы.
4. Для чего применяется модель проектной группы?
5. В чем суть подхода, основанного на вехах?
6. Перечислите принципы и особенности модели процессов MSF?