## Групповая разработка программного обеспечения компьютерных систем

**Разделение труда в коллективе разработчиков программ**

Проблема разделения труда при разработке программ возникла, прежде всего, в связи с возрастанием сложности и объема создаваемого программного продукта. Один человек даже исключительных способностей оказался не в состоянии создавать сложные программные комплексы в приемлемые сроки. Возникла необходимость более совершенной формы организации труда. Новая организация труда должна была обеспечить коллективное создание программной системы. При этом должно быть реализовано не только продуманное разделение труда, но и взаимодействие участников разработки друг с другом. Работа такого коллектива должна быть хорошо управляемой. Это означает возможность планирования работы каждого исполнителя и всего коллектива, объективное нормирование труда, легкость контроля над ходом работ по содержанию и потребляемым ресурсам. Немаловажное значение имеет всестороннее стимулирование участников разработки к повышению производительности труда, сокращению сроков и обеспечению высокого качества разрабатываемой системы. Здесь технические вопросы методологии выполнения работ тесно переплетаются с организационными решениями.

Как и в вопросах технической методологии, при организации труда программистов нельзя предложить универсального решения, пригодного для любых условий. Рассмотрим два отличающихся варианта:

* групповую организацию работ, ориентированную на единичное производство уникальных программных систем,
* и организацию, напоминающую работу производственного участка, рассчитанную на массовое создание однотипных программ.

Сущность групповой организации состоит в том, что для разработки программной системы создается временная (на период этой разработки) группа. В этой группе, небольшой по численности, предусмотрено четкое функциональное разделение труда. Для выполнения работы предоставляются хорошо продуманные средства труда, носящие универсальный характер относительно содержания разрабатываемой программной системы. Средства труда, включающие и методологию проектирования, обеспечивают и предметное разделение труда. В рамках групповой организации были предложены методы организации взаимодействия и контроля, имеющие достаточно универсальный характер и пригодные для использования и при других организационных формах. При групповой организации предполагается достаточно высокая квалификация всех сотрудников группы.

Во главе группы стоит главный программист, имеющий высшую квалификацию и решающий все основные технические вопросы. Полнота власти при решении технических и организационных вопросов в группе определяется полнотой ответственности за разрабатываемую систему. Можно сказать, что главный программист разрабатывает систему, а группа ему помогает.

Вторым обязательным сотрудником группы является заместитель руководителя с квалификацией, близкой к квалификации руководителя. Выполняя работы по проектированию и реализации программных модулей системы, он должен быть в курсе всех решений, принимаемых руководителем, и иметь возможность в любой момент заменить руководителя. Исполнители, число которых зависит от объема разрабатываемой системы, выполняют проектирование, написание и проверку отдельных программных модулей или групп связанных программных модулей. Количество исполнителей не должно быть более четырех-пяти человек.

Особую роль в группе выполняет библиотекарь. В его обязанности входит ведение всей машинной и немашинной документации группы, начиная от программных модулей на машинных носителях и кончая протоколами заседаний группы. Кроме того, библиотекарь берет на себя всю рутинную работу на ЭВМ, освобождая от нее основных разработчиков. Хотя библиотекарь и не выполняет творческой части работы, но от его профессиональных качеств в значительной степени зависит успех работы всей группы. При большом количестве исполнителей отдельным из них может быть поручено выполнение специальных функций. Например, инструментальщика, разрабатывающего вспомогательные средства автоматизации проведения разработки и отладки. Системщика, имеющего дело с операционной системой, трансляторами и другими применяемыми при работе средствами сторонней разработки. Редактора, занимающегося оформлением документации, и секретаря.

Для достижения высокой эффективности группа при разработке различных систем должна пользоваться одной и той же методологией. Отсюда следует, что это должна быть универсальная методология, не зависящая от содержательных особенностей систем. При проектировании это, прежде всего, эвристические методы структурирования и связанные с ними методы документирования, что еще раз подчеркивает необходимость высокой квалификации привлекаемых в группу разработчиков.

Разделение труда при разработке одной и той же программной системы может привести к ошибкам, связанным с недостаточным пониманием явных и скрытых взаимодействий модулей, разрабатываемых разными исполнителями. Кроме технических решений, позволяющих ослабить межмодульные связи, принимаются организационные меры, обеспечивающие активное взаимное информирование исполнителей. Для этих целей с успехом используются так называемые структурные просмотры.

Наряду с мерами обеспечения взаимодействия предусматриваются и методы контроля хода разработки. Контроль при групповой организации осуществляется в виде внешней инспекции по окончании каждого из этапов работы.

Оперативный контроль над ходом разработки должен быть легко реализуемым и объективным, не зависящим от чьих бы то ни было экспертных оценок. Разбиение системы на небольшие по трудоемкости модули позволяет оценивать состояние работы по количеству модулей, разработка которых полностью завершена. Особенно наглядная картина складывается при такой оценке на последнем этапе работы при написании и проверке программы. Полуфабрикаты, независимо от их состояния, рассматриваются как незавершенные работы. Сведения о завершенности работы над модулем исходят не от исполнителя, а от библиотекаря, помещающего готовые модули в отдельную библиотеку только после выполнения их полной проверки. Объективные сведения об объеме выполненных и оставшихся работ, и статистические данные о средних сроках разработки модуля каждым из исполнителей, позволяют делать достоверные прогнозы и своевременно управлять ресурсами.

В ряде случаев возникает необходимость организации массового производства однотипных программ в течение длительного периода. В этих случаях применение универсальной, эвристической технологии не оправдано. Сужение класса задач позволяет разработать более детальную методологию проектирования и реализации программы. Такая разработка требует высокой квалификации, но ее применение позволяет привлечь к работе программистов с гораздо меньшим опытом, знаниями и способностями, чем при групповой организации.

В реальной ситуации редко встречается какой-либо один тип программного производства в «чистом виде». Даже при массовом производстве возникает необходимость уникальных разработок как целевых, так и для развития и поддержания технологии.

**Коллективная  разработка программного обеспечения**

**1. Авторская разработка**

Авторская разработка — принцип создания программных продуктов, при котором весь жизненный цикл разработки поддерживается одним — единственным человеком.

Этот принцип  был достаточно  широко  распространен  в  70 — 80-е  годы ХХ века. Сейчас он применяется редко. Примерами авторских разработок являются операционная система Диспак (В. Ф. Тюрин), текстовый редактор Лексикон (Е. М. Веселов), трансляторы с языков Algol – 68 (П. Наур) и Pascal (H. Вирт).

Принцип авторской  разработки неприменим для многих современных  разработок из-за их сложности, объема и требований к качеству и сопровождению.

С другой стороны, программное обеспечение, начиная  с момента появления персональных компьютеров, стало продуктом массового потребления, приносящим огромный доход. В этой области быстро выросли и стали доминировать крупные компьютерные компании с развитой структурой менеджмента и мощной рекламой.

Наиболее интересен  принцип авторской разработки с  точки зрения применения в области наукоемких приложений. Для таких приложений характерна необходимость многолетнего изучения предметной области, практически полное отсутствие начального финансирования проекта, малая рентабельность, определяемая узким кругом пользователей.

Об интеллектуальных работниках

Отметим, что программисты относятся к  интеллектуальным работникам (knowledge workers). Так называют работников, которые могут создавать продаваемый продукт (и, следовательно, зарабатывать себе на жизнь) самостоятельно, какой-либо компании. Основным «рабочим оборудованием» таких специалистов (к их числу относятся также юристы и психологи) является их собственная голова.

Авторская работка  может выигрывать по производительности в 30 и более раз у коллективной разработки, что достигается за счет:

-        исключения межличностных коммуникаций, связанных с необходимостью порождения и изучения большого количества технологической документации;

-        исключения работ по разбиению проекта на составляющие, по распределению их между исполнителями, по координации деятельности исполнителей и контролю за их работой.

Объем программного продукта, выполненного методом авторской  разработки, в 5 — 20 раз меньше по сравнению  с индустриальными аналогами.

Авторская разработка предполагает достижение профессионального успеха, известности и славы в одиночку. Такое вполне реально, следует только правильно выбрать профессиональную «нишу», область ведения разработки.

О применении авторской разработки

В наибольшей степени авторская разработка в наши дни применяется при создании условно-бесплатных программных продуктов (shareware).

**2. Коллективная разработка**

Одним из основных вопросов коллективной разработки является разделение труда — от равноправных соисполнителей до явного и безоговорочного  лидера (например, в случае бригады главного программиста).

**Технические командные роли**

**Иерархическая модель**

Известно, что  первые коллективные разработки программ велись пример так. Начальник выполнял разделение большого проекта на меньшие  части и передавал далее по иерархии. Через некоторое время, теперь уже снизу ерх, шла сборка программы из написанных фрагментов. Заметим, что в итоге собрать работающий программный продукт удавалось не всегда.

**Матричная модель (равноправные соисполнители)**

Бригада равноправных соисполнителей обычно состоит из специалистов, занимающихся примерно одинаковыми задачами в рамках одного проекта, естественно, специализаций в рамках одной бригады может быть несколько.

Примерный состав такой бригады разработчиков, а  именно:

-        инженеры-разработчики  (специалисты по инженерии программирования и программисты);

-        технические писатели;

-        инженеры тестирования;

-        инженеры качества;

-        специалисты по сопровождению продукта;

-        специалисты по продажам продукта.

Тип работы определяет содержание и природу выполняемой  работы. Приведем список типов работ  и областей специализации на основе классификации Сью Конгер (Sue Conger).

Разработка приложений (ApplicationDevelopment):

•     программист (Programmer);

•     специалист по инженерии программирования (Software Design Engineer);

•     специалист по инженерии знаний (Knowledge Engineer).

Разработка  документации (Documentation):

•     технический писатель (Technical Writer). Тестирование продукта (Testing):

•     инженер тестирования (Tester);

•     инженер по разработке тестов (Test Design Engineer). Продажи (Sales):

•     консультант по продукту (Product Support);

•     специалист по маркетингу (Product Marketing).

Управление  разработкой (Management):

•      менеджер проекта (Project Manager);

•     менеджер информационных систем (IT Manager).

Разработка  аппаратуры (Hardware Design):

•     разработчик аппаратуры (Hardware Design Engineer).

Работа с  приложениями (Application Support):

•     специалист по приложениям (Application Specialist);

•     администратор данных (Data Administrator);

•     администратор базы данных (Database Administrator).

Техническая поддержка (Technical Specialists):

•     системный администратор (System Administrator);

•     сетевой администратор (Network Administrator);

•     администратор коммуникаций (Communications Administrator).

Системное интегрирование (System Integration):

•     системный интегратор (System Integrator).

Из перечисленных  специализаций очень интересна  специализация системного интегратора. Основные задачи системного интегратора — это предложить заказчику вариант решения его проблемы, выбрав наиболее приемлемый по цене и технике, и реализовать его. Таким образом, системный интегратор продает решения и несет ответственность за их реализацию. Системный интегратор как профессионал должен обладать знаниями очень многих областей: прикладное и системное программное обеспечение, администрирование систем, аппаратура, сети, экономика и т. п.

**Бригада главного программиста**

Миллз Брукс предложил организовывать команды (бригады) главного программиста(chief programmer teams), подобные хирургическим бригадам. Лишь один участник команды занимается основной работой, остальные оказывают ему всевозможную поддержку. Бригада главного программиста включает десять человек, выполняющих специализированные роли в команде.

Основные члены  бригады выполняют следующие  функции.

Главный программист. Лично выполняет анализ и проектирование, создание и отладку кода, написание документации. Должен обладать том, большим опытом работы и существенными знаниями.

Дублер. Может выполнять любую работу главного программиста, но нее опытен. Подстраховывает главного программиста, может заниматься написанием кода, но не несет ответственности за проект.

Администратор, он же менеджер. Под его контролем находятся деньги и люди, помещения, машинные ресурсы, контакты с другими группами и руководством.

Редактор. Фактически, это технический писатель. Его задача — критически переработать черновики документации, созданные главным программистом, снабдить их ссылками и библиографией и обеспечить публикацию или помещение в Интернете.

Языковед. Эксперт в тонкостях языков программирования. Может найти эффективные способы использования языка для решения сложных задач. Обычно работает с несколькими бригадами.

Инструментальщик. Разработчик специализированных инструментов — утилит и сценариев. Поддерживает основной инструментарий и оказывает по нему консультации. При необходимости может осуществлять администрирование операционной системы.

Отладчик. Разработчик тестов и организатор тестирования программного продукта.

Делопроизводитель. Отвечает за регистрацию всех технических данных бригады в библиотеке программного продукта. Благодаря делопроизводителю, активные программисты освобождаются от рутинных работ. Заметим, что в настоящее время функции делопроизводителя автоматизированы и переданы репозиторию проекта.

Рекомендуется создавать команды на основе бригады  главного программиста лишь для небольших  проектов. Отметим, что большие трудности могут возникать при поиске кандидатур на роль главного программиста, который должен быть весьма талантливым специалистом. Причем как программистом, так и техническим руководителем.

**Программирование в парах**

Программирование  в парах предусматривает разработку, при которой два человека в одно и то же время занимаются программированием одной задачи за одним компьютером, используя одну клавиатуру, одну мышь и один монитор. В каждой паре существуют две роли.

-        Первый партнер решает задачу непосредственной реализации одного методов наилучшим образом. Именно в его руках находится клавиатура и мышь.

-        Второй партнер решает стратегические задачи:

•     будет ли работать используемый подход в целом;

•     какими могут быть дополнительные тестовые случаи;

•     существуют ли способы упростить всю систему так, что текущая проблема просто исчезнет.

Состав пар  обычно меняется динамически, возможно несколько раз в день.

**Ядерная модель**

Ядерная модель предполагает наличие первого исполнителя, олицетворяющего ядро команды и создающего прототип системы. На основе прототипа командой разработчиков создается программный продукт. Наиболее сложным действием здесь является передача работы от исполнителя прототипа к команде, которая будет доводить работу до состояния программного продукта.

Психологические командные роли

Кроме технических  командных ролей следует разбираться  и в психологических ролях. С  одной стороны, важно понимать характеристики конкретно людей. Это может очень  пригодиться, когда надо иметь дело с определенным человеком, и именно для него придумывать способы мотивации, характеристики  должны  определяться  профессиональным  психологом, правило, на основе интерпретации тестов. Одним из традиционных тестов является Миннесотский стандартный многофакторный метод исследования личности — СМИЛ (Minnesota Multiphasic Personality Inventory — MMPI). По результатам тестирования можно прогнозировать и диагностировать или иные возможные проблемы или хотя бы области проблем. С другой стороны, важны характеристики людей в отношениях с внешним миром. Такая типология может быть полезной, когда надо решить, кто с какой работой лучше справится. Далее мы рассмотрим два примера командных характеристик.

Ключевые проектные роли

Томсет (Rob Thomsett) предложил восемь ключевых ролей проекте.

Председатель. Выбирает путь, по которому команда движется вперед к общим целям. Умеет обнаружить сильные и слабые стороны команды и обеспечить наибольшее применение потенциала каждого ее участника.

Архитектор. Он же оформитель. Придает законченную форму действиям команды. Имеет четкое представление о проблемах и их возможных решениях.

Генератор идей. Предлагает радикально новые идеи и стратегии, новые подходы к решению проблем, с которыми сталкивается группа. Особое внимание уделяет главным проблемам.

Критик. Он же скептик, оценивающий проблемы с прагматической точки зрения. Ищет недостатки, изъяны и недоделки. Компенсирует оптимизм генератора идей.

Исполнитель. Работник, собственно занимающийся написанием кода. Как правило, он не обладает широтой кругозора.

Завершающий. Поддерживает в команде настойчивость в достижении цели. Играет доминирующую роль на завершающих стадиях разработки.

Дипломат. Поддерживает силу духа в участниках проекта. Оказывает им помощь в трудных положениях. Пытается улучшить взаимоотношения в команде.

Организатор. Обнаруживает и сообщает о новых идеях, разработках и ресурсах. Имеет много друзей и связей в своей организации, с помощью которых можно выпросить или одолжить необходимые ресурсы.

В реальных командах программистов могут быть выделены не все из этих ролей. Роль исполнителя часто берут на себя сразу несколько членов команды.

Соционические роли

Существует  много методик для определения  психологического портрета человека, а также для помощи при формировании «почти идеальных» малых трудовых коллективов. Для создания и расширения программистских коллективов с максимально продуктивной деятельностью можно применять, например, интересные соционические разработки. При определении прогноза эффективности одной личности и степени совместимости внутри коллектива соционика предлагает использовать легкие и простые при обработке тесты.

Методика основана на выделении концентрированно выраженных и легко узнаваемых типов личности. Основы методики созданы швейцарским  психологом Карлом Густавом Юнгом (Carl G. Jung), а развита и усовершенствована она  была в работах литовской исследовательницы    Ayшры Аугустинавичуте.

Экстраверт, по определению Юнга, — это человек, чья деятельность направлена на объект и определяется этим объектом. Такой человек имеет тенденцию к направленному взаимодействию с внешней средой. Интровер ориентируется на свою оценку предмета или события, а не на объект как таковой. Юнг выделил четыре базовых области восприятия: материя, энергия, пространство и время (иначе — логика, этика, сенсорика, интуиция). Им соответствуют следующие типы личности: мыслительный, эмоциональный, ощущающий и интуитивный.

В соционике  принято выделять восемь основных функций,  по которым сознание человека связано с окружающим его миром.

-        Логика: деловая (экстраверт), отношений (интроверт).

-        Этика: эмоций (экстраверт), отношений (интроверт).

-        Сенсорика: волевая (экстраверт), ощущений (интроверт).

-        Интуиция: возможностей (экстраверт), времени (интроверт).

В каждом типе выделяется два подтипа: мыслительный и эмоциональный, которые имеют сенсорную или  интуитивную ориентацию, и наоборот.

На основе знания типа личности могут быть сделаны  рекомендации о выборе сферы- деятельности (профессиональная ориентация).

Выделим четыре типа личности, наиболее склонных к работе, связанной с программированием.

-        Интуитивно-логический   экстраверт —   »впередсмотрящий»,   импульсивный стратег, склонный к синтезу (характерный представитель типа Александр Суворов).

-        Логико-интуитивный интроверт — сильный логист, имеющий аналитический характер мышления, стремление к выявлению системообразующих факторов (характерный представитель типа — Робеспьер).

-        Интуитивно-логический интроверт — критик, стратег, генератор новых алгоритмов, знаток динамических структур (характерный представитель типа — Оноре де Бальзак).

-        Логико-интуитивный экстраверт — активный инициатор, изобретатель и рационализатор (характерный представитель типа — Джек Лондон).

**Типы  совместной деятельности**

Коллективная  разработка предполагает большое количество различных действий, причем степень  совместной деятельности может существенно  изменяться от одного действия к другому. Можно выделить четыре типа совместной деятельности[Robillard, Robillard 2000].

Мандатная деятельность, обычно представленная формальными собраниями, проводимыми на регулярной основе. Обычно собрания планируются заранее, а присутствие на них обязательно. Статистика показывает, что программисты проводят около 4% своего рабочего времени на собраниях.

Созываемая деятельность, которая имеет место в случае намерения двух или более программистов собраться вместе для решения некоторого технического вопроса. Такие собрания обычно не планируются заранее, и в них участвуют только действительно заинтересованные в решении проблемы программисты. На эту деятельность уходит около 14% рабочего времени.

Естественная  совместная деятельность, когда как минимум двое программистов работают над одной и той же задачей одновременно и обмениваются информацией о выполняемой работе. Эта деятельность занимает около 41% рабочего времени.

Индивидуальная деятельность, когда программист работает над задачей, которая не выполняется в то же самое время никаким другим программистом, и поэтому маловероятно его взаимодействие по этому предмету с любыми другими программистами группы. Эта деятельность занимает также около 41% рабочего времени.

**Общинная модель разработки**

Идеология общинной («базарной») модели разработки сформулирована в программной статье Эрика Раймонда (Eric Raymond) «Собор и Базар». Общинная модель характеризуется тремя основными факторами.

-        Децентрализованность разработки. Не существует ограничения сверху на количество людей, принимающих участие в проекте. Как правило, разработки такого типа ведутся на базе сети Интернет и могут включать любого заинтересованного разработчика.

-        Разработка на базе открытых исходных текстов. По ним можно разобраться с сутью задачи и в любой момент подключиться к разработке.

-        Большое количество внешних тестеров (бета-тестеров), позволяющих быстро обнаруживать ошибки и проблемы в программе.

Эрик Раймонд  сформулировал несколько уроков, которые позволяют лучше понять особенности общинной разработки.

-        Каждая хорошая программа начинается с энтузиазма разработчика.

-        Хорошие программисты знают, что можно написать, а великие — можно переписать.

-        При правильном отношении интересная проблема найдет вас сама.

-        Когда вы теряете интерес к программе, ваша последняя обязанность передать ее компетентному преемнику.

-        Следует выпускать ранние и частые версии программ. П   Обнаружить проблему и исправить ее могут разные люди.

-        Иногда использовать идеи пользователей лучше, чем свои.

В сети  Интернет можно найти достаточно большое количество сайтов спроектами,      разрабатываемыми      по     общинной     модели.

Отступление «об оффшорном программировании»

Оффшорное программирование — это выполнение внутренних работ компании сторонними специалистами, вне ее офиса и, как правило, на территории другой страны. Эта разновидность коллективного программирования получила известность в начале 1990-х годов. Формы предоставления услуг оффшорного программирования прошли следующие этапы эволюции.

-        Аутстаффинг — использование  программистов  »поштучно» для   конкретной работы под руководством менеджера заказчика.

-        Аутсорсинг — передача исполнителю разработки отдельных модулей компонентов систем, полная сборка которых производится заказчиком.

-        Полная разработка — выполнение проекта по полной разработке и внедрению системы.

Довольно широкое  распространение оффшорного программирования в настоящее  время  обусловлено  состоянием  мирового  рынка заказного  программного обеспечения. По очень приблизительным данным разработкой программного обеспечения в мире занято от 7 до 20 миллионов человек.

В России — от 5 до 10 тысяч. В мире существует огромный неудовлетворенный спрос на услуги профессионального программирования (например, в США дефицит профессиональных программистов составил в 2003 году около 1,5 миллионов). Следовательно, для некоторых компаний передача части работ исполнителям в другой стране обусловлена естественной необходимостью.

Поскольку стран, в которые могут быть переданы оффшорные заказы, достаточно много, мы можем сформулировать для них три основных желательных условия.

-        Оплата профессионалов ниже, чем в стране-заказчике.

-        Присутствуют высокие стандарты образования и доступны технические эксперты.

-        Доступны передовые технологии, повышение технической квалификации.

Легко видеть, основные страны, удовлетворяющие данным условиям, — Россия, Индия, Китай. Среди лидеров  оффшорного программирования уже находятся  Уругвай и Израиль.

Некоторый идеальный  портрет оффшорного программиста и оффшорной компании может выглядеть следующим образом.

Высокое качество работы каждого сотрудника. В оффшорных  компаниях в первую очередь «люди решают все». На уровне компании должны быть созданы системы отбора и оценки сотрудников, системы карьерного роста, обучения, мотивации, социальной защиты и решения многих других задач.

Высокий уровень  ведения проекта.

В основе любого проекта лежат несколько ключевых процессов, грамотное выполнение которых должен обеспечить менеджер проекта.

Оптимальная структура управления компанией. Такая структура подразумевает возможность быстрого формирования и переформирования команды, а также отсутствие жесткой и сложной иерархии.

В завершении краткого обзора оффшорного программирования укажем положительные и отрицательные стороны создания оффшорных компаний в России. Конечно, положительных сторон много, и они достаточно хорошо известны. В первую очередь, это фундаментальная подготовка, позволяющая браться за большие проекты. Немаловажную роль может играть территориальная близость и близость культурных сред. Слабые стороны мы явно подчеркнем в следующих тезисах.

-        Небольшое количество профессионально подготовленных менеджеров, способных грамотно управлять программными проектами. Большинство компаний, передающих заказы, считает это основной проблемой, препятствующей резкому росту оффшорного программирования в России.

-        Недостаточное знание естественного языка той страны, из которой поступает работа. Последнее время ситуация начинает меняться к лучшему благодаря росту интереса к иностранным языкам в школах и их грамотному преподаванию.

-        Большие сложности поездок в страну фирмы, передавшей заказ, известно много случаев отказа в выдаче виз по надуманным причина инженерам, выезжающим на короткий срок для осуществления консультаций и участия в совещаниях рабочих групп. Особенно «славятся» такой политикой США.

-        Относительно высокая стоимость качественных телекоммуникационных услуг. Это может стать причиной дополнительных расходов.

-        Низкий уровень сертификации на соответствие стандартам качеств Для ряда зарубежных компаний наличие такого сертификата у российской компании, в которую передается заказ, является обязательным.

-        Повышенная осторожность зарубежных клиентов при взаимодействии с российскими компаниями. В ряде случаев эта осторожность обусловлена историческими и политическими причинами. В других случаях — предыдущим неудачным опытом, связанным, например, с традиционным российским упованием на «авось».

### Закрепление функций и полномочий в проекте

Ранее в разделе, посвященному формированию устава проекта, мы уже кратко описывали характер ролей и основные должностные инструкции членов группы управления проектом. В данном же разделе мы подробно рассмотрим этот аспект и сформируем более детальный *список* функций и *полномочий* для каждой из упомянутых ниже ролей.

**Куратор проекта (спонсор)** - *проектная роль* должностного лица, отвечающего за стратегическое управление ходом реализации проекта. Куратор принимает решение по стратегическим вопросам проекта, осуществляет утверждение основных изменений в объеме *работ*, сроках, этапах, в бюджете проекта, находящихся вне компетенции руководителя проекта. Как правило, куратором проекта (спонсором) является *менеджер* высшего звена организации.

Основные функции:

* общее руководство ходом реализации проекта;
* обеспечение выделения необходимых ресурсов для выполнения проекта, обеспечение финансирования работ;
* рассмотрение и утверждение регламентирующих документов, необходимых для организации и выполнения проекта;
* получение и анализ сводной отчетности о ходе реализации проекта;
* управление изменениями базовых параметров проекта и решение проблем, находящихся вне компетенции руководителя проекта.

Основные *полномочия*:

* утверждение целей проекта;
* согласование назначения руководителя проекта;
* утверждение общего плана и бюджета проекта;
* получение от руководителя проекта сводной отчетности о ходе его выполнения;
* принятие принципиальных решений при возникновении критических изменений, влияющих на сроки, стоимость и качество результатов проекта.

**Руководитель проекта** - *проектная роль* *должностного лица, ответственного за управление проектом*.*Руководитель проекта* непосредственно отвечает за достижение целей проекта в рамках выделенного бюджета, в соответствии с плановыми сроками осуществления проекта и с заданным уровнем качества.

Основные функции:

* формирование команды проекта и команды управления проектом;
* планирование, организация и контроль выполнения работ по достижению целей проекта с требуемыми качеством, затратами и в заданный срок;
* распределение ресурсов проекта и организация взаимодействия команды проекта в процессе его выполнения;
* организация взаимодействия с заказчиком и обеспечение всех необходимых коммуникационных связей с другими участниками проекта;
* учет фактических затрат ресурсов по исполнению проекта;
* формирование и предоставление куратору отчетности по проекту.

Основные *полномочия*:

* назначение задач команде проекта (отдельным ее членам) и контроль их выполнения;
* требование от команды проекта выполнения своих ролевых функций;
* подтверждение или отклонение отчетов о фактических затратах исполнителей проекта;
* обоснование необходимости и запрос *куратору проекта* на выделение дополнительных ресурсов на проект;
* обращение к куратору за поддержкой в случае необходимости.

**Архитектор системы** - *проектная роль* должностного лица, отвечающего за предметную область проекта. *Архитектор* системы подчиняется непосредственно руководителю проекта.

*Архитектор* системы непосредственно отвечает за разработку информационной системы в соответствии с плановыми сроками проекта и с заданным уровнем качества.

На роль архитектора системы назначается специалист, наиболее компетентный по внедряемой информационной системе. *Архитектор* системы должен знать методологии и технологии построения ИС, стандарты и нормативные документы в области проектирования и создания ИС, разработки и оформления технической документации.

Основные функции:

* определение состава, продолжительности и технологии выполнения работ по разработке и внедрению информационной системы;
* определение ресурсов, которые необходимы для разработки и внедрения ИС в рамках, заданных условиями проекта;
* определение квалификационных требований и состава рабочих групп специалистов по направлениям деятельности, распределение их по задачам, организация работ и верификация результатов в процессе реализации проекта;
* обеспечение целостности функциональной архитектуры внедряемой информационной системы;
* организация подготовки, согласования и утверждения всей технической документации, необходимой для создания ИС в рамках проекта;
* планирование и согласование фактических трудозатрат специалистов при исполнении проекта;
* формирование и предоставление руководителю проекта необходимой отчетности;
* анализ хода выполнения и промежуточных результатов создания ИС;
* организация, проведение и документирование процедур передачи заказчику разработанной ИС.

Основные *полномочия*:

* участие в календарном планировании работ по созданию ИС;
* назначение задач рабочим группам проекта и контроль их выполнения;
* требование от исполнителей качественного выполнения порученных задач и своевременной информации о возникающих проблемах;
* обоснование необходимости и запрос руководителю проекта на выделение дополнительных ресурсов на проект.

**Администратор проекта** - *проектная роль* должностного лица, отвечающего за информационное обеспечение руководителя проекта, организацию и ведение документооборота по проекту.*Администратор* проекта функционально закрепляется за конкретным проектом и подчиняется непосредственно руководителю проекта.

Основные функции:

* обеспечение руководителя проекта структурированной информацией, дающей возможность контроля проекта, планов, ресурсов и приоритетов;
* ведение протоколов совещаний;
* обеспечение своевременной подготовки, движения и архивации документов по проекту.

Основные *полномочия*:

* передача и получение от участников проекта необходимой документации по проекту;
* контроль соблюдения участниками проекта установленной системы документооборота;
* требование от конкретных исполнителей по проекту оперативной информации и отчетов о ходе работ по проекту.

Для того чтобы закрепить функции и обязанности по проекту, составляют ролевые инструкции или положение по *проектной роли*. В ролевой инструкции должно быть определено следующее:

* какие цели стоят перед сотрудником, назначенным на данную роль;
* кому подчиняется сотрудник, назначенный на ту или иную роль;
* каковы его функции, обязанности, *полномочия*.

Крайне важное замечание, высказываемое многими экспертами, состоит в том, что *определение* ролей и ответственности в проекте должны производиться с учетом факторов внешней среды предприятия. В [табл. 6.3](http://www.intuit.ru/studies/professional_retraining/963/courses/502/lecture/11398?page=2#table.6.3) приведены примеры возможного влияния организационных, технических, межличностных, политических и прочих факторов на *процесс планирования* команды проекта.

|  |  |
| --- | --- |
| Таблица 6.3. Влияние факторов внешней среды на планирование команды проекта | |
| Факторы внешней среды | Влияние на определение ролей команды и ответственности |
| Организационные | Взаимоотношения организаций или отделов, участвующих в проекте, механизмы взаимодействия между ними |
| Технические | Навыки и специальности, необходимые для выполнения проекта, необходимость обеспечения координации между языками программного обеспечения, наличие специфических сложностей при переходе от одной фазы жизненного цикла к другой |
| Межличностные | Официальные и неофициальные отношения между потенциальными членами команды проекта, их должностные обязанности. Культурные или языковые различия между членами команды, которые могут оказать влияние на их рабочие взаимоотношения |
| Политические | Цели и интересы потенциальных членов команды проекта, люди (или группы людей), которые имеют неформальное влияние в областях, представляющих важность для проекта, существование неформальных связей между потенциальными участниками проекта |

На этапе планирования для каждой роли должен быть определен *список* навыков, необходимых членам команды проекта. Для разработки списка рекомендуется использовать *реестр навыков* -*список* категорий и компонентов навыков для определенного класса команды исполнителей проекта (см.[табл. 6.4](http://www.intuit.ru/studies/professional_retraining/963/courses/502/lecture/11398?page=2#table.6.4)).

Для обеспечения анализа совокупностей навыков компоненты группируются в четыре категории: технические навыки, административные, навыки межличностного общения, стратегические навыки. Для каждого навыка отмечаются *рейтинг* критичности и *рейтинг* способностей. Для оценки рейтинга принято использовать 4-балльную шкалу (см. [табл. 6.5](http://www.intuit.ru/studies/professional_retraining/963/courses/502/lecture/11398?page=3#table.6.5)).

|  |  |
| --- | --- |
| Таблица 6.4. Реестр навыков для команды исполнителей проекта | |
| Категории и компоненты навыков | |
| Технические навыки  Умение управлять проектом и его технологией  Оказание помощи в разрешении проблем проекта  Взаимодействие с техническим персоналом  Участие в достижении компромиссов  Понимание тенденций  Понимание основных задач маркетинга  Наличие навыков системного анализа | Навыки межличностного общения и лидерства  Оказание помощи в решении проблем  Построение многофункциональной команды  Определение целей  Получение поддержки высшего руководства  Мотивация членов команды  Управление конфликтами |
| Административные навыки  Привлечение уникальных специалистов  Навыки эффективного общения  Умение делегировать полномочия  Ведение переговоров с целью обеспечения ресурсами  Календарное планирование  Понимание политик и рабочих процедур  Сотрудничество с другими проектными командами | Стратегические навыки  Стратегическое планирование  Принятие стратегических решений  Умение работать в условиях риска  Умение лидировать |

### Реестры навыков

*Реестры навыков* должны быть составлены для каждого класса персонала, как, например, для руководителя проекта, системного архитектора, специалиста по качеству. Критичность навыков для руководителя проекта определяется во многом масштабом проекта и организационной структурой проекта. Наибольшими *полномочиями* наделен *руководитель проекта* в проектных организационных структурах, и следовательно, к нему должны предъявляться самые высокие требования. *Список* навыков может быть определен на основе профессиональных стандартов в области информационных технологий. Распределение навыков зависит от уровня административной ответственности. *Рейтинг* критичности смещается от "технических" в сторону "административных", а затем в сторону "межличностного общения" и "стратегических навыков" по мере роста административной ответственности. Следует подчеркнуть важность навыков межличностного общения.*Команда* проекта может многократно снизить количество возникающих проблем и повысить взаимодействие сотрудников, если будет понимать настроение членов команды проекта, предвидеть их действия, внимательно выслушивать и признавать их мнение и решать их проблемы. Для управления командой проекта и работы в проекте такие навыки, как умение сопереживать, оказывать влияние, творческий подход к работе и способность облегчать работу группы, приобретают *значение* ценных активов. После того как реестр сформирован, он может быть использован с минимальной точной подстройкой к новой проектной ситуации

**Пример разработки** *реестра навыков*

Ниже ([табл. 6.5](http://www.intuit.ru/studies/professional_retraining/963/courses/502/lecture/11398?page=3#table.6.5), [табл. 6.6](http://www.intuit.ru/studies/professional_retraining/963/courses/502/lecture/11398?page=3#table.6.6)) выделены категории навыков для консультантов и менеджеров проектов: технические, административные, навыки межличностного общения, стратегические навыки. Для каждого консультанта (как при приеме на работу, так и при зачислении в команду проекта) необходимо проводить оценку навыков по шкале 1-4 ("плохо", "удовлетворительно", "хорошо", "отлично" соответственно)

Стоит подробно остановиться на том, каким образом оценивать технические компетенции консультантов компании. Это наиболее важные навыки, которые необходимо учитывать при планировании сотрудников на проекты.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблица 6.5. Шкала рейтингов критичности и способностей | | | | |
| Рейтинг | Критичность | Квалификация | | |
| 1 | Неважно/Маловажно | Отсутствие навыков / слабые навыки | | |
| 2 | Важно | Базовые навыки | | |
| 3 | Очень важно | Высокая квалификация | | |
| 4 | Критично для успеха проекта | Уникальная квалификация | | |
| Таблица 6.6. Реестр навыков для членов команды исполнителей | | | | | |
| Категории и компоненты навыков | | | Критичность | ФИО | |
| Технические навыки (категория I)  Специальные знания SAP ERP HCM  Оказание помощи в разрешении проблем  Взаимодействие с техническим персоналом  Облегчение достижения компромиссов  Интеграция технических, деловых и человеческих целей  Системное мышление  Понимание технологий и трендов (тенденций)  Понимание прикладных задач маркетинга и применение продукта  Сплочение технической команды | | | Очень важно | Хорошо | |
| Административные навыки (категория II)  Способность к эффективному общению (устному и письменному)  Способность к эффективному делегированию обязанностей (от старших к младшим)  Минимизация изменений  Понимание политик и рабочих процедур | | | Важно | удовлетворительно | |
| Навыки межличностного общения и лидерства (категория III) Навыки общения  Легко понимает клиента, нравится ему  Последователен  Не принуждает к совершению тех или иных действий  Помогает обдумывать и принимать решения  Не подменяет свои решения клиентскими  Честность, способность признавать ошибки  Предлагает аргументы, а не просто готовые решения  Оптимизм, умение оказать положительное влияние  Чувство юмора  Владение рядом тактик убеждения  Урегулирование конфликтов  Командная работа и сотрудничество: взаимодействие с другими работниками и создание команды  Понимание профессиональных нужд | | | Очень важно | хорошо | |
| Стратегические навыки (категория IV)  Построение альянсов, коалиций и достижение сотрудничества  Способность работать в условиях рисков и неопределенностей  Мотивирование и вдохновление других  Стратегическое мышление, планирование и принятие решений  Понимание бизнес-окружения  Дальновидность | | | В некоторой степени важно | отлично | |

|  |  |  |
| --- | --- | --- |
| Таблица 6.7. Реестр навыков члены команды управления проекта | | |
| Категории и компоненты навыков | Критичность | Способности |
| Технические навыки (категория I)  Знание SAP ERP HCM  Оказание помощи в разрешении проблем  Взаимодействие с техническим персоналом  Облегчение достижения компромиссов  Интеграция технических, деловых и человеческих целей  Понимание технологий и трендов (тенденций)  Понимание прикладных задач маркетинга и применение продукта  Сплочение технической команды | В некоторой степени важно |  |
| Административные навыки (категория II)  Привлечение и удержание работников высокого класса  Способность к эффективному общению (устному и письменному)  Способность к эффективному делегированию обязанностей  Оценивание ресурсов и ведение переговоров с целью их получения  Измерение состояния и хода исполнения работ и производительности  Календарное планирование многодисциплинарных операций  Понимание политик и рабочих процедур  Работа (сотрудничество) с другими организациями | Важно |  |
| Навыки межличностного общения и лидерства (категория III)  Поощрение развития способностей других людей с помощью отзывов и наставлений  Способность инициировать преобразования, совершенствовать методы управления  Урегулирование конфликтов  Ориентация на действия  Оказание помощи при принятии групповых решений  Оказание помощи в решении проблем  Построение многофункциональных команд  Обеспечение вовлеченности персонала на всех уровнях  Формирование перспективной точки зрения  Доверие  Определение четких и ясных целей  Управление конфликтами  Мотивация людей  Понимание профессиональных нужд  Понимание организации | Очень важно |  |
| Стратегические навыки (категория IV)  Построение альянсов, коалиций и достижение сотрудничества  Способность работать в условиях рисков и неопределенностей  Мотивирование и вдохновление других  Ведение переговоров о ресурсах и мобилизация ресурсов  Стратегическое мышление, планирование и применение решений  Понимание бизнес-окружения  Дальновидность | Важно |  |

В консалтинговых компаниях, в том числе системных интеграторах, довольно сильно распространена практика разделения консультантов на так называемые грейды, или уровни, в соответствии с их технической компетенцией и возможностями выполнения тех или иных задач1 Пример четко сформулированных требований к грейдам представлен в сводной [табл. 6.8](http://www.intuit.ru/studies/professional_retraining/963/courses/502/lecture/11398?page=3#table.6.8) - в ней приведен пример описания технической компетенции консультанта по внедрению модуля управления человеческими ресурсами в системе *SAP*.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица 6.8. Реестр технических компетенций | | | | | |
| Уровень 1 | Вес | Уровень 2 | Вес | Уровень 3 | Вес |
| Компоненты НСМ-1 | 70 | Администрирование персонала | 12 | Бизнес-процессы | 2 |
|  |  | Инфотипы | 1 |
|  |  | Мероприятия | 2 |
|  |  | Стажи | 2 |
|  |  | Отчетность | 4 |
|  |  | Интерфейсы | 1 |
| Управление временными данными | 10 | Бизнес-процессы | 1 |
|  |  | Графики | 0,3 |
|  |  | Отсутствие, присутствие | 1 |
|  |  | Лимиты | 1,5 |
|  |  | Временные события | 1 |
|  |  | Оценка времени | 1,5 |
|  |  | Рабочий стол менеджера | 0,5 |
|  |  | Планирование смен | 0,7 |
|  |  | Сдельная оплата труда | 2 |
|  |  | Отчетность | 1 |
| Расчет заработной платы | 21 | Бизнес-процессы | 1 |
|  |  | Инфотипы | 1 |
|  |  | Расчет базовых видов оплат | 2 |
|  |  | Расчеты по среднему | 2 |
|  |  | Налоги | 2 |
|  |  | Удержания | 3 |
|  |  | Внециклические расчеты | 3 |
|  |  | Перечисления | 1 |
|  |  | Проводки | 3 |
|  |  | Отчетность | 3 |
| Организационный менеджмент | 5 | Бизнес-процессы | 0,5 |
|  |  | Стандартные объекты, инфотипы, связи | 1 |
|  |  | Интеграция с другими компонентами | 0,5 |
|  |  | Архитектура иерархии | 1 |
|  |  | Собственные объекты, инфотипы, связи | 1 |
|  |  | Версии плана. Статусы объектов | 0,5 |
|  |  | Отчетность | 0,5 |
| Льготы, предоставляемые работодателем | 3 | Бизнес-процессы | 1 |
| Инфотипы | 1 |
| Отчетность | 1 |
| Управление глобальными сотрудниками | 3 | Инфотипы | 0,5 |
| Процесс | 1 |
| Компенсационный пакет | 0,5 |
| Расчет заработной платы | 1 |
| Управления сотрудниками, имеющими несколько контрактов | 5 | Процессы администрирования | 1 |
| Управление временем | 1 |
| Льготы, предостав-ляемые работодателем | 0,5 |
| Расчет заработной платы | 2 |
| Проводки | 0,5 |
| Управление бюджетами должностей | 3 | Обязательства | 0,5 |
| Бюджеты | 0,5 |
| Интеграция с другими компонентами | 1 |
| Управление бюджетами | 1 |
| Управление командировками | 5 | Бизнес-процессы | 1 |
| Планирование | 1 |
| Командировочные расходы | 2 |
| Отчетность | 1 |
| Пенсионные фонды | 3 | Бизнес-процессы | 1 |
| Функции | 1 |
| Интеграция с другими компонентами | 0,5 |
| Отчетность | 0,5 |
| Программирование в НСМ-1 | 19 | Стандартная отчетность/SAP Query/BW | 2 | Использование стандартных отчетов | 1 |
| BW content для НСМ-1 | 0,5 |
| Расширения для SAP Query | 0,5 |
| Workflow в HCM-1 | 5 | Базовый процесс | 2 |
| Workflow в Администрирование персонала | 1 |
| Wbrkflow в управлении временными данными | 1 |
| Wjrkflow в управлении командировками | 1 |
| АВАР в НСМ-1 | 10 | АВАР workbench | 4 |
| User-exits, badis, includes, enhancements | 1 |
| АВАР репозиторий | 2 |
| MS Office integration (OLE, DPI), Adobe | 1 |
| ALV | 2 |
| Drilldown отчетность + HR forms | 2 | Создание drilldown отчетов | 1 |
| Создание Hrforms отчетов | 1 |
| Администрирование в НСМ-1 | 11 | Полномочия | 3 | Настройка ролей, полномочий | 1,5 |
| Структурные полномочия | 1 |
| Полномочия, зависимые от контента | 0,5 |
| ALE | 2 | Модель распределения | 1 |
| Создание, изменение idoc | 1 |
| CATS | 2 | Настройка CATS | 1 |
| Интеграция с использованием CATS | 1 |
| LSMW+SXDA | 2 | Batch input, direct input, BAPI | 1 |
| Выгрузка во внешние системы | 1 |
| Архивация данных | 1 | Процессы архивирования | 1 |
| Archive Link | 1 | Archive link | 1 |

В столбце "*Вес*" определено максимальное *значение* для навыка, исходя из общей значимости навыка для знания компонента в целом.

Следующим шагом требуется определить уровень сотрудников в разрезе каждого навыка и рассчитать общий коэффициент.

Далее, согласно технической компетенции, определяется грейд каждого сотрудника. Коэффициент определяется как суммарное *значение* всех весовых значений по каждому навыку.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Таблица 6.9. Пример оценки технических навыков членов команды исполнителей проекта | | | | | | | |
| Уровень 1 | Вес | Уровень 2 | Вес | Уровень 3 | Вес | Петров Иван | Сидоров Артур |
| Компоненты НСМ-1 | 70 | Администрировав ние персонала | 12 | Бизнес-процессы | 2 | 1,5 | 2 |
|  |  | Инфотипы | 1 | 0,7 | 1 |
|  |  | Мероприятия | 2 | 1,5 | 2 |
|  |  | Стажи | 2 | 1 | 1,7 |
|  |  | Отчетность | 4 | 1 | 3 |
|  |  | Интерфейсы | 1 | 1 | 1 |
| Управление временными данными | 10 | Бизнес-процессы | 1 | 0,8 | 1 |
|  |  | Графики | 0,3 | 0,4 | 0,5 |
|  |  | Отсутствие, присутствие | 1 | 0,8 | 1 |
|  |  | Лимиты | 1,5 | 1 | 1,4 |
|  |  | Временные события | 1 | 0 | 0,5 |
|  |  | Оценка времени | 1,5 | 0,8 | 1,2 |
|  |  | Рабочий стол менеджера | 0,5 | 0,2 | 0 |
|  |  | Планирование смен | 0,7 | 0.1 | 0,7 |
|  |  | Сдельная оплата труда | 2 | 0,05 | 1,5 |
|  |  | Отчетность | 1 | 0,6 | 1 |
| Расчет заработной платы | 21 | Бизнес-процессы | 1 | 0,8 | 1 |
|  |  | Инфотипы | 1 | 0,8 | 1 |
|  |  | Расчет базовых видов оплат | 2 | 1 | 2 |
|  |  | Расчеты по среднему | 2 | 1,8 | 2 |
|  |  | Налоги | 2 | 1 | 1,5 |
|  |  | Удержания | 3 | 1 | 1,7 |
|  |  | Внециклические расчеты | 3 | 0,5 | 1 |
|  |  | Перечисления | 1 | 0,3 | 1 |
|  |  | Проводки | 3 | 1 | 3 |
|  |  | Отчетность | 3 | 1 | 2 |
| Организационный менеджмент | 5 | Бизнес-процессы | 0,5 | 0,4 | 0,5 |
|  |  | Стандартные объекты, инфотипы, связи | 1 | 0,3 | 0,7 |
|  |  | Интеграция с другими компонентами | 0,5 | 0,2 | 0,4 |
|  |  | Архитектура иерархии | 1 | 0 | 0,5 |
|  |  | Собственные объекты, инфотипы, связи | 1 | 0 | 0,2 |
|  |  | Версии плана. Статусы объектов | 0,5 | 0,1 | 0,4 |
|  |  | Отчетность | 0,5 | 0,1 | 0,5 |
| Льготы, предоставляемые работодателем | 3 | Бизнес-процессы | 1 | 0 | 0,5 |
| Инфотипы | 1 | 0 | 0 |
| Отчетность | 1 | 0 | 0 |
| Управление глобальными сотрудниками | 3 | Инфотипы | 0,5 | 0 | 0 |
| Процесс | 1 | 0 | 0,2 |
| Компенсационный пакет | 0,5 | 0 | 0 |
| Расчет заработной платы | 1 | 0 | 0,1 |
| Управления сотрудниками, имеющими несколько контрактов | 5 | Процессы администрирования | 1 | 0 | 0,1 |
| Управление временем | 1 | 0 | 0 |
| Льготы, предостав-ляемые работодателем | 0,5 | 0 | 0 |
| Расчет заработной платы | 2 | 0 | 0,2 |
| Проводки | 0,5 | 0 | 0 |
| Управление бюджетами должностей | 3 | Обязательства | 0,5 | 0 | 0,3 |
| Бюджеты | 0,5 | 0 | 0,3 |
| Интеграция с другими компонентами | 1 | 0 | 0,3 |
| Управление бюджетами | 1 | 0 | 0,5 |
| Управление командировками | 5 | Бизнес-процессы | 1 | 0 | 0,4 |
| Планирование | 1 | 0 | 0 |
| Командировочные расходы | 2 | 0 | 0 |
| Отчетность | 1 | 0 | 0 |
| Пенсионные фонды | 3 | Бизнес-процессы | 1 | 0 | 0,5 |
| Функции | 1 | 0 | 0,3 |
| Интеграция с другими компонентами | 0,5 | 0 | 0,3 |
| Отчетность | 0,5 | 0 | 0 |
| Программирование в НСМ-1 | 19 | Стандартная отчетность/SAP Query/BW | 2 | Использование стандартных отчетов | 1 | 0,7 | 1 |
| BW content для НСМ-1 | 0,5 | 0,2 | 0,5 |
| Расширения для SAP Query | 0,5 | 0 | 0 |
| Workflow в HCM-1 | 5 | Базовый процесс | 2 | 0 | 0,1 |
| Workflow в Администрирование персонала | 1 | 0 | 0,2 |
| Wbrkflow в управлении временными данными | 1 | 0 | 0,1 |
| Wjrkflow в управлении командировками | 1 | 0 | 0 |
| АВАР в НСМ-1 | 10 | АВАР workbench | 4 | 0,5 | 2 |
| User-exits, badis, includes, enhancements | 1 | 0,05 | 0,5 |
| АВАР репозиторий | 2 | 0 | 1 |
| MS Office integration (OLE, DPI), Adobe | 1 | 0 | 0 |
| ALV | 2 | 0 | 1 |
| Drilldown отчетность + HR forms | 2 | Создание drilldown отчетов | 1 | 0 | 0 |
| Создание Hrforms отчетов | 1 | 0 | 0,5 |
| Администрирование в НСМ-1 | 11 | Полномочия | 3 | Настройка ролей, полномочий | 1,5 | 0,8 | 1 |
| Структурные полномочия | 1 | 0,5 | 0,7 |
| Полномочия, зависимые от контента | 0,5 | 0 | 0,2 |
| ALE | 2 | Модель распределения | 1 | 0 | 0,8 |
| Создание, изменение idoc | 1 | 0 | 0 |
| CATS | 2 | Настройка CATS | 1 | 0 | 0,5 |
| Интеграция с использованием CATS | 1 | 0 | 1 |
| LSMW+SXDA | 2 | Batch input, direct input, BAPI | 1 | 0,4 | 0,8 |
| Выгрузка во внешние системы | 1 | 0 | 0,5 |
| Архивация данных | 1 | Процессы архивирования | 1 | 0 | 0,2 |
| Archive Link | 1 | Archive link | 1 | 0 | 0,1 |
| Итого |  |  |  |  |  | 24,9 | 55,7 |

В [табл. 6.10](http://www.intuit.ru/studies/professional_retraining/963/courses/502/lecture/11398?page=4#table.6.10) представлены требования к грейдам, разработанные на основании опыта внедрения проектов по функциональности *SAP* *HCM*-1.

Важно не просто составить реестр такого рода, но и поставить цель - развитие навыков у максимального количества молодых сотрудников; только тогда организация может аккумулировать и поддерживать знания и навыки, которые она продает на рынке.

|  |  |  |
| --- | --- | --- |
| Таблица 6.10. Описание грейдов консультантов | | |
| **Код** | **Описание** | **Коэффициент** |
| К1 | Консультант-стажер | 0-19 |
| К2 | Консультант | 20-34 |
| К3 | Старший консультант | 35-44 |
| К4 | Ведущий консультант | 45-59 |
| К5 | Консультант-эксперт | 60-100 |

При планировании команды проекта кроме определения ролей, ответственности и подотчетности в проекте, создается план управления обеспечением персоналом, который включает в себя*определение* сроков и способов набора членов команды проекта, критерии их высвобождения, рекомендации по проведению дополнительного обучения. В процессе планирования формируется концепция мотивации, определяются способы разрешения конфликтов, разрабатывается *график* проведения собраний команды проекта и его участников.