

Общие вопросы управления проектами

Общие вопросы управления проектами.....	1
Введение.....	1
Что такое проект и управление проектами?.....	2
Ограничения в проектах.....	3
WBS: Work Breakdown Structure - структура декомпозиции работ.....	6
Стандарты в области управления проектами.....	8
Концепция и структура PMI PMBOK.....	10
Проекты информационных систем.....	14
Расширения PMBOK в приложении к ИТ.....	15
Управление инженерной деятельностью в проекте.....	15
Управление приобретением программного обеспечения.....	16

Введение

Прислушаемся к себе. Как часто мы говорим “программный проект”, “проект создания ПО”, “проект разработки”? Наверное, не один раз в день. И это вполне естественно. Интуитивно мы понимаем, что “проект” есть некая совокупность упорядоченных действий, направленных на получение конкретного результата. Мы “занимаемся проектом”, когда работаем над программным продуктом – будь это новое, пусть и небольшое, приложение с ограниченным кругом пользователей или адаптируем и интегрируем корпоративную информационную систему класса ERP.

Являясь ИТ-специалистами, мы постоянно взаимодействуем с окружающей бизнес-средой, профессиональными менеджерами, специалистами в области финансов и т.п. Обсуждая с ними “проект”, степень его готовности, ассоциированные с ним ресурсы и другие характеристики, мы обычно не задаемся вопросами согласования терминологии (думая, что “уж мы то, наверняка, лучше знаем что такое “программный проект”...), не связанной с сутью, содержанием проекта. Профессиональный сленг, термины и понятия, которые мы принимаем и используем в повседневной практике, часто изначально сформулированы на английском языке. У профессиональных менеджеров, финансистов, руководителей отдела кадров – свое понятийное и терминологическое пространство. И мы не учитываем того, как велика вероятность недопонимания при использовании профессиональной терминологии. Как среди специалистов одной профессии, так и, тем более, с представителями других областей деятельности.

Иногда сотрудники внутрикорпоративных ИТ-департаментов, отвечающие за автоматизацию линейных подразделений (Line of Business - LOB), считают, что знают об области деятельности конечных пользователей существенно больше, чем сами пользователи. В многих случаях, небезосновательно. В большинстве случаев, это совсем не забавное заблуждение. В то же самое время, все чаще руководством ИТ-проектов и самих ИТ-подразделений занимаются профессиональные управленцы, на которых “айтишники” смотрят с высоты своего технократического снобизма. Зря. Что стоит за терминами “риски”, “бюджетирование”, “закрытие проекта”, “планирование работ”? ИТ-специалисты часто не только не понимают, но и не хотят понимать этого. А ведь это вопросы, решение которых непосредственно влияет на работы в рамках ИТ-проектов.

Забавный, но более чем поучительный эпизод приводят Александр Товб и Григорий Ципес, обсуждая ключевые понятия и термины управления проектами [Товб А., Ципес Г., 2003, с.35]:

“Однажды в нашей компании проходили практику студенты-дипломники, специализирующиеся по управлению проектами. Выдавая им задание, руководитель практики попросил описать *scope* проекта (он так и сказал – “скоуп”). “А что такое *scope*?” - осторожно уточнила одна девушка. “О, *scope* – это ...” – ответил руководитель и нарисовал руками в воздухе нечто, напоминающее средних размеров глобус. “Понятно, - грустно сказала девушка. – Нам в институте так же объясняли”

Это не только вопрос использования профессионального жаргона или “чистоты языка”. Хорошо известна опасность различной трактовки одних и тех же понятий специалистами одной области

деятельности. Непонимание же друг друга представителями заказчика и исполнителя (а именно исполнителями являются ИТ-специалисты по отношению к бизнесу) – это почти стопроцентная гарантия провала проекта. Идя навстречу друг другу, заказчики/пользователи и исполнители/разработчики, должны оперировать одними и теми же концепциями и понятиями. В противном случае ... – ну, дальше вы знаете. Более того, готовность говорить с бизнесом на его языке, это не только еще один шаг на встречу заказчику, но и возможность взгляда на разработку программного обеспечения как на еще один бизнес-процесс, пусть и обладающий некоторыми специфическими характеристиками.

Одно из ведущих мировых аналитических агентств - Standish Group с 1994 года ведет исследования успешности ИТ-проектов, результаты которого выпускает в виде известного The Chaos Report. В 2004 году [Chaos, 2004] 18% проектов провалились, 53% (более половины!) – не уложились в заданный бюджет или сроки. Только 29% были признаны успешными. И это по более чем 9 тысячам проектов (кумулятивное число проектов, по которым накоплены данные – 50.000). Интересно, что результаты исследований ничего не могут сказать о влиянии на успешность проектов факта применения формальных методологий или отсутствие такового. Это может говорить (пусть и косвенно) о том, что само по себе следование таким методологиям не является залогом успеха. В то же самое время, четко просматривается влияние уровня компетенции менеджеров проектов на результат. Успешное завершение и, тем более, спасение тонущего проекта тем вероятнее, чем большим кругозором обладает менеджер проекта, пришел ли он в ИТ как “чистый управленец” или вырос из среды ИТ-специалистов.

Чем сложнее бизнес-задача, чем важнее ИТ-проект, направленный на ее решение, тем чаще мы оперируем общими управленческими понятиями и применяем знания и навыки, характерные для общего менеджмента (“general management”) и, конечно же, проектного менеджмента, т.е. “управления проектами”.

Что такое проект и управление проектами?

Рассел Арчибальд, один из признанных классиков управления проектами, формулирует, что “Проект – это комплекс усилий, предпринимаемых с целью получения конкретных уникальных результатов в рамках отведенного времени и в пределах утвержденного бюджета, который выделяется на оплату ресурсов, используемых или потребляемых в ходе проекта.” [Арчибальд Р., 2003, с.34].

Разные источники по разному формулируют понятие *проекта*. Интересную подборку определений можно найти в [Товб А., Ципес Г., 2003, с.24]. Все эти определения сходятся в одном – *проект, есть комплекс действий, направленных на получение уникального результата, будь то продукт или услуга. Суть результата – его содержание*. Для информационной системы – ее функциональность.

“Проект представляет собой комплекс уникальных действий, не опирающийся на организационную структуру, имеющий определенные дату начала и окончания, расписание, стоимость и технические задачи. Управление проектом, следовательно, сильно отличается от управления обычным функциональным подразделением, в котором постоянно выполняется одна и та же работа, не имеющая четкой даты и завершения.” [Арчибальд Р., 2003, с.64].

Необходимо четко разграничивать *проектные работы* (например, направленные на создание нового программного продукта) и *операционную деятельность* (например, функционирование службы технической поддержки) – см. [PMI PMBOK3, 2004, Рус, “1.2.2 Чем проекты отличаются от операционной деятельности”, с.6-7]. В этом плане, обязательными признаками или, если хотите, характеристиками проекта является *временный характер проекта и определенный результат*.

Однако, говоря о *результате проекта*, не стоит путать *цель проекта* и *содержание проекта*. Например, цель проекта создания автоматизированной системы учета договоров страхования – автоматизация бизнес-процессов, направленных на операции учета договоров страхования. В то же время, содержание данного проекта может звучать так: создание программного приложения или комплекса приложений, обеспечивающих хранение, учет и управление информацией по договорам в электронной форме, т.е. функциональность создаваемой системы. Содержание

проекта как раз и есть его “*scope*”. Таким образом, *цель проекта описывает какие задачи должны быть решены в результате проекта, а содержание проекта - что именно является результатом проекта. Управление проектом определяет “как”, с помощью каких действий, будет достигнута цель проекта и создан необходимый результат.* При этом, управление проектами может и должно применяться на всех этапах жизненного цикла проекта, т.е. *управление проектом есть постоянная деятельность, начиная с его инициации, вплоть до завершения проекта, то есть получения результата.* В отличие от результата проекта, управление проектами не является уникальным с точки зрения процессов. Кроме того, необходимо понимать, что получение результата проекта и достижение целей проекта – не одно и то же. Проект можно считать успешным при условии, что результат проекта соответствует заданному содержанию проекта и его целям. В какой степени цели проекта достигнуты зависит от адекватности определенного содержания проекта его целям. Корректно определить содержание в контексте целей, провести работы в рамках заданных ограничений, довести работы до конца, то есть до получения результата – это и есть задачи, стоящие перед менеджерами проектов.

В силу масштабности содержания проекта либо, например, разнородности его составных частей (например, программно-аппаратный комплекс шифрования данных) проект может быть разбит на несколько более мелких проектов.

Проектная программа или просто *программа* – *совокупность связанных проектов.* При этом, обычно подразумевается, что программа может быть успешно завершена только при условии успешного завершения всех проектов, ее составляющих. Например, программа создания нового космического корабля невозможна, если всего лишь один из ее проектов, направленный на разработку обновленной системы жизнеобеспечения, окажется неуспешен. В этом плане, можно говорить об определенной “*атомарности*” программ, по аналогии с понятием транзакции в СУБД. Отличие от такой аналогии, к сожалению, состоит в том, что целиком “откатить” <проектную> программу - невозможно, так как уже затрачены определенные время и деньги.

В последние годы стала активно применяться практика трансформации определенных видов операционной деятельности в программы, состоящие из однотипных проектов, выполняемых с определенной периодичностью. Такой подход получил название “*управление через проекты*”.

В то же время, не стоит путать *программы* с *портфелями проектов*, которые, в свою очередь, являются комплексами проектов с единым или пересекающимся пулом ресурсов и единым центром управления/координации. Проекты также могут объединяться в портфели на основе общих стратегических целей. Типичным примером портфеля проектов может выступать портфель инвестиционных проектов, проводимых конкретным финансовым институтом или компанией. Общая цель таких проектов – получение прибыли в интересах пайщиков, например, негосударственного пенсионного фонда. Общность ресурсов проектов, в данном случае – финансовые средства, направляемые в проекты.

Так как с любым проектом ассоциированы цели, ресурсы, время, мы можем сформулировать, что управление проектами – есть дисциплина применения методов, практик и опыта к проектной деятельности для достижения целей проектов, при условии удовлетворения ограничений, определяющих их рамки.

Ограничения в проектах

Что же это за *ограничения* (англ. *constraints*), в рамках которых мы принимаем те или иные решения, влияющие на первичное планирование и дальнейший ход работ по выполнению проектов?

Чаще всего говорят о трех основных ограничениях или “железном треугольнике” [PMI PMBOK3, 2004, Рус, с.8]:

1. Содержанию проекта
2. Времени
3. Стоимости

В приложении к индустрии программного обеспечения обычно добавляют четвертое ограничение – *качество (quality)*. Если быть более точным – *приемлемое качество*. “Что значит “приемлемое”?” - с негодованием может спросить кто-то из вас. Это тот уровень качества, который позволяет считать результат достигнутым. Если в течение шести месяцев эксплуатации клиентских приложений системы учета договоров на трех десятках рабочих станций система “падала” каждый месяц хоть на какой-то машине – это неприятно. Если она упала за 6 месяцев один раз, но “накрылись” данные за последний месяц – это беда. Субъективно, конечно. Поэтому и в индустрии программного обеспечения и в управлении проектами и в любой другой области деятельности все больше внимания уделяется уточнению вопроса “что такое качество” и разрабатываются подходы, методы и метрики для измерения качества (это и Six Sigma и TQM – Total Quality Management и многие другие). В зависимости от контекста и обсуждаемых в конкретном случае критериев качества, “*приемлемое*” качество может рассматриваться как *необходимое*, например, заданное требованиями качества и внутрикорпоративными стандартами. С точки зрения нахождения баланса между всей совокупностью требований и бюджетом (или затратами), ассоциированными с проектом, приемлемое качество может считаться *достаточным* или *обоснованным (достижимым)*. Любая оценка качества должна базироваться на *измерениях и количественно выражаемых результатах измерений*. Требования к качеству также должны описываться *исчисляемыми характеристиками*. Только при таком подходе вообще можно говорить о качестве как о конкретной характеристике процесса создания продукта и самого продукта как результата проекта. Далее, используя термин “приемлемое качество” мы будем подразумевать его интерпретацию как “необходимого” или “достаточного” *в зависимости от контекста*, в то же время, всегда оставаясь в рамках *количественного подхода*.

Является ли качество *ограничением* при таком взгляде? Наверное, да – в том случае, когда мы говорим о необходимом качестве, то есть тех количественных характеристиках качества, которые диктуются (явно или неявно) соответствующими требованиями (как к продукту, так и к процессу). Если же мы обсуждаем достаточное качество, скорее такой уровень качества определяется другими ограничениями, например, сроками и стоимостью. Кроме того, достаточное качество в качестве ограничения может принимать и требования к качеству. Наверняка, многие из читателей сразу вспоминают аббревиатуру “НиДУ” – Необходимые и Достаточные Условия. В какой-то степени, обсуждение достижения определенного уровня качества (всегда характеризующегося количественными характеристиками) относится к *классической дилемме управления проектами - поиску баланса* в рамках системы ограничений, связанных с проектом.

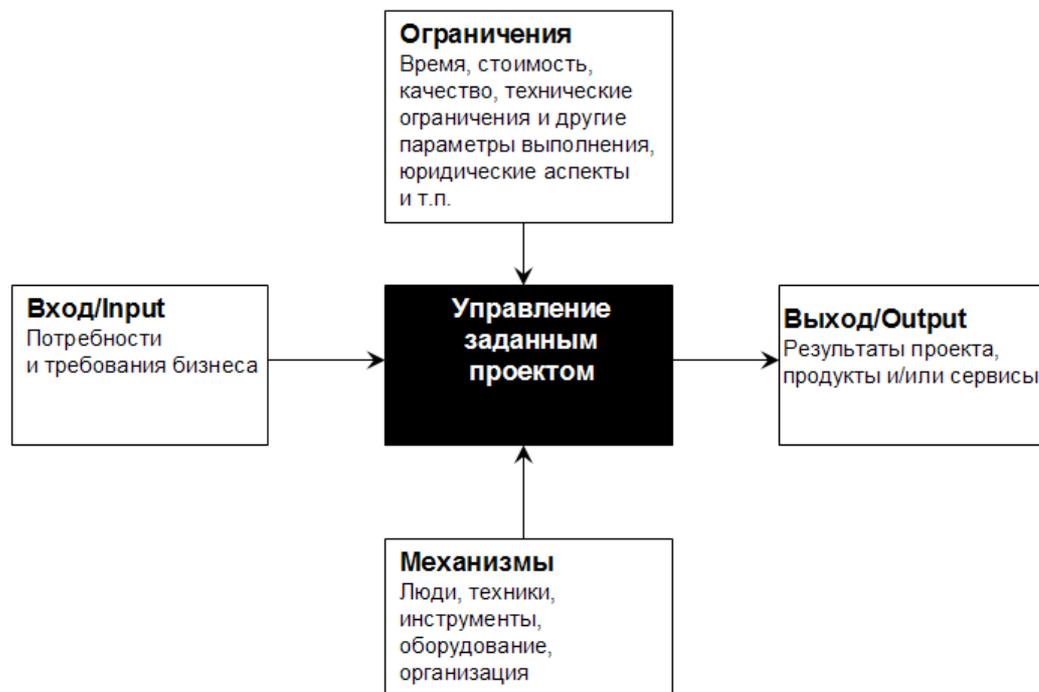


Рисунок 1. Процесс управления проектом. Роль ограничений. Источник: [APM PMBoK, 2000, с.15]

Система ограничений может строиться на основе приоритетов проекта и должна учитывать требования потребителей к создаваемому продукту или услуге. Если необходим жесткий предопределенный набор функциональности - понятно, что “плавающими” характеристиками проекта (вторичными по своей природе, требующими компромисса в контексте требуемого объема функциональности) будут требуемое время, квалификация и опыт специалистов, необходимый бюджет.

В индустрии информационных технологий идет дискуссия по поводу того, какие проекты могут быть реализованы на основе подхода “фиксированная цена” и когда такой подход обоснован. В то же время, ограничения, связанные с имеющимися ресурсами, которые можно направить на решение проектных задач, также может рассматриваться как фиксированное ограничение. Качество? Тоже может относиться к такому ограничению. Сроки – скорее всего. Как же тогда работать, если мы попадаем в “железный треугольник”. Что это за треугольник? Например, создатели одного из популярных учебников по управлению проектами – Клиффорд Грей и Эрик Ларсон [Грей и Ларсон, 2003, с. 81] отмечают: “Одной из основных задач управляющего проектом является управление соотношением между временем, стоимостью и результативностью” (понимая под результативностью, в первую очередь, достижение заданного содержания).

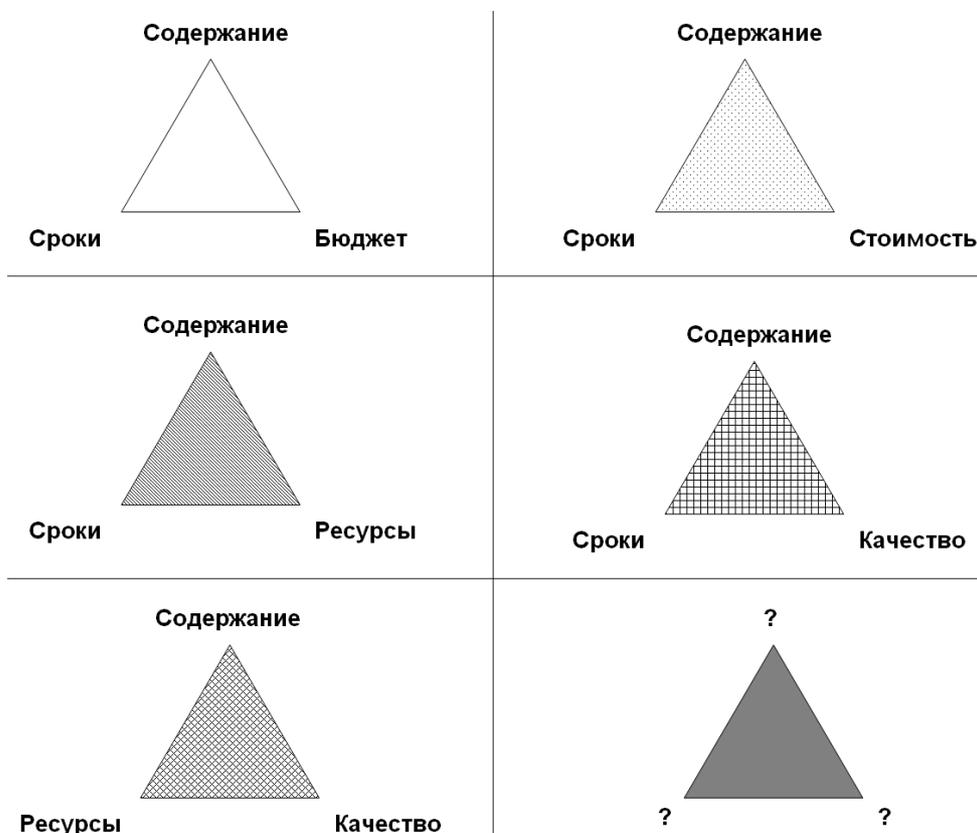


Рисунок 2. Возможные варианты “железного” треугольника ограничений проекта. Определите *ваши* ограничения на основе анализа приоритетов и поиска компромиссов.

Между тем, *система ограничений определяется совокупностью приоритетов*. Не забывайте, что ограничения всегда специфичны для конкретного проекта и определяются его приоритетами. Для определения приоритетов применяется ряд техник управления проектами. В то же время, любые заданные или определенные (например, в процессе анализа проектных требований и их обсуждения с “заказчиком”) ограничения порождают риски. Управление проектами как систематическая деятельность предполагает *систематическое управление рисками*.

В общем случае существует два распространенных подхода в отношении рисков – *реактивный*, когда мы реагируем на проблемы, ставшие результатом трансформации риска, как ожидания “что-

то случится”, в сам факт того, что “это уже случилось” – типичным представителем такого класса менеджеров является небезызвестный Индиана Джонс. Решение проблем по мере их поступления в индустрии информационных технологий можно встретить особенно часто. Это прекрасно видно по результатам анализа проектов [Chaos, 2004]. К сожалению, в реальной жизни, везение не столь благосклонно к реальным проектам, в отличие от Джонса Младшего. Второй подход – *упреждающий* или *проактивный*, предполагающий идентификацию возможных рисков и разработку плана действий на те случаи, когда невозможно предотвратить превращение риска в проблему. Менеджеров, уделяющих внимание анализу и предотвращению возможных проблем, то есть управлению рисками – существенно меньше, один из них – Шерлок Холмс. Практика управления проектами, в том числе в области информационных технологий, столь значима, что управление рисками, наравне с определением ограничений, рассматривается практически в любом своде знаний, методологии или практике. В большинстве случаев, *основная причина рисков – сам факт существования ограничений*, иначе в чем будет состоять риск, как не в нарушении этих ограничений?

Например, если бюджет не является жестко определенным и может быть предметом обсуждения, другие ограничения все равно будут играть свою роль. Поэтому считается, что неотъемлемой частью любого программного проекта (в частности, в силу высокой неопределенности, связанной с самой областью программной инженерии) является *анализ компромиссов - trade-off анализ*, направленный на выделение тех функциональных требований, который возможно реализовать в заданных рамках проекта, будь то сроки, ресурсы, стоимость или другие характеристики. В общем случае, *задача такого анализа – нахождение баланса, приемлемого для всех сторон, связанных с проектом* – и заказчиков (которым нужна такая-то функциональность в такие-то сроки, при таком-то бюджете) и исполнителей (которые обладают такими-то ресурсами, при определенной стоимости использования ресурсов и трудоемкости достижения качества в заданные сроки).

Для того, чтобы провести такой анализ, необходимо обладать информацией о деталях содержания проекта. Многие ИТ-специалисты чувствуют себя в этой области дисциплины управления проектами достаточно комфортно. Почему? Потому как *структурный подход*, приводящий к *декомпозиции задач*, а, следовательно, и *проектных работ* очень часто используется при разработке программного обеспечения. Значит ли это, что можно опустить данную тему? Нет. Сбор и анализ требований, является обязательной составной частью проекта, так как требования определяют цель и содержание проекта. Может ли разработка кода программной системы начинаться если требования еще не определены? Это вопрос модели разработки. Практика индустрии демонстрирует, что это возможно и даже желательно. Мы будем позднее более подробно рассматривать вопросы управления требованиями. Сейчас же нам важно отметить следующие аспекты управления проектами:

- ограничения являются следствием приоритетов;
- приоритеты “заказчика” и “исполнителя” могут противоречить друг другу
- анализ компромиссов позволяет определить баланс приоритетов, приемлемый для всех сторон, вовлеченных либо заинтересованных в проекте;
- ограничения являются неотъемлемой частью проекта;
- ограничения порождают риски;
- ограничения рассматриваются в контексте уровня детализации проекта

Ну а детализация проекта, в свою очередь, есть одна из важней составных частей дисциплины управления проектами. Иначе, как оценить, например, общую стоимость проекта, как не через совокупность стоимости отдельных его работ?

WBS: Work Breakdown Structure - структура декомпозиции работ

Структура декомпозиции работ (Work Breakdown Structure, WBS) есть результат детализации содержания проекта. Несколько менее часто используется термин структура декомпозиции проекта (Project Breakdown Structure). В большинстве случаев, такая структура является *иерархической*.

При этом сам процесс детализации - *структурная декомпозиция работ*, то есть деятельность по созданию детализированной структуры работ или задач проекта.

В контексте детализации мы часто используем термины “задачи” и “работы” как взаимозаменяемые. Все же корректнее говорить, что *задача определяет достижение промежуточного результата, а работы являются комплексами действий для достижения этих результатов*. Так как любая задача, требует проведения определенных работ при заданных ограничениях, безусловно, можно говорить об определенном “отображении” (mapping) задач на работы и наоборот. Это и есть причина взаимозаменяемости терминов в повседневной жизни. “Иерархическая структура проекта разрабатывается путем разумного комбинирования иерархической структуры продукта с процессом его разработки.” [Арчибальд Р., 2003, с.301]. В то же время понимание отличий между этими понятиями позволяет почувствовать нюансы процессного взгляда на создание продукта, а следовательно, и на жизненный цикл проектов, в том числе и проектов программных систем.

В общем случае, говорят о детализации: *программа-проект-задача-операция*. На рисунке 3. представлен пример такой детализации (показаны только операции Задачи А1).

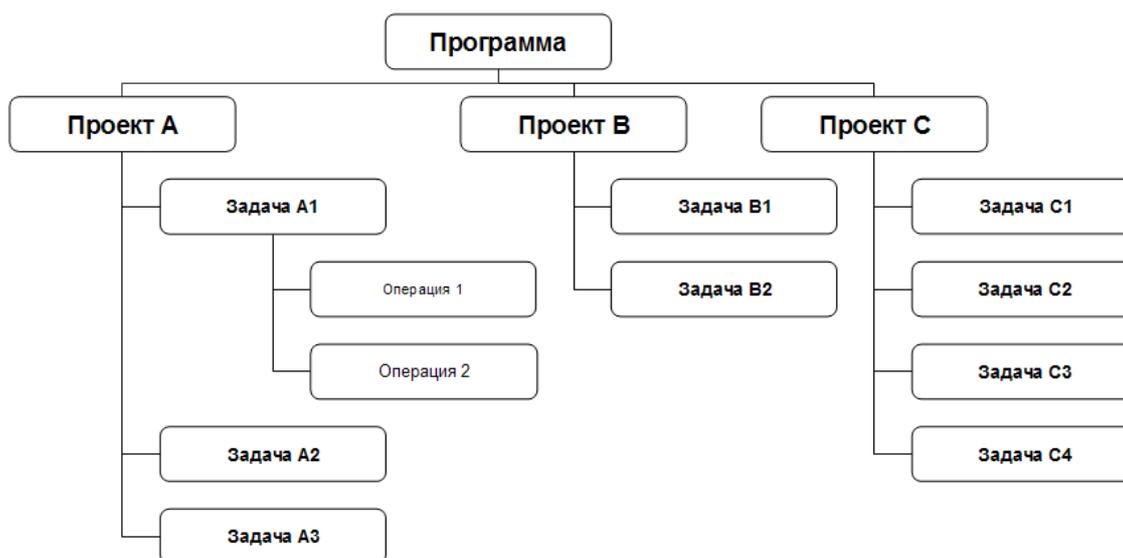


Рисунок 3. Пример контекста применения терминов “программа”, “проект”, “задача”, “операция”. Показаны только операции Задачи А1.

В дальнейшем, (например, в SWEBOOK) мы будем иногда использовать более подробную детализацию: *программа-проект-действие/работа-задача-операция*, где “действия/работы” - *activities*, “задачи” – *tasks*: термин “задача” используется для более глубокого уровня детализации, чем “действия/работы”; термины “действия” и “работы”, как вы уже заметили, используются взаимозаменяемым образом.

Декомпозиция, создание структуры работ – есть ключевой элемент борьбы со *сложностью*, являющейся одной из ключевых особенностей программных систем [Fred Brooks, 1987]. Сложные проекты не делаются в одиночку. Значит, необходимо и *распределение работ* между членами коллектива. А это означает и *ответственность* (responsibility) за реализацию задач, составляющих проект. Управление проектами не может вестись вне контекста организационной структуры – как организации, так и проектной команды. За этим стоят и обязанности и роли, которые выполняют люди, входящие в проектную команду. Это не только вопросы дисциплины управления проектами, но и вопросы общего менеджмента. Не бывает одинаковых организаций или проектов. Также не бывает универсальной организационной структуры. В любой книге (по крайней мере я не знаю исключений из этого правила) по управлению проектами вы найдете обсуждение вопросов организационной структуры и обязанностей, ролей в проектной команде. В данный момент мы не будем касаться этого вопроса более детально, так как ролевая структура проектной команды в большой степени зависит от используемой методологии и взгляда на жизненный цикл проекта. Так что мы будем уделять внимание этим аспектам уже в контексте конкретных подходов и методов, которым посвящена большая часть этой книги.

Таким образом, с каждым элементом такой структуры связаны ограничения и другие значимые характеристики и данные. Под *значимостью* мы подразумеваем *необходимость или полезность для принятия решений*. Глубина же детализации и уровень применения тех или иных терминов зависит от конкретного проекта, культуры управления, стандартов жизненного цикла, специфики и масштабов проекта, а также других факторов, уникальных как в контексте конкретной организации, так и конкретного проекта.

Слишком субъективно? Безусловно. Однако теоретический и практический опыт, накопленный в различных областях человеческой деятельности приводит ко все более точному описанию вопросов управления проектами, в том числе терминологических. А общий словарь различных участников проекта и лиц, заинтересованных в нем (т.е. в его результате) – это уже определенное достижение. Значит, необходимо формировать и оперировать определенными стандартами в этой области деятельности.

Сама концепция декомпозиции столь значима для повседневной практики управления проектами, что американский Институт Управления Проектами (Project Management Institute, PMI) выпустил специальные рекомендации – Practice Standard for Work Breakdown Structures [PMI WBS, 2001], описывающие что такое WBS, почему стоит использовать WBS и как создавать WBS, а также примеры WBS для разных категорий проектов.

Стандарты в области управления проектами

В предисловии к третьему изданию книги “Управление высоко-технологичными программами и проектами” [Арчибалд Р., 2003, с.25] Арчибалд пишет “За десять лет ... фундаментальные концепции управления проектами не претерпели существенных изменений. Однако за это десятилетие заметно эволюционировали практические подходы к управлению проектами.” В частности, он отмечает “повышение зрелости функции управления проектами и признание растущей важности во многих организациях...” и, что не менее важно “развитие самой специальности управления проектами, сопровождаемое разработкой ... нескольких сводов (“древ”) знаний по управлению проектами, ряда моделей зрелости управления проектами и различных подходов к сертификации практикующих специалистов в данной области”.

На сегодняшний день существует множество стандартов (в том числе национальных) и моделей, направленных на систематизацию знаний в области управления проектами. Наряду с ними представлены и системы сертификации специалистов в области управления проектами, задачей которых является подтверждение профессионального статуса управляющих проектами (менеджеров проектов) как профессионалов в дисциплине управления проектами.

Наиболее известны и широко распространены результаты деятельности Project Management Institute, Inc. (PMI) – американского Института Управления Проектами, Международной Ассоциации Управления Проектами - International Project Management Association (IPMA) и Ассоциации по Управлению Проектами СОВНЕТ (источник Национальных требований к компетентности специалистов по Управлению Проектами, НТК [СОВНЕТ НТК, 2000]), английской Ассоциации Управления Проектами – Association of Project Management (APM), национальные стандарты и методологии Великобритании, например, PRINCE® (PProjects IN Controlled Environments), международные стандарты ISO.

Например, в приложении к ИТ стоит обратить внимание на рекомендации ISO/IEC TR 16326:1999 Software Engineering – Guide for the application of ISO/IEC 12207 to project management; ISO/IEC 12207 (в американском варианте – семейство IEEE/EIA 12207.x) и его российский вариант ГОСТ Р ИСО/МЭК 12207-99 (будет рассмотрен далее в этой книге) и многие другие. Дополнительную информацию по источникам стандартов и сертификации в области управления проектами можно, например, найти в работах и презентациях Александра Товба и Григория Ципеса, в частности, в [Товб А., Ципес Г., 2003] или на странице сайта PMForum – “Project Management Standards and Professional Certification” (<http://www.pmforum.org/prof/standard.htm#STANDARDS>) .

Пытаясь описать и регламентировать ту или иную область деятельности, различные ассоциации и организации, в том числе государственные, формулируют стандарты, как де-факто, так и де-юре.

Желание максимально охватить конкретную область деятельности привело к формированию тенденции создания *сводов знаний* в данной области – *Body of Knowledge (BoK)*.

Данная тенденция наблюдается не только в отношении дисциплины управления проектами, хотя именно здесь она проявилась впервые, когда британская Ассоциация Управления Проектами APM (Association of Project Management) в 1992 году выпустила первую редакцию Project Management Body of Knowledge [APM PMBoK, 2000]. Области знаний управления проектами также посвящен PMI PMBOK, выпущенный в 2004 году уже в третьей редакции [PMI PMBOK3, 2004, Рус]. При том, что существуют и другие работы в этой области, наиболее широко распространены именно указанные своды знаний APM и PMI.

Где еще мы можем наблюдать формирование сводов знаний? Как раз в информационных технологиях, а точнее программной инженерии, например:

- IEEE* Guide to Software Engineering Body Of Knowledge (<http://www.swebok.org>) – IEEE SWEBOK (будет рассмотрен позднее в этой книге), созданный при поддержке ACM (Association for Computing Machinery) [SWEBOK, 2004];
- SEEK: Software Engineering Education Knowledge, входящий в IEEE Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering [SE, 2004].

* IEEE - Computer Society of the Institute for Electrical and Electronic Engineers, IEEE Computer Society или просто IEEE.

Так как любая область человеческой деятельности эволюционирует, объем накопленных знаний увеличивается, такие своды также развиваются с течением времени. Желание “объять необъятное” всегда приводит к краху. Попытка выделить то, что кажется значимым – естественна, а результат достижим. Поэтому, чаще говорят не о сводах знаний, как всецелом описании области деятельности, а о *руководствах к сводам знаний* – *guide to the body of knowledge*, которые акцентируют внимание на аспектах существенных с теоретической и практической точки зрения. Это необходимо понимать. А, следовательно, и воспринимать такие своды, в том числе имеющие статус стандартов, как концентрат опыта индустрии, но никак не в качестве идола для поклонения, требующего неукоснительного следования и неприемлющего расширений и трактовок. Все, в том числе приложение стандартов, необходимо рассматривать в конкретном *контексте* на основе вашего *опыта и профессионализма*. Если вы чувствуете, что опыта не хватает – не бойтесь обращаться к более опытным коллегам и профессиональным консультантам в той или иной области. Если же вы обратились к ним, ни в коем случае не забывайте, что и они – люди, и любое беспрекословное и бездумное следование их рекомендациям – гарантия вашего собственного поражения.

Как соотносится свод знаний по управлению проектами с другими смежными областями деятельности и экспертизы, например, общим менеджментом? Точка зрения PMI на этот счет представлена на рисунке 3.



Рисунок 4. Экспертные области, необходимые для команды управления проектом. Источник: [PMI PMBOK3, 2004, Рус, с.13]. Используется с разрешения PMI.

Мы рассмотрим один из этих стандартов – PMI PMBOK, в его третьей редакции [PMI PMBOK3, 2004, Рус]. Почему я выбрал именно его? Этот стандарт доступен на русском языке и, вступая в члены PMI (информацию об этом можно найти на сайте московского отделения PMI - PMI Moscow Chapter <http://www.pmi.ru>), вы автоматически получаете доступ как к оригинальному варианту на английском языке, так и к его актуальному переводу на русский язык. Этот стандарт является национальным стандартом США (ANSI/PMI 99-001-2004 для редакции PMBOK 3, выпущенной в 2004 году). Кроме того, ряд ключевых положений PMI PMBOK получил статус де-юре в рамках впитавшего их международного стандарта ISO10006:1997. На момент публикации третьей редакции PMI PMBOK было продано более одного миллиона экземпляров этого Руководства. И, наконец, один из фундаментальных стандартов Software Engineering, начинающих проникать в повседневную жизнь – IEEE Guide to the Software Engineering Body of Knowledge [SWEBOOK, 2004] также ссылается на PMI PMBOK. Поэтому, далее я буду использовать в книге аббревиатуру “PMBOK”, подразумевая именно [PMI PMBOK3, 2004, Рус].

Концепция и структура PMI PMBOK

PMBOK, как уже отмечалось ранее, является стандартом. Объем стандарта PMBOK (наверное, как и любого другого стандарта) немалый – 402 страницы. В то же время, важно понимать, что “это не означает, что приведенные знания, навыки и процессы должны одинаковым образом применяться во всех проектах. Менеджер проекта совместно с командой проекта в каждом конкретном случае всегда отвечает за выбор подходящих процессов, а также необходимой точности выполнения каждого процесса.” [PMI PMBOK3, 2004, Рус, с.37]. Это положение в той же мере применимо к любому стандарту или практикам, в том числе, касающимся разработки программного обеспечения.

Знания по управлению проектами представленные в PMBOK в форме 9 областей знаний (*knowledge areas*), детализированных в виде 44 процессов, сгруппированных по областям знаний, и относящимся к пяти группам, называемым “группы процессов управления проектом” или, как их часто называют – “процессным группам”. Эти процессные группы, по-сути, привязаны к фазам проекта:

- Инициация
- Планирование
- Исполнение
- Мониторинг и управление
- Завершение

При этом процессы взаимосвязаны, а *фазы* или как их еще принято называть *этапы проекта* могут и в подавляющем большинстве случаев должны перекрываться - накладываться во времени друг на друга. Это необходимо для оптимизации работ и ресурсов, в особенности, задействованности членов проектной команды, то есть человеческих ресурсов проекта.

Для процессов, определенных в PMBOK, определяются *инструменты и методы* (практики), *входы* (*inputs*) и *выходы* (*outputs*). Например, входами для мониторинга и контроля проектных работ могут рассматриваться план проекта и фактические данные по степени завершенности работ.

Структура PMBOK приведена на рисунке 5. Обратите внимание, что она не включает детализацию самих областей знаний. Структура областей знаний и их процессов представлена на рисунке 6 и в таблице 1.

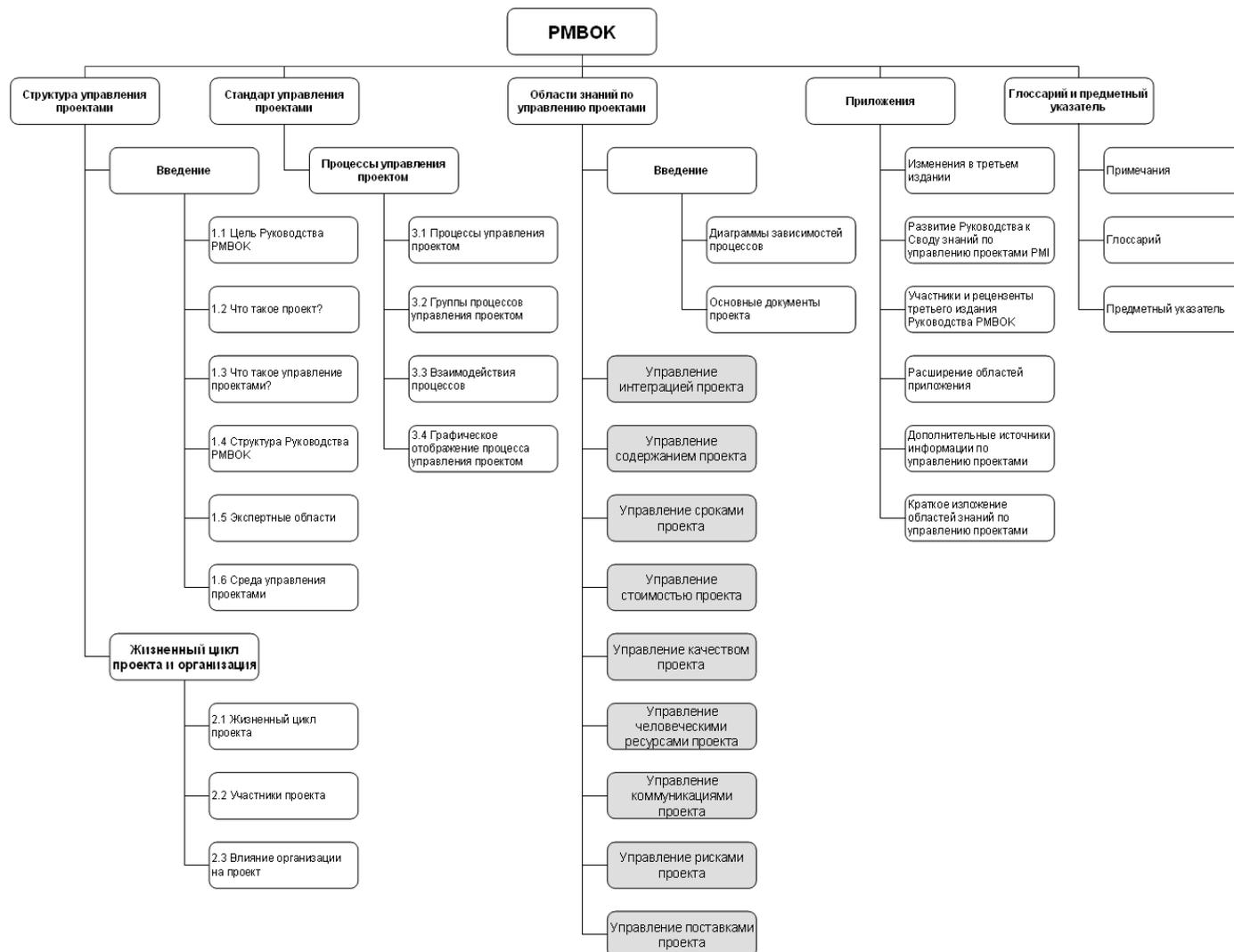


Рисунок 5. Структура PMBOK. Области знаний не детализированы. Построено на основе оглавления [PMI PMBOK3, 2004, Рус].
 Используется с разрешения PMI.

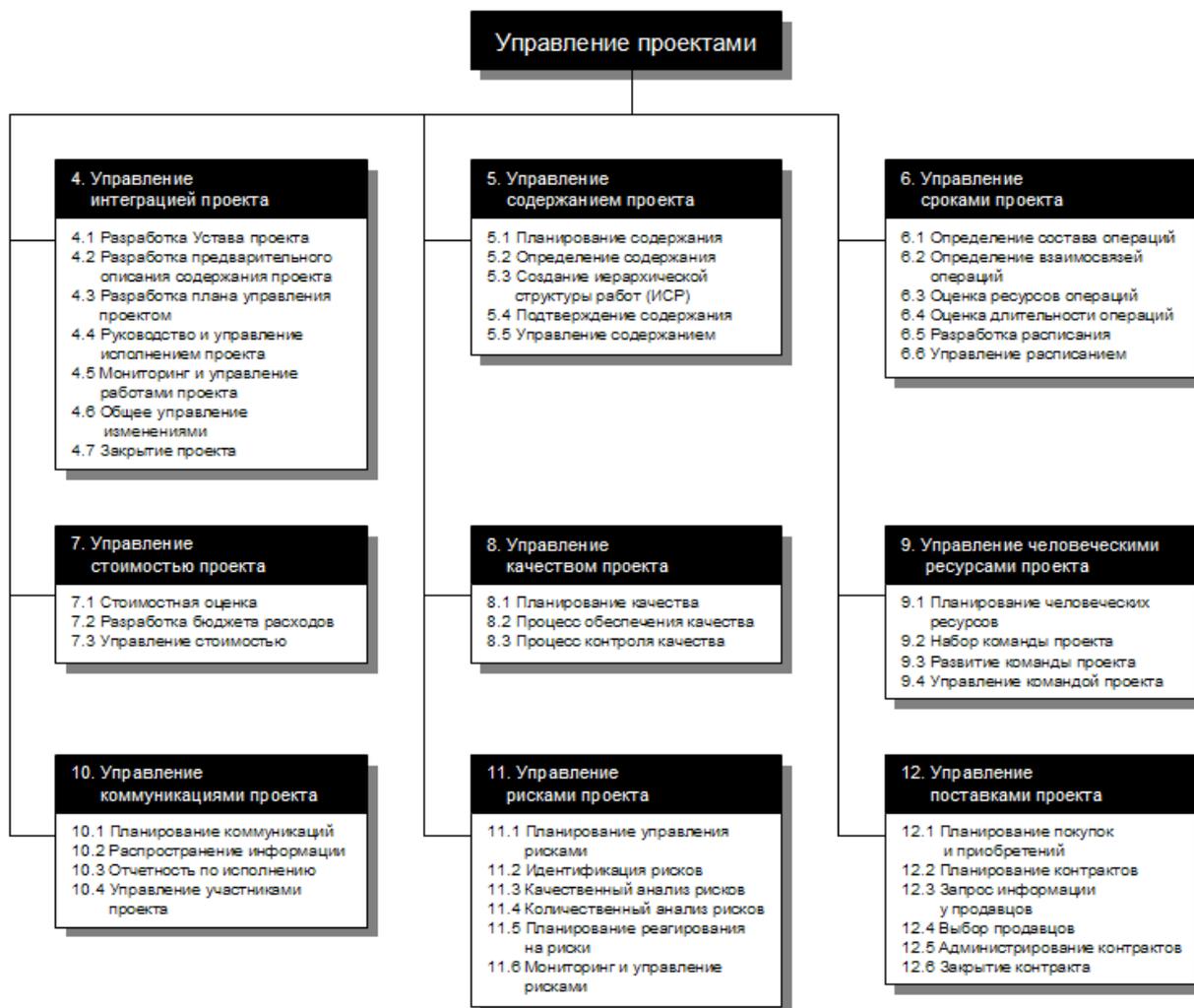


Рисунок 6. Обзор областей знаний по управлению проектами и процессов управления проектами РМВОК. Источник: [PMI РМВОК3, 2004, Рус, с.11]. Используется с разрешения PMI.

Введение в программную инженерию и управление жизненным циклом ПО

Общие вопросы управления проектами

Процессы и области знаний	Группы процессов управления проектами по фазам проекта				
	Инициация	Планирование	Исполнение	Мониторинг и управление	Завершение
4. Интеграция управления проектом	4.1 Разработка Устава проекта 4.2 Разработка предварительного описания содержания проекта	4.3 Разработка плана управления проектом	4.4 Руководство и управление исполнением проекта	4.5 Мониторинг и контроль (управление) работ проекта 4.6 Общее управление изменениями	4.7 Закрытие проекта
5. Управление содержанием проекта		5.1 Планирование содержания проекта 5.2 Определение содержания 5.3 Создание иерархической структуры работ (ИСР)		5.4 Подтверждение содержания 5.5 Управление содержанием	
6. Управление сроками проекта		6.1 Определение состава операций 6.2 Определение взаимосвязей операций 6.3 Оценка ресурсов операций 6.4 Оценка длительности операций 6.5 Разработка расписания		6.6 Управление расписанием	
7. Управление стоимостью проекта		7.1 Стоимостная оценка 7.2 Разработка бюджета расходов		7.3 Управление стоимостью	
8. Управление качеством проекта		8.1 Планирование качества	8.2 Процесс обеспечения качества	8.3 Процесс контроля качества	
9. Управление человеческими ресурсами проекта		9.1 Планирование человеческих ресурсов	9.2 Набор команды проекта 9.3 Развитие команды проекта	9.4 Управление командой проекта	
10. Управление коммуникациями проекта		10.1 Планирование коммуникаций	10.2 Распространение информации 10.3 Ответность по исполнению	10.4 Управление участниками проекта	
11. Управление рисками проекта		11.1 Планирование управления рисками 11.2 Идентификация рисков 11.3 Качественный анализ рисков 11.4 Количественный анализ рисков 11.5 Планирование реагирования на риски		11.6 Мониторинг и управление рисками	
12. Управление поставками проекта		12.1 Планирование покупок и приобретений 12.2 Планирование контрактов	12.3 Запрос информации у продавцов 12.4 Выбор продавцов	12.5 Администрирование контрактов	12.6 Закрытие контракта

Таблица 1. Соответствие между процессами управления проектами, группами процессов по фазам проекта и областями знаний. Источник: [PMI PMBOK3, 2004, Рус, с.70]. Используется с разрешения PMI.

Проекты информационных систем

Арчибальд отмечает важность классификации проектов [Арчибальд Р., 2003, с.66-73]: “Процесс управления проектом может значительно варьироваться в зависимости от категории проектов, при этом наиболее подходящий процесс должен применяться к каждому отдельному проекту”. Это утверждение, конечно, применимо и к проектам программного обеспечения.

Однако, в ряду рекомендованных категорий проектов, Арчибальд не выделяет отдельных типов категории “проектов информационных систем (программного обеспечения)”, он только дает пример такой категории – “новая информационная система управления проектами”, отмечая, что “аппаратное обеспечение такой системы относится к категории разработки новых продуктов”.

Любая классификация становится предметом дискуссии, иногда бурной. Тем не менее, я не побоюсь предложить следующие типы проектов указанной категории - “проектов информационных систем”:

1. *Создание новых систем/приложений* - в англ. часто называются “from the scratch”, в той или иной степени подразумевая работу “с чистого листа” (“с нуля”);
2. *Развитие существующих систем/приложений* - подразумевает расширение доступной функциональности, которую невозможно или нецелесообразно выделять в отдельное приложение или систему; кстати, вводя такой тип программных проектов, мы решаем вопрос отделения функциональной деятельности, например, службы технической поддержки, от проектной, результатом которой как раз и будет новая версия системы, обладающая новой <требуемой> функциональностью; в качестве примера можно предложить выпуск новой версии системы управления проектами;
3. *Интеграция систем/приложений* - как существующих, так и вновь создаваемых, уделяя внимание в рамках содержания проектных работ только вопросам интеграции, включая те изменения, которые необходимо внести в системы для обеспечения возможности интеграции, но не более того;
4. *Проекты внедрения готовых программных систем*
 - a. *Приобретение (и развертывание) информационной системы (software acquisition*)* – часто выделяется как самостоятельный проект, а не только как один из процессов проекта; такая тенденция особенно заметна в крупных организациях, занимающихся выбором и закупкой необходимых готовых систем COTS – Commercial Off-The-Shelf (т.е. готовый коммерческий продукт, приобретаемый “с полки”);
 - b. *Адаптация корпоративных информационных систем (КИС)* – обычно, это процесс “кастомизации” (customization, англ., т.е. настройки и расширения) корпоративной системы, созданной внешним поставщиком как тиражируемое решение и требующее настройки в соответствии с бизнес-процессами конкретной организации, например, это системы ERP или CRM. Необходимо отметить, что расширение в рамках адаптации обычно подразумевает использование средств, предлагаемых самой системой (например, использование ABAP в SAP R/3 или SAP Business All-in-One), в противном случае – это уже становится дополнительным проектом категории 2. *развитие существующих систем* либо 3. *интеграция*, в зависимости от целей и объема работ.

* При работе с англоязычными источниками информации очень важно понимать, что термин “software acquisition” часто применяется не только для “приобретения программного обеспечения”, как такового. Более того, в официальных документах он связан с охватом всего комплекса процессов для заказа покупки и/или разработки программных средств у третьих сторон (независимых поставщиков) и в подконтрольных (дочерних) подразделениях, организациях и структурах, отвечающих за разработку приложений в крупных организациях. Поэтому иногда также используется термин “поставка программного обеспечения”.

Необходимо отметить, что отличия между такими типами проектов могут не влиять на жизненный цикл проекта, за исключением типа 4.а, который иногда выносится в самостоятельный тип или даже категорию (жизненный цикл проектов, в той или иной форме, будет обсуждаться на протяжении всей книги). Однако, наверняка, тип проекта будет придавать определенную специфику решениям в области формирования проектной команды, вовлеченности в проект тех или иных специалистов (в том числе, возможно, внешних по отношению к данному подразделению или организации в целом), структуре бюджета и, наверняка, вопросах коммуникации.

Вместе с тем, *проекты различных категорий и типов часто можно рассматривать и как программы*. Например, реализации проекта категории “3. Коммуникационные системы” по классификации Арчибальда может рассматриваться как программа, в рамках которой создание соответствующего программного обеспечения будет выделено в один или несколько проектов.

Расширения PMBOK в приложении к ИТ

PMBOK так описывает необходимость в расширении областей приложения [PMI PMBOK3, 2004, Рус, с.342]: “Расширение областей приложения необходимо в тех случаях, когда для отдельной категории проектов из одной области приложения имеются общепризнанные знания и практики, которые не являются общепринятыми для всех типов проектов в большинстве областей приложения. ... Специфичные для данной области приложения знания и практики могут определяться многими факторами, включая, в частности, различия в культурных нормах, технической терминологии, социального влияния или жизненных циклов проектов.”

Безусловно, индустрия программного обеспечения удовлетворяет указанным критериям области приложения, для которой могут быть разработаны соответствующие расширения PMBOK. Так как PMBOK не только одно из общепризнанных руководств по сводам знаний в области управления проектами, но и соответствующий американский стандарт, вполне естественно что появилось соответствующее расширение, освещающее, в частности, и вопросы индустрии информационных технологий. Речь идет о нескольких областях знаний Расширений PMBOK, подготовленных Defense Acquisition University (DAU) при министерстве обороны США [PMBOK US DoD Ext, 2003] на основе PMI PMBOK второй редакции (2000 год) [PMI PMBOK, 2000]. В силу специфики источника данного расширения, его применимость должна обсуждаться в каждом конкретном случае. В то же время, необходимо признать, что министерство обороны США является одним из крупнейших мировых потребителей информационных технологий, и его публично доступный опыт с определенными поправками может быть использован и в гражданских проектах за пределами США. Кроме того, это расширение также является стандартом PMI, что расширяет потенциал его применения.

Для нас, в контексте информационных технологий, наиболее интересны три из пяти областей знаний, добавленных в [PMBOK US DoD Ext, 2003].

1. (13) Project Systems Engineering Management
2. (14) Project Software Acquisition Management
3. (16) Project Test and Evaluation (T&E) Management

Рассмотрим первые две группы процессов – управление инженерной деятельностью и приобретением программного обеспечения, так как третья область детализирует вопросы управления тестированием и оценкой и выделена в отдельную составляющую по причине высокой значимости для получения качественного продукта.

Управление инженерной деятельностью в проекте

Именно такой перевод мне показался наиболее адекватным для группы процессов (13) Project Systems Engineering Management. Данная группа процессов включает три процесса:

1. (13.1) Планирование инженерной деятельности (Systems Engineering Planning)
2. (13.2) Инженерная деятельность (Systems Engineering Activities)
3. (13.3) Анализ и контроль (Analysis and Control)

Задача процессов этой группы состоит в исследовании, управлении и контроле работ, связанных с техническими аспектами проекта. “По своей природе <управление инженерной деятельностью> охватывает все функциональные дисциплины, необходимые для проектирования, разработки, тестирования, производства и поддержки продуктов.” [PMBOK DoD Ext, 2003, с.167]. Не забывайте, что в данном контексте “продукты” являются любым результатом проектных работ, будь то оборудование или программное обеспечение, что не мешает применять соответствующие практики и при создании программных систем и приложений.

Одной из ключевых техник управления инженерной деятельностью является комплекс процедур интегрированной разработки продуктов и процессов – Integrated Product and Process Development (IPPD), которым, например, в контексте разработки программного обеспечения уделяется специальное внимание в Capability Maturity Model Integration [CMMI 1.1, 2002], созданной в Институте Программной Инженерии университета Карнеги Меллон - Software Engineering Institute (SEI) Carnegie Mellon University (<http://www.sei.cmu.edu>).

Управление приобретением программного обеспечения

В оригинале эта группа процессов названа (14) Project Software Acquisition Management (SAM). Она описывает процесс “деятельность по приобретению ПО” – 14.1 SAM Activities, которая обычно включает (или может включать), в частности, следующие функции:

1. *Interoperability global information grid (GIG) (14.1.1.2)* – проверку соответствия требованиям интероперабельности, то есть прозрачности взаимодействия (обеспечения такового) в контексте существующих и планируемых к использованию информационных систем и/или заданных критериев интероперабельности;
2. *Capability Development Document (14.1.1.6)* – документ, подготавливаемый заказчиком/пользователем программной системы, описывающий ключевые высокоуровневые требования к системе с точки зрения параметров производительности, готовности/способности к интеграции и операционного окружения, в котором данная система будет эксплуатироваться;
3. *System/Sybsystem Specification (SSS) (14.1.1.7)* – спецификация системы и ее подсистем, включающая высокоуровневые требования и методы проверки соответствия системы этим требованиям;
4. *Systems Engineering Plan (14.1.1.9)* – подробно описывается в группе процессов управления инженерной деятельностью (13) и представляет собой общий (высокоуровневый) план и график работ, необходимых для получения необходимого программного продукта;
5. *Test and evaluation master plan (TEMP) (14.1.1.10)* – план работ по тестированию и оценке готовности программной системы;
6. *Software development and management plans (14.1.1.11)* – обычно включает Software Development Plan (SDP) или аналогичные планы-графики работ, описывающие жизненный цикл конкретного программного проекта; такой план базируется на нормативных актах, стандартах и/или регламентах, принятых в конкретной организации/подразделении;
7. *Software requirements (14.1.1.15)* – подробные требования к системе, детализирующие высокоуровневую спецификацию системы/подсистем (14.1.1.7); эту спецификацию в индустрии также часто называют *спецификацией требований к программному обеспечению* – Software Requirements Specification (SRS);
8. *Developer software capability evaluation (14.1.1.18)* - оценка возможностей разработчика программного обеспечения; требует от исполнителя работ, в частности, “полного соответствия Software Engineering Institute (SEI) Capability Maturity Model (CMM) уровня 3 или его эквивалента”. Такая позиция уже стала де-факто стандартным требованием к компаниям, выполняющим заказную разработку программного обеспечения для средних и крупных компаний и организаций во всем мире. Модель SEI CMM, а точнее CMMI -

Capability Maturity Model Integration [CMMI 1.1, 2002] будет рассматриваться в этой книге позднее.

Можно сказать, что функции этой группы процессов детализируют вопросы управления инженерной деятельностью (13) в применении к программному обеспечению. Если же касаться практик и подходов, связанных с такого рода деятельностью, то они вполне четко определены (приводится ряд общих техник, не связанных со спецификой данного источника Расширений РМВОК):

1. *Software development maturity assessments* (14.1.2.1) – оценка (может подразумевать сертификацию) зрелости <процессов> разработки программного обеспечения; эти вопросы мы будем рассматривать отдельно и более подробно;
2. *Software acquisition maturity assessments* (14.1.2.2) – оценка зрелости <процессов> приобретения ПО; подробно описаны в Software Acquisition моделях оценки зрелости CMM и CMMI (SA-CMM и SA-CMMI, соответственно);
3. *Software measures* (14.1.2.3) – метрики программного обеспечения; включают измеримые характеристики качества (quality metrics);
4. *Life-cycle standards tailoring* (14.1.2.4) – адаптация стандартов и методологий в области управления жизненным циклом; подразумевает, например, адаптацию стандарта ISO/IEC 12207 (или IEEE/IEC 12207, ГОСТ Р ИСО/МЭК 12207);

Таким образом, мы видим, что в приложении к индустрии программного обеспечения, стандарты и расширения PMI могут быть отличной базой для рассмотрения, адаптации и применения различных подходов в управлении разработкой прикладных систем. В принципе, можно выделить две ключевых группы аспектов, связанных именно с *управлением* такими работами (а не, например, с архитектурными и технологическими вопросами):

1. Модели, стандарты и методологии управления жизненным циклом программного обеспечения;
2. Модели и методы оценки зрелости и совершенствования процесса разработки;

Именно этим темам и будут посвящены следующие главы.