

Новиков Ф.А., к.ф.-м.н.

Опалева Э.А., к.т.н.

Степанов Е.О., к.ф.-м.н.

**Учебно-методическое пособие по дисциплине
«Управление проектами и разработкой
программного ПО»**

Оглавление

Тема 1. Основные понятия менеджмента	6
1.1. Введение	6
1.2. Понятие и виды менеджмента	6
1.3. Функции менеджмента	8
1.3.1. Прогнозирование и планирование	9
1.3.2. Создание организационных структур	10
1.3.3. Руководство командой	11
1.3.4. Координация	12
1.3.5. Контроль	12
1.4. Модели менеджмента	13
1.4.1. Особенности европейского менеджмента	14
1.4.2. Особенности американского менеджмента	14
1.4.3. Особенности японского менеджмента	15
1.4.4. Особенности российского менеджмента	16
1.5. Заключение	16
1.6. «Карта памяти» по теме 1	16
1.7. Список использованной и дополнительной литературы	17
Тема 2. Психология в менеджменте	19
2.1. Введение	19
2.2. Личность менеджера	19
2.2.1. Психологические роли менеджера по Г. Минцбергу	19
2.2.2. Профессиональная подготовка менеджера	22
2.2.3. Требования к личности менеджера. Психологический отбор	23
2.2.4. Тренинг в менеджменте	24
2.3. Мотивация	25
2.3.1. Потребности и мотивированное поведение	25
2.3.2. Теории мотивации	27
2.3.3. Особенности управления персоналом в области ИТ	37
2.3.4. Корпоративная культура	39
2.4. Заключение	42
2.5. «Карта памяти» по теме 2	42
2.6. Список использованной и дополнительной литературы	43
Тема 3. Тема 3. Основы принятия управленческих решений	44
3.1. Введение	44
3.2. Понятие и виды управленческих решений	44
3.2.1. Управленческие решения	44
3.2.2. Виды управленческих решений	45
3.3. Процесс принятия управленческих решений	48
3.3.1. Лицо, принимающее решение	48
3.3.2. Цели и ресурсы	49
3.3.3. Риски и неопределенности	50
3.3.4. Системный подход при принятии решений	51
3.4. Методы принятия управленческих решений	52
3.4.1. Игры с природой	53
3.4.2. Деревья принятия решений	56
3.4.3. Методы экспертных оценок	58
3.4.4. Проблема горизонта планирования	61
3.5. Заключение	62

3.6.	«Карта памяти» по теме 3	63
3.7.	Список использованной и дополнительной литературы	63
Тема 4.	Проектный менеджмент	64
4.1.	Введение	64
4.2.	Области эффективного приложения проектного менеджмента	64
4.3.	Типы проектов	65
4.4.	Жизненный цикл проекта разработки программного продукта	67
4.4.1.	Модели жизненного цикла программных продуктов	68
4.4.2.	История и эволюция	70
4.4.3.	Классификации моделей жизненного цикла	72
4.5.	Комплекс работ по внедрению технологии управления проектом в организации	78
4.6.	Функции менеджера проекта	80
4.7.	Основные стандарты в области управления проектами разработки программного обеспечения	81
4.7.1.	Классификация стандартов	81
4.7.2.	Организации по стандартизации	81
4.7.3.	Общие стандарты управления проектами	82
4.7.4.	Стандарты управления проектами разработки программного обеспечения	89
4.8.	Заключение	96
4.9.	«Карта памяти» по теме	97
4.10.	Список использованной и рекомендованной литературы	97
Тема 5.	Планирование проекта	98
5.1.	Введение	98
5.2.	Предынвестиционная фаза проекта	98
5.2.1.	Постановка задачи	99
5.2.2.	Экспертиза идеи специалистами	99
5.3.	Планирование проекта: определение целей проекта	100
5.3.1.	Планирование целей	100
5.3.2.	Описание целей проекта	101
5.3.3.	Подтверждение целей проекта	102
5.3.4.	Контроль изменения целей	102
5.4.	Виды планов	103
5.5.	Методика разработки и анализа плана проекта	104
5.5.1.	Выявление и анализ критического пути	105
5.5.2.	Методики оценок времени и затрат	106
5.5.3.	Распределение работ	106
5.6.	Структура декомпозиции работ (WBS)	107
5.6.1.	Основные термины и определения	107
5.6.2.	Необходимость использования WBS	109
5.6.3.	Разработка структуры декомпозиции работ	110
5.6.4.	Определение приемлемого уровня детализации	112
5.6.5.	Взаимосвязь между риском проекта и WBS	113
5.6.6.	Взаимосвязь планирования и контроля ресурсов и WBS	114
5.6.7.	Разработка WBS	115
5.7.	Разработка проектно-сметной документации	116
5.7.1.	Материально-техническая подготовка проекта	116
5.7.2.	Типовая смета расходов	117
5.8.	Организационная структура исполнителей	118
5.8.1.	Иерархическая модель	120
5.8.2.	Модель «Бригада главного программиста»	121

5.8.3.	Модель «Команда равных».....	125
5.8.4.	Матрица ответственности.....	126
5.9.	Прикладные программные средства для менеджера проекта	127
5.10.	Заключение	128
5.11.	«Карта памяти» по теме	129
5.12.	Список использованной и рекомендованной литературы	129
Тема 6.	Бизнес-анализ и управление проектом	131
6.1.	Введение	131
6.2.	Проекты разработки и внедрения корпоративных информационных и управляющих систем.....	131
6.3.	SADT — методология структурного анализа и проектирования.....	134
6.3.1.	Основные принципы SADT	135
6.3.2.	Стандарт IDEF0. Синтаксис диаграмм, применяемых для функциональных моделей	140
6.3.3.	Стандарт и методология моделирования IDEF3	145
6.4.	Моделирование данных.....	154
6.4.1.	Диаграммы «сущность-связь».....	155
6.4.2.	Выражение модели сущность-связь средствами UML	159
6.5.	Заключение	160
6.6.	«Карта памяти» по теме	160
6.7.	Список использованной и дополнительной литературы	161
Тема 7.	Кадровый менеджмент	162
7.1.	Введение	162
7.2.	Кадровая политика фирмы.....	162
7.3.	Основные принципы и методы управления персоналом	163
7.3.1.	Анализ работы персонала	164
7.3.2.	Планирование потребности в персонале.....	169
7.3.3.	Подбор и наём работников	173
7.3.4.	Увольнение работников	185
7.4.	Персонал проекта по разработке программного обеспечения	186
7.4.1.	Управление персоналом проекта	186
7.4.2.	Развитие команды проекта	188
7.5.	Заключение	189
7.6.	«Карта памяти» по теме	189
7.7.	Список использованной и дополнительной литературы	190
Тема 8.	Управление временем проекта	191
8.1.	Введение	191
8.2.	Календарный план проекта	191
8.2.1.	Создание списка задач и структуры декомпозиции работ	192
8.2.2.	Оценка длительности задач и трудозатрат	193
8.2.3.	Создание зависимостей между задачами	194
8.2.4.	Получение удобного представления календарного плана.....	196
8.3.	Сетевое представление проекта.....	198
8.4.	Заключение	202
8.5.	Список использованной и рекомендованной литературы	202
8.6.	«Карта памяти» по теме	203
Тема 9.	Управление ресурсами проекта.....	204
9.1.	Введение	204
9.2.	Ресурсы проекта	204
9.2.1.	Назначение ресурсов при планировании	204
9.2.2.	Загрузка ресурсов	207
9.3.	Распределение ресурсов и календарный план	208

9.3.1.	Трудозатраты, объем и длительность работы.....	208
9.3.2.	Выравнивание загрузки ресурсов	209
9.4.	Заключение	211
9.5.	Список использованной и рекомендованной литературы	211
9.6.	«Карта памяти» по теме	212
Тема 10.	Организация проектного финансирования	213
10.1.	Введение	213
10.2.	Финансовый менеджмент.....	213
10.2.1.	Менеджмент проекта и финансовый менеджмент.....	213
10.2.2.	Функции менеджера проекта в управлении финансами.....	216
10.2.3.	Общие принципы управления финансами проекта.....	216
10.2.4.	Оценка финансовой эффективности проекта	217
10.3.	Оценки стоимостей и планирование затрат	220
10.3.1.	Грубая предварительная оценка стоимости проекта	220
10.3.2.	Планирование затрат и составление смет	226
10.4.	Заключение	227
10.5.	«Карта памяти» по теме	227
10.6.	Список использованной и рекомендованной литературы	228
Тема 11.	Управление рисками проекта	229
11.1.	Введение	229
11.2.	Риски при разработке программного обеспечения.....	229
11.2.1.	Риск и неопределенность	230
11.2.2.	Характеристики риска.....	230
11.2.3.	Причины возникновения рисковых ситуаций	232
11.3.	Порядок управления рисками программных проектов	233
11.3.1.	Основные модели управления рисками программных проектов	234
11.3.2.	Выявление, идентификация и анализ рисков	235
11.3.3.	Планирование ответов на риски. Методы снижения риска	241
11.3.4.	План управления рисками	242
11.3.5.	Мониторинг рисков.....	243
11.4.	Заключение	244
11.5.	«Карта памяти» по теме	244
11.6.	Использованная литература.....	244
Тема 12.	Особенности аутсорсинга	245
12.1.	Введение	245
12.2.	Аутсорсинг.....	245
12.2.1.	Что такое аутсорсинг?.....	246
12.2.2.	Плюсы и минусы аутсорсинга.....	247
12.3.	Аутстаффинг.....	248
12.4.	Аутсорсинг информационных технологий	250
12.4.1.	Аутсорсинг программных проектов	251
12.4.2.	Виды аутсорсинга программных проектов	252
12.5.	Заключение	254
12.6.	Список использованной и рекомендованной литературы	255
12.7.	«Карта памяти» по теме	255

Тема 1. Основные понятия менеджмента

1.1. Введение

Менеджмент – это синоним русского слова «управление». Менеджмент проектов разработки программного продукта имеет как общие аспекты, присущие всем видам менеджмента, так и специальные особенности, характерные именно для процессов разработки программного продукта. В этой теме рассматриваются основные понятия менеджмента применительно к разработке программных продуктов.

Изучив учебный материал данной темы, Вы:

- узнаете или пополните свои знания о видах менеджмента;
- узнаете или пополните свои знания о функциях, выполняемых менеджером;
- сможете выбрать модель менеджмента, наиболее подходящую для конкретной разработки.

В рамках темы рассматриваются следующие учебные вопросы:

- Определение понятия «менеджмент»
- Виды менеджмента
- Функции менеджера
- Модели менеджмента

1.2. Понятие и виды менеджмента

В условиях развитого рынка программного обеспечения и жесткой конкуренции важнейшее значение имеют:

- возможность получения максимальной прибыли;
- сроки разработки;
- тесное взаимодействие с другими коллективами разработчиков.

При профессиональной разработке программного продукта в крупных фирмах и компаниях предъявляются дополнительные требования к процессу разработки:

- наличие формализованной модели для разработки программного продукта;
- наилучшая расстановка приоритетов и ресурсов;
- принятие ясных и документированных решений;
- соблюдение многочисленных стандартов;

- обеспечение эффективного взаимодействия между командами разработчиков, быстрое создание и вхождение в процесс разработки таких команд;
- тесное взаимодействие с группами поддержки и продаж.

Выполнение этих требований предполагает эффективное (по определенному критерию) управление процессом. Каждая фирма строит свою деятельность, ориентируясь на основной критерий – прибыль. Получение прибыли связано с необходимостью производить для рынка товары (например, программный продукт) и услуги (например, консультации по программному продукту). Как правило, речь идет о производстве только той продукции, в которой нуждается потребитель.

Стратегия предпринимательской деятельности фирмы строится исходя из задачи увеличения прибыли, объединяющей пять направлений [4]:

- стратегия в области исследования и развития;
- оперативная стратегия;
- финансовая стратегия;
- маркетинговая стратегия;
- стратегия человеческих отношений.

Согласно этим направлениям, управление фирмой можно поделить на три составные части:

- производственную;
- финансовую;
- маркетинг.

Определим менеджмент как систему принятия решений в области управления фирмой:

Менеджмент – это система принятия управленческих решений.

Таким образом, можно выделить три вида менеджмента:

- производственный менеджмент;
- финансовый менеджмент;
- маркетинговый менеджмент.

Управленческие решения принимают люди. Поделим всех служащих компании на четыре уровня.

- **Высший менеджмент** (президент фирмы, председатель совета директоров, исполнительный президент, вице-

президенты). В его задачу входит определение генеральной линии компании.

- **Управляющие среднего звена** (менеджеры структурных подразделений). В их задачу входит координация и стратегическое планирование деятельности структурного подразделения.
- **Управляющие первичного звена** (менеджеры проектов, менеджеры-координаторы). Одна из важнейших задач этой категории управляющих – планирование на уровне программных проектов. Далее мы сконцентрируемся в основном на изучении обязанностей менеджеров проекта.
- **Инженеры** (внеменеджментовые служащие). Именно инженеры заняты разработкой и созданием программного продукта.

1.3. Функции менеджмента

В кабинетах многих менеджеров висят плакаты со словами Анри Файоля:

«Управлять – значит прогнозировать и планировать, организовывать, руководить командой, координировать и контролировать».

В этих словах одного из основоположников научного менеджмента сформулированы основные функции управления.



Анри Файоль
(*Henri Fayol*)
(1841-1925)

Анри Файоль по образованию был горным инженером. Автор первой законченной концепции менеджмента, в которой он выделил ставшие уже классическими понятия “функции управления”, “принципы менеджмента”. Известен также как практический менеджер и преподаватель управленческих дисциплин. Основные работы вышли в период 1916 – 1930 г.г. В 1916 был опубликован труд Файоля "Основные черты промышленной администрации – предвидение, организация, распорядительство, координирование, контроль". Этот труд – основной вклад Файоля в науку об управлении.

Выделенные Файолем пять функций менеджмента дают основу для анализа работы современного управляющего.

Рассмотрим их подробнее.

1.3.1. Прогнозирование и планирование

В работе менеджера прогнозирование и планирование тесно связаны.

Прогнозирование – взгляд в будущее, оценка возможных путей развития, последствий тех или иных решений.

Планирование – разработка последовательности действий, позволяющей достигнуть желаемого.

Иногда прогноз основан на хорошо изученных закономерностях и осуществляется наверняка. Никто не сомневается, что вслед за ночью наступит день. Методы прогнозирования движения космических аппаратов разработаны настолько, что возможна автоматическая стыковка кораблей. Однако проблемы, встающие перед менеджером проектов разработки программного продукта, обычно не позволяют дать однозначный обоснованный прогноз.

Простейшие методы прогнозирования основаны на восстановлении зависимостей путем экстраполяции. В случае управления разработкой программных продуктов экстраполяция возможна, только если есть в наличии достоверные данные (метрики) о ранее разработанных продуктах. Именно поэтому измерение процессов разработки имеет столь большое значение.

Согласно концепции немецкого профессора Д. Хана планирование – это ориентированный в будущее систематический процесс принятия решений.

Обычно выделяют восемь этапов в процессе планирования.

1. **Целеполагание (формулировка целей).** Чего именно вы (или ваша фирма) хотите достичь? Это – самый трудный этап. Его нельзя формализовать. Личность менеджера проявляется именно в том, какие цели он ставит.
2. **Подбор, анализ и оценка способов достижения поставленных целей.** Обычно можно действовать разными способами. Какой из них представляется наилучшим? Какие можно сразу отбросить как нецелесообразные?
3. **Составление перечня необходимых действий.** Что конкретно нужно сделать, чтобы осуществить выбранный на предыдущем этапе вариант достижения поставленных целей?
4. **Составление программы работ (плана мероприятий).** В каком порядке лучше всего выполнять намеченные на предыдущем этапе действия, учитывая, что многие из них связаны между собой?

5. **Анализ ресурсов.** Какие материальные, финансовые, информационные, кадровые ресурсы понадобятся для реализации плана? Сколько времени уйдет на его выполнение?
6. **Анализ разработанного варианта плана.** Решает ли разработанный план поставленные на этапе 1 задачи? Являются ли затраты ресурсов приемлемыми? Есть ли соображения по улучшению плана, возникшие в ходе его разработки при движении от этапа 2 к этапу 5? Возможно, целесообразно вернуться к этапу 2 или 3, или даже к этапу 1.
7. **Подготовка детального плана действий.** Необходимо детализировать разработанный на предыдущих этапах план, выбрать согласованные между собой сроки выполнения отдельных работ, рассчитать необходимые ресурсы. Кто будет отвечать за отдельные участки работы?
8. **Контроль за выполнением плана.** Выполняется ли план? Имеются ли основания для внесения необходимых изменений? Контроль как функция менеджмента будет обсуждаться в одном из следующих разделов.

1.3.2. Создание организационных структур

Эту функцию менеджмента превосходно выразил великий полководец А.В.Суворов: «Каждый солдат должен знать свой маневр».

Таким образом, каждый сотрудник фирмы должен знать, что ему надо делать в той или иной ситуации. Лишь в очень маленьких организациях менеджер может сам рассказать каждому об его обязанностях.

Возможности психики любого человека ограничены – психологи установили, что *число непосредственных подчиненных у начальника*, который ежедневно с ними работает, *должно быть не более семи* (если больше – деловой контакт оказывается поверхностным). Поэтому создают иерархические системы управления – рядовой работник действует под началом руководителя первичного звена, тот имеет дело с руководителем среднего звена и так далее до высшего менеджмента.

Обычно у первого лица предприятия (менеджера, генерального директора) четыре непосредственных помощника:

- по производству (технический директор или главный инженер),
- по финансам (финансовый директор, иногда главный бухгалтер),
- по маркетингу (директор по маркетингу, иногда начальник отдела сбыта),

- по персоналу (начальник управления кадров, директор по кадрам).

Каждому из них подчиняются свои службы, состоящие из отделов, групп и иных структур.

Письменные инструкции, указывающие обязанности сотрудников и правила действий в тех или иных случаях, также имеют целью согласованность действий и обеспечивают единство фирмы как хозяйствующего субъекта. Организационную структуру фирмы можно сравнить со скелетом живого существа, но только скелет рукотворный – его создает и меняет главный менеджер.

1.3.3. Руководство командой

Команда – это те, с кем менеджер работает ежедневно. Высокий профессионализм и ответственность членов команды, слаженность их работы, взаимная поддержка обеспечивают успех. И наоборот, плохой подбор команды может сделать беспомощным даже самого сильного менеджера. Если приказы не исполняются, файлы теряются, встречи срываются из-за бестолковости сотрудников, то ожидать эффективной работы фирмы невозможно.

Создание команды – одно из самых важных дел менеджера. Можно сказать, что команда – его основной инструмент работы. Недаром, меняя место службы, менеджер часто «перетаскивает» за собой и свою команду. Большое значение имеет психологическая совместимость членов команды. В ней не должно быть случайных людей. Возникновение ссор и раздоров в команде сильно снижает ее эффективность. Поэтому зачинщиков ссоры целесообразно удалить из команды, даже если их профессионализм весьма высок.

Менеджер управляет с помощью распоряжений, приказов, которые с помощью своей команды доводит до всех подчиненных и добивается их выполнения. Приказы и распоряжения могут быть письменные и устные, а также доведенные до адресата с помощью компьютера и электронной почты. Они должны однозначно восприниматься исполнителями, быть ясными, четкими и по возможности короткими. Иногда нужна вводная часть, разъясняющая необходимость данного приказа.

Важные приказы, особенно касающиеся сложных проблем, необходимо оформлять письменно. Связано это, прежде всего, с тем, что каждый из собеседников запоминает разговор по-своему. Обычно в памяти остается то, что выгодно данному лицу.

Приказы оформляются по правилам, принятым в делопроизводстве. Необходимо помнить, что некоторые виды приказов, в частности, по кадровому составу, могут быть обжалованы в суде.

В деятельности многих менеджеров переплетены решения управленческих и профессиональных задач. Например, главный инженер химического завода – не только менеджер, но и инженер. Главный врач больницы должен уметь не только управлять, но и лечить. Менеджеры такого типа должны время от времени демонстрировать свои возможности в профессиональной деятельности, а проявленная ими некомпетентность сильно снижает уважение сотрудников.

1.3.4. Координация

Одно из основных условий успешной деятельности организации – согласованность действий менеджеров этой организации. Они не только не должны противоречить друг другу, напротив, необходимо, чтобы они дополняли друг друга и вели к одной цели – цели фирмы, выраженной в долгосрочных и оперативных планах.

Обратим внимание на важность эффективной организации потоков информации. Она должна быть достаточной, но не излишней.

Поэтому совершенно необходимы ***регулярные совещания менеджеров***. Не так легко грамотно подготовить и провести совещание, добиться принятия полезных для работы фирмы решений. Необходимо заранее обеспечить участников необходимой информацией, организовать деловую дискуссию, подавлять основанную на эмоциях перепалку и одновременно не превращать заседание в монолог начальника, и т.д. Это – наука и искусство, которыми должен владеть хороший менеджер.

Координация действий менеджеров происходит и при ***подготовке документов*** – планов, приказов, предложений, направляемых в другие организации, ответов на распоряжения и запросы властей и др.

Обычно один из сотрудников – назовем его Исполнителем – готовит первоначальный вариант документа. Он размножается и рассылается на отзыв заинтересованным в нем менеджерам, а иногда и в другие организации. Исполнитель составляет сводку отзывов, с одними из замечаний соглашается, против других высказывает возражения. Затем собирают так называемое «согласительное совещание», на которое приглашают всех тех, с чьим мнением Исполнитель не согласен. В результате дискуссии по ряду позиций достигается компромисс, и возражения снимаются.

1.3.5. Контроль

Сколько бы хорошо ни были разработаны планы, они, как правило, не могут быть выполнены так, как были задуманы. Будущее нельзя абсолютно точно предсказать. Неблагоприятные условия, аварии на производстве и на транспорте, болезни и увольнения сотрудников и многие другие причины нарушают наши планы. Эти нарушения, прежде всего, надо обнаружить с помощью *системы контроля*. Например, надо регулярно – раз в день, неделю или месяц – возвращаться к плану и выявлять нежелательные отклонения от запланированного.

Есть два основных подхода к отклонениям:

- Во-первых, можно стремиться к возврату на плановую траекторию движения. Для этого понадобятся дополнительные ресурсы – материальные, кадровые, финансовые. Иногда такие ресурсы создают согласно плану, заранее предвидя возможность осложнений.
- Во-вторых, можно изменить сам план, заменив намеченные рубежи на другие, реально достижимые в создавшейся обстановке. Возможность такого подхода зависит от того, насколько для фирмы важен план – является ли он «законом» или же только «руководством к действию», задающим желательное направление движения.

На менеджере лежит обязанность *контроля за выполнением принятых ранее решений*, не только включенных в план, но и оперативных, текущих. Частично контроль осуществляется в ходе совещаний и визирования документов. Но этого недостаточно. При планировании собственной работы менеджеру следует предусмотреть регулярные проверки деятельности своих подчиненных, причем не только членов своей команды, но и всех остальных. Могут применяться как официальные отчеты и аттестации, так и неформальные беседы, например, выборочная инспекция разработанного программного кода. Надо отметить, что беседа с менеджером, стоящим на иерархической лестнице на несколько ступенек выше, производит большое положительное воздействие на сотрудника.

В Великобритании считается, что генеральный директор должен побеседовать с каждым сотрудником хотя бы раз в год. К сожалению, в России подобные собеседования не приняты.

1.4. Модели менеджмента

Основы менеджмента были заложены в начале XX века в европейской и американской науке. История менеджмента связана с древним Египтом и философами античности. В настоящее время можно говорить об особенностях европейского, американского, японского и

российского менеджмента. Остановимся на каждой модели немного подробнее.

1.4.1. Особенности европейского менеджмента

Европейский менеджмент имеет очень хорошую теоретическую базу, основанную на работах Макса Вебера (Германия), Анри Файоля (Франция), Кароля Адамецки (Польша) и многих других.

Приведем несколько принципов управления Файоля, упор на которые является основой европейского менеджмента [5]:

- Дисциплина, т.е. послушание и уважение к достигнутым соглашениям между фирмой и ее работниками. Дисциплина предполагает также справедливо применяемые санкции.
- Вознаграждение персонала, в том числе справедливая зарплата.
- Справедливость: сочетание доброты и правосудия.
- Корпоративный дух, т.е. гармония персонала, его сплочение.
- Подчиненность личных интересов общим. Интересы отдельного работника или группы не должны превалировать над интересами компании.

1.4.2. Особенности американского менеджмента

Менеджеры как социальная группа составляют в Америке около 15% работающего населения. Условием квалифицированного руководства считается соблюдение трех правил:

- Анализируй состояние инженеров, запросы рынка, процесс создания программного продукта.
- Разделяй обязанности среди подчиненных.
- Контролируй исполнение.

Американский менеджмент поощряет конкуренцию среди инженеров [5]. Конкуренция представляется им необходимым условием социального и личностного развития. Это приводит к ряду негативных явлений, например, сокрытию информации друг от друга и поиску более легкого, а не оптимального пути решения задачи. Наиболее слабым местом американского менеджмента является неполное использование человеческого фактора в управлении.

Об особенностях американского менеджмента в области разработки программного обеспечения и о влиянии передовых информационных технологий на современный бизнес достаточно подробно рассказано в книге Билла Гейтса (Bill Gates) [1].

1.4.3. Особенности японского менеджмента

На основе анализа деятельности японских компаний, И. Олстон сформулировал пять основных принципов японского менеджмента [5].

- Работник достаточно разумен, он самостоятельно увеличивает производительность и качество своего труда. Три административных метода помогают претворить этот принцип в жизнь:
 - кружки качества;
 - практика стимулирования всех работников к совершенствованию профессиональных умений и навыков. Здесь очень интересно то, что японская традиция «минарай» – наблюдение за опытными работниками с целью освоения их навыков – очень близка русской традиции наставничества;
 - практика перевода работников с одного места на другое. Работник повышает квалификацию, меняет специализацию и продвигается по служебной лестнице.
- Работник стремится сделать свою работу лучше. Методы реализации этого принципа:
 - пожизненный найм работников;
 - премирование рабочих в случае повышения прибыли фирмы. Заметим, что в неблагоприятной ситуации руководящий состав получает значительно более урезанную зарплату, чем рабочие.
- Работники образуют «семью». Методы реализации принципа:
 - работники фирмы оказывают новичку помощь, сочувствие и поддержку, ожидая от него в дальнейшем проявления такого же поведения по отношению к ним;
 - свободное время работники фирмы проводят вместе;
 - взаимные обязательства фирмы и работников. Если сотрудник фирмы вступает в брак или у него рождается ребенок, то он получает денежную прибавку, поскольку его расходы возросли.
- Группа важнее отдельной личности. Этот принцип основан на традиционной японской ценности – никто не должен быть эгоистичным и думать только о себе.
- Партнерство и сотрудничество в производственных отношениях.

1.4.4. Особенности российского менеджмента

Понимание необходимости целенаправленной подготовки специалистов для управленческой деятельности на научной основе сложилось в нашей стране лишь в 60-е годы XX века. Российский менеджмент должен учитывать особенности национальной психологии, исторический опыт, традиции и ценности, которые веками складывались в России. Существует ряд приоритетных ценностей, на которые российский менеджмент должен опираться [5]:

- Социальные права личности.
- Честь и достоинство личности.
- Согласие и сотрудничество в обществе.
- Справедливость, основанная на понимании и принятии правовых норм.
- Профессионализм и мастерство. Коллективизм, поддержка и взаимопомощь.
- Духовность и психическое здоровье в основе деловой и поведенческой практики.

1.5. Заключение

Менеджмент проектов разработки программного продукта имеет все основные характеристики, присущие менеджменту вообще, а также некоторые специальные аспекты, связанные с особенностями процесса разработки программного обеспечения.

Прекрасную формулировку особенностей менеджера проектов разработки программного продукта дал Том де Марко в своей книге «Сделать в срок» [6].

Части тела, необходимые для управления проектами:

- ⊙ Для руководства нужны сердце, нутро, душа и нюх.
- ⊙ Руководить надо сердцем.
- ⊙ Чувствовать нутром.
- ⊙ Вкладывать в команду и проект душу.
- ⊙ Иметь нюх на всякую ерунду и бессмыслицу.

1.6. «Карта памяти» по теме 1



1.7. Список использованной и дополнительной литературы

1. Гейтс Б. Бизнес со скоростью мысли.- М.: ЭКСМО-Пресс, 2001.
2. Одинцов И. Профессиональное программирование. Системный подход.- СПб.: BHV-СПб, 2002.
3. Орлов А. Менеджмент: учебник. - М.: Знание, 1999.
4. Пашкус Ю. В., Мисько О. Н. Введение в бизнес. - Л.: "Северо-Запад", 1991.
5. Психология менеджмента: учебник. - СПб.: Изд-во Санкт-Петербургского университета, 2000.
6. T. DeMarco. The Deadline: A Novel About Project Management. Dorset House Publishing Co., N.Y., 1997.

Тема 2. Психология в менеджменте

2.1. Введение

Психология (от греч. *psyche* – душа, *logos* – учение) – это наука, изучающая субъективные аспекты поведения человека. Психология в менеджменте, особенно в менеджменте проектов по разработке программного продукта имеет ряд специальных черт.

В этой теме рассматриваются основные понятия психологии в менеджменте применительно к информационным технологиям вообще и разработке программных продуктов в частности.

Изучив учебный материал данной темы, Вы:

- узнаете или пополните свои знания о психологических особенностях личности менеджера;
- узнаете или пополните свои знания о потребностях, регулирующих поведение, и мотивации, как способа влиять на него;
- познакомитесь с особенностями управления персоналом в проектах в области информационных технологий.

В рамках темы рассматриваются следующие учебные вопросы:

- Личность менеджера
- Потребности и мотивация
- Управление персоналом
- Корпоративная культура

2.2. Личность менеджера

2.2.1. Психологические роли менеджера по Г. Минцбергу

Рассмотрим основные психологические роли, в которых в разных ситуациях выступают менеджеры различных уровней.



Генри Минцберг (Henry Mintzberg) – изучал природу управленческого труда, автор первой ролевой модели, описывающей профессиональную деятельность менеджера; известен также как разработчик концепции пяти основных организационных структур. Основные работы вышли в период 1970 – 1990 г.г.

Название психологической роли	Описание действий	Примеры из практики управления, требующие активизации соответствующей роли
Межличностные роли		
Главный руководитель (лицо организации)	Символический глава организации, выполняющий обязанности социального или правового характера	Церемониалы, ходатайства, представления, проведение приемов, все другие действия, обязываемые руководящей должностью
Лидер	Мотивация подчиненных, взаимодействие с ними, подбор и подготовка работников	Практически все управленческие действия с участием подчиненных
Связующее звено	Установление сети контактов с руководителями и специалистами других подразделений и организаций, а также информирование подчиненных об этих контактах	Деловая переписка, участие в совещаниях, с представителями других подразделений (организаций)
Информационные роли		
Приемник информации (монитор)	Целенаправленно ищет, получает и накапливает разнообразную информацию, необходимую для работы	Обработка поступающей корреспонденции, периодические осмотры, ознакомительные поездки, посещения информационных семинаров и выставок
Распространитель информации (передатчик)	Передает информацию, полученную как из внешних источников, так и от подчиненных, всем	Распространение информационных справок, дайджестов, выступления с обзорами, проведение бесед, информирование подчиненных о

	заинтересованным лицам внутри организации	достигнутых договоренностях
Представитель (рупор организации)	Передаёт информацию о планах, нынешнем состоянии, результатах работы подразделений (организаций) внешним лицам и органам	Подготовка и рассылка информационных писем, участие в заседаниях с отчётами о проделанной работе
Управленческие роли (роли связанные с принятием решений)		
Предприниматель (инициатор)	Ищет возможности для совершенствования как процессов внутри организации, так и системы взаимосвязей с другими подразделениями и структурами, является инициатором внедрения новшеств, направленных на улучшение положения дел и работников	Участие в заседаниях с обсуждением и принятием перспективных решений, совещания, посвященные ходу внедрения тех или иных новшеств
Устранитель проблем (стабилизатор)	Забывается об организации, вносит коррективы в действия, берет ответственность на себя, когда возникают факторы, угрожающие существованию и нормальной деятельности организации	Обсуждение и принятие решений по стратегическим текущим вопросам, связанным с поиском путей выхода из кризисных ситуаций
Распределитель ресурсов	Определяет, как будут расходиться материальные, финансовые и трудовые ресурсы организации	Составление и утверждение графиков, планов, смет, бюджетов, контроль за из исполнением
Ведущий	Представляет	Ведение переговоров,

переговоры (посредник)	организацию на всех значительных переговорах	установление официальных контактов между организацией и другими фирмами
-----------------------------------	--	--

2.2.2. Профессиональная подготовка менеджера

Менеджер – это профессия, а не должность.

Приведенное определение предусматривает наличие у квалифицированного менеджера специальной профессиональной подготовки. Профессиональная подготовка менеджера проводится в двух направлениях:

- специальная подготовка в предметной области управления (в случае, если речь идет о менеджменте в области информационных технологий, то это подготовка в области современных технологий разработки программного обеспечения, телекоммуникациях, использовании современных инструментальных средств и т.п.);
- подготовка в области собственно менеджмента, которая в известной мере не зависит от конкретной предметной области.

Современный менеджер вне зависимости от конкретной предметной области управления должен:

- знать основные категории экономической науки и социологии, понимать суть социально-экономических явлений, владеть методами анализа экономических и социальных процессов;
- понимать законы функционирования организаций, уметь анализировать и осуществлять основные функции менеджмента;
- владеть практическими навыками менеджера (осуществление коммуникаций, принятие управленческих решений, управление конфликтами и стрессами и др.).

Менеджер проектов в области информационных технологий должен кроме того обладать рядом специальных знаний, умений и навыков, а именно:

- быть в курсе современных информационных технологий, в том числе технологий разработки ПО, и владеть специальным профессиональным жаргоном;

- быть легко обучаемым и постоянно совершенствовать свои знания в области ИТ (все соответствующие технологии имеют весьма короткий срок жизни);
- умение работать с командой узких технических специалистов с высокой или завышенной самооценкой.

2.2.3. Требования к личности менеджера. Психологический отбор

Требования, предъявляемые к личности менеджера, можно разделить на следующие категории:

- **Психологические требования** (интеллектуальный уровень, мотивационная сфера, коммуникативная сфера, приоритетные личностные черты, свойства и качества). Обычно от менеджера требуется ответственность, коммуникабельность, умение работать в команде, лидерство, умение находить компромиссы и разрешать конфликты.
- **Требования к уровню образования.** От менеджера проектов по разработке программного обеспечения требуется высокая техническая квалификация в технологиях и инструментальных средствах, используемых при проектировании и разработке ПО в конкретных условиях. Это как правило подразумевает высшее техническое (или физико-математическое) образование, хотя и не обязательно специальное образование в области информационных технологий.
- **Требования к опыту работы.** Менеджер проектов в области разработки ПО должен иметь значительный опыт проектирования и разработки ПО в качестве исполнителя – программиста или системного архитектора (но, как правило, не тестера или технического писателя).
- **Требования к профессиональным знаниям, умениям и навыкам.** Желателен широкий кругозор в области современных технологий проектирования и разработки ПО. Поскольку средний срок жизни этих технологий невелик, это влечет необходимость постоянного профессионального обучения.
- **Дополнительные требования.** Для менеджеров аутсорсинговых проектов, или, шире, проектов с иностранными заказчиками, необходимо владения иностранным языком, как минимум, английским, на уровне коммуникации.

2.2.4. Тренинг в менеджменте

Изучение основных понятий и закономерностей современного менеджмента может быть организовано на двух уровнях, отличающихся друг от друга степенью вовлеченности слушателей в учебный процесс. Это:

- **пассивно-созерцательный уровень**, когда обучающиеся прослушивают на лекциях соответствующие темы учебного плана и, самое большее, пытаются применить усвоенные схемы к типовым задачам на практических занятиях;
- **активно-деятельный уровень**, обучение, которое построено на создании собственного управленческого опыта путем выработки навыков поведения и принятия решений в неоднозначных, слабо структурированных ситуациях, когда достижению цели препятствуют объективные обстоятельства и намерения других людей.

Несмотря на то, что пассивно-созерцательный уровень имеет не очень высокую эффективность обучения и непривлекательное название, представляется все же невозможным полностью отказаться от него. Ведь именно здесь закладываются основы знаний по соответствующей учебной дисциплине, формируются логические схемы и концепции для последующего анализа собственной управленческой практики. Вместе с тем решающее значение для успешного формирования управленческих навыков в ходе обучения приобретает активно-деятельный уровень, позволяющий уже в ходе занятий моделировать будущую деятельность в качестве менеджера. Отсюда возникает необходимость подготовки методического обеспечения для проведения занятий на активно-деятельном уровне, включающем как саму схему проведения упражнений, собственно управленческие ситуации, так и технические средства обучения, направленные на обеспечение необходимой информации и установление обратной связи.

Основной формой пассивно-созерцательного уровня обучения являются лекции, а основной формой активно-деятельного обучения являются ролевые тренинги. В ходе ролевого тренинга перед участниками ставятся следующие задачи:

- повторно (с учетом знаний, полученных на лекциях) познакомиться с особенностями управленческой деятельности, спецификой труда руководителя, а также с основными функциями менеджера;
- уяснить специфику управленческих действий, характерных для каждой из ролей менеджера;

- выработать навыки эффективного поведения в различных управленческих ситуациях, требующих проигрывания соответствующих ролей;
- сформировать навыки анализа и самоанализа успешности управленческого поведения с точки зрения соответствия той или иной роли, а также с точки зрения качества принятого управленческого решения.

По результатам ролевого тренинга может быть проведено заключительное обсуждение, а соответствующие теоретические вопросы могут быть вынесены на экзамен (зачет) или включаются в тест итогового контроля знаний.

2.3. Мотивация

Слово «мотивация» происходит от латинского «*motus*» – движение. Можно определить мотивацию как методы воздействия на людей (в том числе и на себя самого) с целью получения желаемого результата.

2.3.1. Потребности и мотивированное поведение

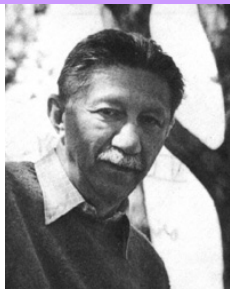
Потребность – осознанная недостаточность чего-либо.

Именно потребности заставляют людей действовать определенным образом. Выделяют две основные группы потребностей.

- **Первичные**, являющиеся физиологическими по своей природе (например, потребность в пище).
- **Вторичные**, являющиеся психологическими (например, потребность в уважении).

Следует отметить, что не все потребности являются сознательными. Есть и бессознательные (неосознаваемые в данный момент данным индивидуумом) потребности.

Потребности можно удовлетворять с помощью вознаграждения. Менеджеры могут использовать внутренние вознаграждения (которые дает сама работа) и внешние (денежные выплаты и продвижение по службе).



А. Маслоу

Маслоу Абрахам Харольд (Abraham Maslow) (1908-1970) – американский психолог. Выдвинул концепцию целостного подхода к человеку и анализа его высших сущностных проявлений – любви, творчества, духовных ценностей и др.

Абрахам Маслоу в 40-х годах XX века разделил потребности на пять категорий.

1. **Физиологические потребности**, являющиеся необходимыми для выживания (еда, жилище, одежда).
2. **Потребности в безопасности**, включающие потребности в защите от физических и психологических опасностей и уверенность в удовлетворении в будущем физиологических потребностей (пенсия).
3. **Социальные потребности**, заключающиеся в привязанности, принадлежности к какой-либо общности, дружбе.
4. **Потребности в уважении**, включающие внутренние (самоуважение, личные достижения) и внешние (статус, признание, одобрение со стороны других) факторы уважения.
5. **Потребность в самовыражении**, заключающаяся в самореализации и росте как личности.

Маслоу и его последователи организуют перечисленные потребности в виде возрастающей иерархии, выстраивая их тем самым их в виде пирамиды (рис. 1), где:

- уровни 1, 2 относятся к низшим,
- уровни 3, 4, 5 – к высшим потребностям.

Заметим, что, как и изображено на рис. 1, между 4 и 5 уровнями в последнее время принято выделять еще один промежуточный уровень — когнитивно-эстетических потребностей (в красоте, порядке, и т.п.)

В каждый момент времени поведение человека определяется самой сильной из неудовлетворенных потребностей. Однако потребности высших уровней не мотивируют человека, пока не удовлетворены хотя бы частично потребности низших уровней.



Рис 1 Иерархия потребностей по А. Маслоу

Менеджеры, работающие на международном уровне, должны иметь в виду, что относительная значимость различных потребностей людей может меняться в разных странах.

2.3.2. Теории мотивации

Руководителям было издавна известно, что людей следует побуждать к действиям для достижения некоторого желательного результата.

Мотивация – это процесс побуждения себя и других к деятельности для достижения некоторых целей.

В основе теории мотивации лежит понятие побуждения.

Побуждение – это порождаемое потребностью ощущение недостатка в чем-либо, имеющее определенную направленность и цель.

При достижении цели удовлетворение потребности может быть положительным (полным или частичным) и отрицательным. В первом случае человек склонен при повторении ситуации повторять данное поведение, а также искать новые подобные ситуации. Во втором случае такое поведение в будущем избегается.

Рассмотрим следующий пример. Человек имеет потребность в еде. На базе этой потребности у него возникает побуждение сходить пообедать в ресторан. Если этим актом он удовлетворяет только первичную потребность (непосредственно в насыщении), то можно ожидать повторения такого поведения. Однако современный человек социализирован, и этим актом он удовлетворяет (или нет) еще целый букет вторичных потребностей: в принадлежности («мой приятель обедает именно в этом ресторане» или «в моем социальном круге принято обедать в ресторанах, а не в бистро и не на улице» и т.д.), во власти («я обедаю в дорогом ресторане, хотя жена постоянно пилит меня по поводу нехватки денег, потому что я в семье главный»), в успехе («сходить в ресторан дорогое удовольствие, которое не все могут себе позволить, а я могу») и т.д. В этом случае предсказать поведение конкретного человека (в том числе и ему самому) становится уже очень сложно.

Этот простой пример достаточно наглядно показывает возможности и проблемы организации мотивации через потребности:

- структура потребностей у конкретного человека определяется, среди прочих факторов, местом в социальной иерархии и ранее приобретенным опытом, а эти показатели в группе людей имеют большой диапазон изменений – трудно организовать единую мотивацию для группы;
- удовлетворение одной и той же потребности у двух людей может требовать совершенно разных действий. Пример - потребность в доверии со стороны начальника может реализовываться у одного работника получением сложной работы, а у другого – предоставлением свободного режима работы;
- для организации как социального института характерны взаимозависимость работ, недостаток информации о работе каждого, текущее изменение служебных обязанностей – эти факторы дополнительно усложняют процесс мотивации.

Следующее базовое понятие теорий мотивации — вознаграждение.

Вознаграждение (поощрение) – это все, что человек считает ценным для себя.

Поощрения делятся на:

- ***внутренние*** (существующие в сознании индивида – чувство самоуважения, удовлетворенность результатом, ощущение значимости и содержательности своего труда, ощущение

ответственности труда, неформальное общение в процессе труда),

- **внешние** (то, что дает организация за выполненную работу – зарплата, премии, продвижение по службе, символы статуса и престижа, похвалы и признания, дополнительные льготы и вознаграждения).

К базовым понятиям теорий мотивации относится также **разделение факторов, определяющих мотивацию**, на:

☀ **Регуляторы мотивации.** К ним относятся:

- ♦ характеристики рабочей среды (дизайн помещения, система питания, уровень шума, чистота, физические условия работы);
- ♦ вознаграждение (оплата труда и другие выплаты, выходные дни, система медобслуживания, социальное обеспечение, дополнительные материальные выгоды);
- ♦ безопасность (принадлежность, причастность, уважение, одобрение, стиль управления, отношения с окружающими, принятый в компании стиль отношения к сотрудникам).

☀ **Главные мотиваторы.** Ими являются:

- ♦ личное развитие (ответственность, новый опыт, возможность экспериментирования, возможность обучения);
- ♦ чувство причастности (доступ к информации, совместное принятие решений, возможность представлять компанию);
- ♦ интерес и вызов (возрастающая ответственность, перспективные цели и скорость продвижения к ним).

Если факторы этих двух групп сбалансированы, то у сотрудников возникают желаемые для руководителя удовлетворенность работой и преданность организации.

Существует ряд психологических принципов, лежащих в основе теорий мотивации.

- ☉ **Гедонизм** – заинтересованность людей делать то, что им приятно и уклоняться от того, что им неприятно. Менеджер может создать приятную обстановку на работе и таким образом мотивировать подчиненных.

- ◎ **Инстинкты** – автоматическая предрасположенность людей вести себя определенным образом. Некоторые люди находят удовольствие в том, чтобы ходить на работу, поскольку "рождены работать".
- ◎ **Привычки** – формирование стереотипов поведения на основе действий, которые приводили к удовлетворяющему результату в прошлом.
- ◎ **Рациональное поведение** – поведение людей в соответствии с тем, какие они ставят перед собой цели и как видят пути их достижения. Менеджер может сформировать у работников впечатление о выгоде определенного поведения у подчиненных.

Основные теории индивидуальной мотивации

Исторически первый подход к мотивации (известен с рабовладельческих времен) имеет название **метод «кну́та и пря́ника»** – самый примитивный метод воздействия на людей с целью получения желаемого результата. Он заключался в побуждении либо под угрозой наказания, либо с использованием поощрения, либо комбинацией этих двух методов. В настоящее время он трансформировался в систему простейших экономических и административных стимулов и санкций.

Метод «кну́та и пря́ника» имеет свою экологическую нишу и в современных условиях:

- при низкой содержательности работ;
- при авторитарном стиле руководства;
- если работник не может (по каким-либо причинам) найти себе другое место работы;
- на должностях младшего обслуживающего персонала, чей доход в основном и напрямую зависит от объема выполненной работы;
- в критических для фирмы ситуациях.

Модель А. Маслоу. Модель основана на иерархии потребностей в виде пирамиды (см. рис. 1). Основная идея состоит в том, что потребности более высоких уровней удовлетворяются по мере удовлетворения потребностей более низких уровней.

Методика применения модели для организации мотивации: на базе изучения психологического портрета работника выявить его доминирующую потребность и поместить его на должность, где она будет удовлетворена с наибольшей пользой для организации.

Например, для позиции «руководитель отдела по работе с клиентами» фирмы, занимающейся поставками и обслуживанием стандартизованного программного обеспечения, лучше всего подойдет человек, у которого ведущими потребностями являются потребности в общении, причастности и власти (тогда он будет развивать у себя риторические способности, умение вести дискуссию, желание управлять другими и влиять на них, организаторские способности, нужную манеру поведения, способность к компромиссу и т.д., что и требуется на этой должности).

Недостатки модели:

- поведение человека стимулируется не одной активной потребностью, поэтому подобной структуры в чистом виде не существует (в частности, чтобы более высокий уровень потребностей начал влиять на поведение человека, не обязательно полное удовлетворение потребностей низшего уровня);
- в модели не учтены индивидуальные различия людей (в одном и том же поведенческом акте могут выражаться различные потребности – см. пример с обедом в ресторане);
- вообще говоря, не подтверждается идея об автоматическом переходе на следующий уровень потребностей при удовлетворении предыдущего.

Модель Врума выражается формулой мотивации:

$$M = a \times b \times c,$$

где a – личные ожидания по связи «затраты труда–результат»;
 b – личные ожидания по связи «результат–вознаграждение»;
 c – «валентность» (степень относительной личной удовлетворенности результатом).

Иначе говоря, работник должен быть уверен (высокий уровень ожиданий) в том, что если он будет прилагать усилия, то

- у него получится выполнить задание (величина a),
- причем за это ему адекватно заплатят (величина b),
- и это будет очень хорошо для него лично (величина c).



В. Врум

Виктор Врум (Victor H. Vroom) – современный исследователь проблем организационного поведения, психолог, преподаватель управленческих наук, управленческий консультант, доктор психологических наук, лауреат конкурса американских психологов.

Методика применения модели в организации:

- установить твердое соотношение между результатами труда и вознаграждением только за эффективную работу;
- сформировать высокий, но реалистичный уровень результатов, ожидаемых от работника;
- поднимать его самооценку, уровень полномочий и профессиональных знаний.

Недостатки модели:

- очень высок разброс оценок и ожиданий, поэтому трудно формализовать модель и, соответственно, найти инструментальные методы для ее реализации.

Теория справедливости. Содержание модели: работник субъективно определяет соотношения «усилия/вознаграждение» у себя и других работников на сходных должностях. Если он считает эти соотношения несправедливыми, у него возникает психологический дискомфорт, разрешить который можно либо повышая вознаграждение, либо снижая эффективность и качество труда.

Модель реализуется достаточно трудно реализуется:

- трудно согласовать понятие справедливости у работника и руководителя;
- у компании не всегда хватает финансовых возможностей для повышения вознаграждения;
- все потребности предлагается удовлетворять только с помощью денег, что в современном мире неактуально.

Двухфакторная («гигиеническая») теория Ф. Герцберга. Содержание модели поясняет рис. 2. Ф. Герцберг разделяет все факторы, влияющие на мотивацию работника, на две группы:

- Факторы левой группы («гигиенические») рассматриваются работником просто как факторы окружающей среды, т.е. как необходимые, но не достаточные для мотивации. Сколь угодно значительное усиление факторов этой группы

(например, увеличение зарплаты) очень быстро начинает рассматриваться работником как «естественное состояние», т.е. вызывает не более чем нейтральное отношение к работе.

- Факторы правой группы являются истинными мотиваторами. Их ослабление вызывает в худшем случае нейтральное отношение к работе, а любое усиление – удовлетворенность работой.



Рис. 2 «Гигиеническая» (двухфакторная) теория мотивации Ф. Герцберга

Недостатки модели:

- отсутствие тесной корреляции между удовлетворением и производительностью труда
- трудности объективизации модели
- наличие в организации двух групп факторов одновременно не всегда возможно.

Теория каузальной атрибуции (Келли – Маккелла).

Содержание модели:

- человек объясняет свое поведение не так, как поведение других людей;
- процессы каузальной атрибуции не подчиняются логическим нормам;
- человек склонен объяснять неудачные результаты своей деятельности внешними причинами, а удачные – внутренними.

Примеры:

- супружеская ссора – я ссорюсь с мужем, потому что у нас денежные затруднения, а он ссорится со мной, потому что у него плохой характер;
- экзамен – я сдал, потому что занимался, а он сдал, потому что повезло;
- социальная успешность: если я – представитель более высокой социальной группы, то условия жизни представителей «низшего класса» объясняю отсутствием у них таланта, предприимчивости и т.п.; если же я – представитель «низшего» класса, то «они» преуспели, потому что воруют и т.д.; интересно отметить, что при переходе из одной в другую группу взгляд человека, как правило, радикально изменяется.

Теория каузальной атрибуции хорошо объясняет поведение человека в тоталитарном обществе (секты, «работа за идею» и т.д.). Как стимулировать членов секты к активной работе? Возможны два варианта:

- хорошо платить и хвалить; в этом случае при прекращении оплаты и/или похвалы люди перестают работать;
- вообще не платить и побуждать людей работать в трудных условиях и без отдыха; в этом случае люди начинают считать, что дело, которое они делают – правое и благородное, иначе зачем бы они его выполняли; в то же время другим, которые так не работают, это понимание недоступно. Чтобы реализовать второй вариант, можно воспользоваться тактикой Тома Сойера, которому надоело красить забор.

Модель Макклеланда - Аткинсона (модель «успех – власть – причастность») является одной из основных в современных западных теориях управления персоналом.

Содержание модели: основа поведения человека – это потребности высших уровней, среди которых доминируют потребности во власти (желание воздействовать на других; находится между потребностями в уважении и самовыражении), в успехе (доведении работы до успешного конца, причем на основе личных достижений, а не общественного признания) и причастности (общении, принадлежности, помощи другим).

Методика применения модели:

- выделить людей, у которых доминирует (активизирована) одна из перечисленных потребностей;

- людей с потребностью во власти ставить на руководящие должности не ниже среднего ранга; на основе модели хорошо определяется предельный уровень власти, усиленный для данного индивида;
- людям с потребностью в успехе ставить сложные задачи и делегировать достаточно полномочий для их решения, причем гарантировать им конкретное вознаграждение по результатам труда; эти люди наиболее целеустремленны и должны составлять костяк организации;
- людям с потребностью в причастности создавать условия для неформальной коммуникации; именно они проявляются наибольшей преданностью фирме, готовы к успешной групповой работе и к тому, чтобы стать неформальным лидером.

Недостатки модели:

- модель не показывает механизм удовлетворения потребностей низшего уровня, которые не менее важны, чем высшие;
- при реализации модели возникают серьезные организационные трудности.

Концепция организационного роста Литвина - Стрингера считается наиболее современной и продвинутой. Объединяет в себе модель Макклеланда и теорию поля К. Левина.

Содержание теории поля: наблюдаемое в данный момент поведение работника есть функция поля, в котором оно проявляется. При этом поле образуется взаимозависимыми факторами – личностью и внешней средой. Поэтому для трансформации поведения работника (управления его поведением) появляются новые возможности – путем изменения внешней для работника среды, а также путем комбинированного воздействия одновременно на его личность и среду.

Преимущества модели: очевидно, что изменить обстоятельства и окружение обычно легче, чем личность. Особенно это актуально для малого бизнеса и вообще для малой группы (в большой организации изменить внешнюю среду труднее).

Содержание теории организационного роста Литвина – Стрингера хорошо описывается следующей таблицей влияния факторов психологического климата в организации на потребности работника.

Факторы психологического климата в организации	Высшие потребности работника (как рядового сотрудника, так и руководителя)		
	в успехе	во власти	в признании

Структурные ограничения	Снижают (-)	Увеличивают (+)	Снижают (-)
Ответственность	+	+	Не влияет (0)
Теплота отношений	0	0	+
Поддержка	+	0	+
Награда	+	0	+
Конфликт	+	+	0
Стандарты работы	+	0	0
Престиж	0	-	+
Риск	+	0	0

Существует еще целый ряд моделей индивидуальной мотивации, о которых можно прочесть в специальных работах по психологии. Здесь были приведены лишь наиболее употребительные.

Основные теории групповой мотивации

Индивидуальная мотивация не может быть механически перенесена на мотивацию групп (группы часто реагируют на события совсем иначе, чем отдельные люди). На структуру мотивации группы влияют:

- ◆ вид деятельности;
- ◆ нацеленность на совместную работу;
- ◆ объединяющие и разъединяющие факторы в группе;
- ◆ групповые нормы.

Приведем две основные теории групповой мотивации.

Теория Х. Эта теория вызвала к жизни административный стиль руководства и уравнилельные методы (популярные в СССР).

Ее содержание:

- ◆ люди ненавидят работу;
- ◆ чтобы заставить их работать на организацию, нужно угрожать или принуждать;
- ◆ люди любят безопасность, не амбициозны, не любят ответственности, любят, чтобы ими руководили.

Это (и то только отчасти) может быть справедливо для младшего и среднего персонала, как правило не занятого ни в какой степени творческой работой, и практически не относится к работникам с хотя бы минимальной долей творчества. Так, эта теория не может применяться для руководства командой программистов, хотя иногда и может применяться для управления, например, обслуживающим персоналом (скажем, уборщицами).

Теория У. Является базой для современного психологического управления группами персонала.

Ее содержание:

- ◆ для большинства людей работать также естественно, как и отдыхать;
- ◆ если работники интересуются общими целями (целями организации), они будут управлять собой сами, причем значительно эффективнее руководителя;
- ◆ работники заинтересуются общими целями только в той степени, в которой они увидят в реализации общих целей способ удовлетворения своих потребностей, причем высшего порядка (особенно в самосовершенствовании и самовыражении).

Выводы применительно к России:

- ◎ одна из целей и, одновременно, факторов существования организации – дать каждому сотруднику шанс расти и развиваться на своем рабочем месте через совместную работу;
- ◎ мотивировать (или стимулировать) людей только напрямую в современном мире невозможно; более эффективный путь – создание климата (поля), в котором большинство работников сами создадут себе стимулы для того, чтобы помочь организации добиться поставленных целей.
- ◎ Оплата труда является мотиватором только в случае устойчивой связи между ее размером и результатами труда, при этом оптимально разбиение зарплаты на три составляющих:
 - часть, определяемая должностью (постоянна и равна у всех людей с равными должностями);
 - часть, связанная с выслугой лет (равна у всех с одинаковым стажем работы);
 - главная часть, зависящая от результатов конкретного труда.

Начиная с определенного уровня благосостояния (или в определенных социальных ситуациях) роль денег как мотиватора уменьшается; здесь нужны нематериальные вознаграждения и льготы.

2.3.3. Особенности управления персоналом в области ИТ

Управление персоналом в области информационных технологий имеет ряд особенностей.

✓ **Узкая специализация.** Деятельность в области информационных технологий требует высокого уровня узкоспециальной подготовки сотрудников.

Менеджер проекта подбирает команду, в которой каждый из участников максимально эффективен в конкретной области. Например, при разработке сложной клиент-серверной системы, в которой присутствует СУБД Oracle, сервер приложений Oracle Application Server, на котором должна быть реализована часть бизнес-процессов, а также клиентские приложения-апплеты, в команду разработчиков должны войти по меньшей мере три типа специалистов (специалисты по серверному программированию для Oracle, специалисты по Oracle Application Server и технологии EJB, а также специалисты по реализации GUI). При этом никто из них не будет видеть проблему в целом. В результате нагрузка по оптимизации и интеграции системы в значительной мере переносится на системного архитектора, который, в свою очередь, не может быть специалистом во всех областях сразу (скорее всего, он не будет знать подробно ни одну из соответствующих областей). Проблема сбалансированного подбора команды специалистов ложится менеджера проекта.

✓ **Постоянное обучение.** Все информационные технологии, а тем более технологии разработки программного обеспечения весьма быстро морально устаревают, а качественно новые версии инструментальных средств (IDE, компиляторов, СУБД и т.п.) выходят иногда с периодичностью раз в полгода. В связи с этим необходимо постоянное повышение квалификации сотрудников. Менеджер конкретного проекта обязан предусмотреть отвлечение ресурсов на обучение.

✓ **Высокое самомнение.** Узкая специализация и высокая профессиональная квалификация в конкретной области информационных технологий часто вызывают завышенную самооценку своих возможностей у сотрудников. Программистам кажется, что они могут все. В связи с этим возникают две проблемы менеджера проекта. Во-первых, менеджер должен сам точно представлять реальные возможности своих сотрудников, в противном случае неприятные неожиданности неизбежны. Во-вторых, сотруднику с высокой самооценкой трудно что-либо приказать, его необходимо убедить, что бывает непросто (в силу того, что сам менеджер вряд ли может быть авторитетом в той области, в которой сотрудник является узким специалистом).

✓ **Недисциплинированность.** Опыт показывает, что специалистов в области информационных технологий, а особенно программистов, часто трудно заставить приходить на работу вовремя, не опаздывать на совещания, своевременно (и по нужным адресам) посылать отчеты. Это связано, возможно, с кажущимся индивидуальным характером труда. Ставший классическим образ «программера», работающего по ночам появился с распространением персональных компьютеров, в эпоху мэйнфреймов и даже мини-ЭВМ, такого стереотипа не было. Однако сейчас он есть, а значит, менеджеру приходится выполнять дополнительную работу по доведению до сознания сотрудников того факта, что они работают в команде, и разработка программного обеспечения – всегда коллективная деятельность.

✓ **Творчество.** Разработка программного обеспечения – творческий процесс, даже в промышленных условиях. Разработчики являются креативными личностями, и способны привносить энтузиазм, инициативу и собственные нетривиальные решения в общее дело. При наличии сильной мотивации и ясной цели они, как правило, готовы работать с огромной самоотдачей. Это значит, что управление персоналом в программных проектах следует организовывать *по целям*, а не *по заданиям*.

✓ **Мобильность сотрудников.** В современных условиях спрос на квалифицированных программистов и, шире, вообще специалистов в информационных технологиях, существенно превышает предложение. Эта тенденция сохранится в обозримом будущем. Из этого следует, что менеджер проекта должен быть в принципе готов к внезапному уходу из команды (и из компании) любого из сотрудников. Процесс следует организовать так, чтобы подобный уход не вызвал катастрофических последствий для проекта. Здесь необходимо учесть два аспекта: возможность утраты необходимой рабочей силы и возможность безвозвратной потери know-how (или даже важных внутренних документов). Программисты часто не понимают, что программы, разработанные в рамках проекта организации, им не принадлежат.

2.3.4. Корпоративная культура

Корпоративная культура – это способ и средство создания самоподдерживающейся, саморазвивающейся организации.

Для формирования корпоративной культуры в организации важно сформулировать

- цель компании,
- миссию компании,
- корпоративные стандарты.

Цель компании

Цель компании – это основополагающий элемент корпоративной культуры. Он определяет стратегическое видение компанией своего места на рынке и задает общий вектор развития.

Цель компании должна быть значительной! Проверка на значительность проста. Если после оглашения цели перед всем коллективом сотрудников компании можно увидеть горящие глаза, гордо поднятые подбородки, выпрямленные спины, то это то, что надо.

Чтобы создать «значительную» цель, необходимо выполнить следующие условия.

1. Достижение цели сделает компанию и людей, в ней работающих, абсолютно уникальными. Этому правилу, например, отвечает цель «стать абсолютным лидером в области программирования Web-магазинов».
2. Цель достижима «в принципе». Например, цель «стать абсолютным лидером в области разработки программного обеспечения» в принципе недостижима.
3. Цель нравится ВСЕМ сотрудникам, они считают её достойной. Например, цель «стать одним из трех основных производителей контрафактных DVD» будет наверняка противоречить моральным ценностям некоторых сотрудников. Цель «повысить объем продаж на 7,5%» (даже если на самом деле это означает стать абсолютным лидером в определенном секторе рынка) вряд ли вызовет энтузиазм у всех сотрудников (кроме, возможно, сотрудников отдела продаж, заработок которых непосредственно зависит от уровня продаж).

Очень важно не путать цель компании с конкретными целями, появляющимися на этапе планирования проекта. Последние должны иметь чёткие сроки и численные показатели, заданные в явном виде. Главная задача цели компании – наполнить деятельность сотрудников энергией, воздействовать на их эмоции. Как раз с этой точки зрения цифры в формулировке цели компании скорее вредны.

Миссия компании

Миссия компании – это формулировка того, чем полезна компания с точки зрения общества.

Миссию компании не следует путать с целью. От цели компании миссия отличается направленностью: цель – это то, что мы хотим для себя, а миссия – это то, почему наша деятельность будет благом и для всех остальных. Конечно, миссия – это самореклама, направленная и вовне, и вовнутрь. Но, будучи явно сформулированной, миссия становится частью корпоративной культуры, начинает работать и исполняться. Обмана, пренебрежения исполнением миссии, руководству компании не простит никто, ни внешнее окружение, ни сами сотрудники.

Самая распространённая ошибка, которую совершают, формулируя миссию – это подмена внешнего общества (партнёры, клиенты, государство и т.д.) внутренним (сотрудники). Сотрудники должны быть мотивированы целью компании, то есть приближение исполнения цели компании должно способствовать приближению исполнения целей каждого конкретного сотрудника. Миссия – инструмент для внешнего использования. Однако сотрудники компании вынуждены под нее подстраиваться.

Формулировка миссии должна быть ёмкой, чёткой и краткой. Не следует употреблять такие общие (и слабо осмысленные) слова, как «клиентоориентированность», «социальное партнерство», «максимум выгод по минимальной цене». Не стоит также использовать такие пустые фразы, как: «мы стремимся к взаимовыгодному сотрудничеству для развития и улучшения бизнеса», «мы дорожим своими клиентами».

А вот пример разумно сформулированной миссии: *«Наша компания разрабатывает простые инструменты автоматизации документооборота, обеспечивающие клиентам наименьшую совокупную стоимость владения».*

Корпоративные стандарты

Корпоративные стандарты – это кодифицированный набор правил поведения сотрудников компании в различных ситуациях.

Важно, чтобы составляющие корпоративные стандарты правила были кодифицированы, т.е. сформулированы в обязательных для сотрудников компании внутренних документах и сведены в единый свод, с которым каждый из сотрудников должен быть ознакомлен при приеме на работу.

Правила, составляющие корпоративные стандарты, различны для различных процессов.

Приведем ряд наглядных примеров.

1) Для сотрудников, непосредственно общающихся с клиентами, целесообразно сформулировать стандарты коммуникации. Скажем, при первом звонке клиента необходимо снять трубку после второго гудка, сказать приветствие, представить компанию и себя (должность, имя), выразить готовность слушать. В первой трети разговора, после того как сотрудник ответил на один-два вопроса, нужно познакомиться с собеседником – после чего обращаться к нему только по имени или имени-отчеству. В контексте разговора следует делать позитивные высказывания («спасибо, что позвонили», «какой интересный вопрос», «мне приятна ваша осведомленность»), а также при каждой возможности подчеркивать общность взглядов и мнений с клиентом («я согласен с вами», «мы оба так считаем»). Стандарт устанавливает и максимальное время, которое можно потратить на каждом этапе общения с потребителем. На первый звонок отводится не более пяти минут. Первая встреча не должна выходить за рамки 30 минут. Выезд к заказчику не может превышать двух часов.

Вот еще один важный пример.

2) Одним из важных корпоративных стандартов в области создания программного обеспечения является стандарт кодирования. В нем целесообразно перечислить основные обязательные для кодировщиков требования к стилю программирования. Уровень этих требований оказывает значительное влияние на качество работы и, тем самым, на уровень корпоративной культуры. Например, регламентация в стандарте кодирования количество пробелов в отступах является низкоуровневым требованием, а правило выбора и применения конкретных образцов проектирования (design patterns) является высокоуровневым. Не следует злоупотреблять большим количеством низкоуровневых требований (они могут сковывать инициативу сотрудников), но нельзя и допускать «беспринципного программирования».

2.4. Заключение

Психология в менеджменте, особенно в менеджменте проектов по разработке программного продукта имеет ряд специальных черт, рассмотренных в этой теме.

2.5. «Карта памяти» по теме 2



2.6. Список использованной и дополнительной литературы

1. Психология менеджмента: учебник. - СПб.: Изд-во Санкт-Петербургского университета, 2000.
2. Светлана Иванова. Мотивация на 100% : А где же у него кнопка? (2-е издание) Серия "бизнес на 100%", Издательство Альпина Бизнес Букс, 2005.
3. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. - СПб.: Символ-Плюс, 1999.
4. Гусарова Н.Ф. Общая психология / Уч. пособие. СПб: СПбГИТМО (ТУ), 2001.
5. Гусарова Н.Ф. Координация в технологических процессах со слабо формализуемыми критериями. Монография. СПб: СПбГИТМО(ТУ), 2001.

Тема 3. Основы принятия управленческих решений

3.1. Введение

Управленческое решение – это принятие решения, то есть выбор альтернативы, в процессе управления.

В этой теме рассматриваются основные методы принятия управленческих решений применительно к разработке программных продуктов.

Изучив учебный материал данной темы, Вы:

- узнаете или пополните свои знания о видах управленческих решений;
- узнаете или пополните свои знания о теории принятия решений;
- научитесь применять формальных методов принятия решений в условиях риска и неопределенности.

В рамках темы рассматриваются следующие учебные вопросы:

- Основные понятия теории принятия решений
- Виды управленческих решений
- Методы принятия управленческих решений

3.2. Понятие и виды управленческих решений

Принятие решений, так же как и обмен информацией – составная часть любой управленческой функции. Необходимость принятия решений возникает на всех этапах процесса управления, связана со всеми участками и аспектами управленческой деятельности и является её квинтэссенцией. Поэтому так важно понять природу и сущность решений. В самом общем случае решение – это выбор одной из возможных альтернатив.

Решение – это выбор альтернативы.

3.2.1. Управленческие решения

Управленческие решения отличаются от решений, принимаемых в повседневной жизни, следующими особенностями:

- **Цели.** В повседневной жизни мы принимаем решения, исходя из собственных потребностей и предпочтений. Управленческое решение принимается исходя не из личных потребностей менеджера, а из целей конкретной организации.
- **Последствия.** Частный выбор индивида сказывается на его собственной жизни и может повлиять на немногих близких ему людей. Менеджер, особенно высокого ранга, выбирает направление действий не только для себя, но и для организации в целом и её работников, и его решения могут существенно повлиять на жизнь многих людей.
- **Исполнение.** Если в частной жизни человек, принимая решение, как правило, сам его и выполняет, то в организации существует определённое разделение труда: одни работники (менеджеры) заняты решением возникающих проблем и принятием решений, а другие (исполнители) – реализацией уже принятых решений.
- **Полномочия.** В частной жизни каждый человек самостоятельно принимает решения в силу своего интеллекта и опыта. В управлении организацией принятие решений – гораздо более сложный, ответственный и формализованный процесс, требующий профессиональной подготовки. Далеко не каждый сотрудник организации, а только обладающий профессиональными знаниями и навыками наделяется полномочиями самостоятельно принимать определённые решения.

Управленческое решение – это выбор альтернативы, осуществлённый руководителем в рамках его должностных полномочий и компетенции и направленный на достижение целей организации.

3.2.2. Виды управленческих решений

В процессе управления менеджерами принимается огромное количество самых разнообразных решений, обладающих различными характеристиками. Тем не менее, существуют некоторые общие признаки, позволяющие это множество определенным образом классифицировать. Такая классификация представлена в таблице 3.1.

Таблица 3.1

Виды управленческих решений

Классификационный признак	Группы управленческих решений	
Степень повторяемости проблемы	Традиционные	Нетипичные
Значимость цели	Стратегические	Тактические
Сфера воздействия	Глобальные	Локальные
Длительность реализации	Долгосрочные	Краткосрочные
Прогнозируемые последствия	Корректируемые	Необратимые
Метод разработки решения	Формализованные	Неформализованные
Количество критериев выбора	Однокритериальные	Многокритериальные
Форма принятия	Единоличные	Коллегиальные
Способ фиксации решения	Документированные	Недокументированные
Характер использованной информации	Детерминированные	Вероятностные

Рассмотрим эти признаки подробнее.

- **Степень повторяемости проблемы.** В зависимости от повторяемости проблемы, требующей решения, все управленческие решения можно подразделить на *традиционные*, неоднократно встречавшиеся в практике управления, когда необходимо лишь сделать выбор из уже имеющихся альтернатив, и *нетипичные*, нестандартные решения, когда их поиск связан, прежде всего, с генерацией новых альтернатив.
- **Значимость цели.** Принятие решения может преследовать собственную, самостоятельную цель или же быть средством – способствовать достижению цели более высокого порядка. В соответствии с этим решения могут быть *тактическими* или *стратегическими*.
- **Сфера воздействия.** Результат решения может сказаться на каком-либо одном исполнителе или подразделении организации. В этом случае решение можно считать *локальным*. Решение, однако, может приниматься и с целью повлиять на работу организации в целом, в этом случае оно будет *глобальным*.
- **Длительность реализации.** Реализация решения может потребовать нескольких минут, часов, дней, месяцев или лет. Если между принятием решения и завершением его реализации пройдет сравнительно короткий срок (в масштабе общей продолжительности

проекта) – решение *краткосрочное*. В то же время, все более возрастает количество и значение *долгосрочных* перспективных решений, результаты осуществления которых могут сказаться только через несколько лет.

- **Прогнозируемые последствия решения.** Большинство управленческих решений в процессе их реализации, так или иначе, поддается корректировке с целью устранения каких-либо отклонений или учёта новых факторов, т.е. является *корректируемыми*. Вместе с тем имеются и решения, последствия которых *необратимы*.
- **Метод разработки решения.** Некоторые решения, как правило, типичные, повторяющиеся, могут быть с успехом *формализованы*, т.е. приниматься по заранее определённом алгоритму. Формализация принятия решений повышает эффективность управления в результате снижения вероятности ошибки и экономии времени: не нужно заново разрабатывать решение каждый раз, когда возникает соответствующая ситуация. Поэтому руководство организаций часто формализует решения для определённых, регулярно повторяющихся ситуаций, разрабатывая соответствующие корпоративные стандарты. В то же время в процессе управления организациями часто встречаются новые, нетипичные ситуации и нестандартные проблемы, которые не поддаются формализованному решению. В таких случаях большую роль играют интеллектуальные способности, талант и личная инициатива менеджеров, способных принять *неформализованное* решение.
- **Количество критериев выбора.** Если выбор наилучшей альтернативы производится только по одному критерию (что характерно для формализованных решений), то принимаемое решение будет простым, *однокритериальным*. И наоборот, когда выбранная альтернатива должна удовлетворять одновременно нескольким критериям, решение будет сложным, *многокритериальным*. В практике менеджмента подавляющее большинство решений являются многокритериальными, так как они должны одновременно отвечать таким критериям, как: объем прибыли, доходность, уровень качества, доля рынка, уровень занятости, срок реализации и т.п.
- **Форма принятия решений.** Лицом, принимающим решение (ЛПР), может быть один человек и его решение будет соответственно *единоличным*. Однако в современной практике менеджмента всё чаще встречаются сложные ситуации и проблемы, решение которых требует всестороннего, комплексного анализа, т.е. участия группы менеджеров и специалистов. Такие групповые, или коллективные, решения называются *коллегиальными*. Усиление профессионализации и углубление специализации управления приводят к широкому

распространению коллегиальных форм принятия решений. Коллегиальная форма принятия решений, разумеется, снижает оперативность управления и «размывает» ответственность за его результаты, однако, препятствует грубым ошибкам и злоупотреблениям и повышает обоснованность выбора.

- **Способ фиксации решения.** По этому признаку управленческие решения могут быть разделены на *документированные* (т.е. оформленные в виде какого-либо документа – приказа, распоряжения, письма и т.п.) и *недокументированные* (не имеющие документальной формы, устные). Большинство решений в аппарате управления оформляется документально, однако, мелкие, несущественные решения, а также решения, принятые в чрезвычайных, острых, не терпящих промедления ситуациях, могут и не фиксироваться документально.
- **Характер использованной информации.** В зависимости от степени полноты и достоверности информации, которой располагает менеджер, управленческие решения могут быть *детерминированными* (принятыми в условиях определённости) или *вероятностными* (принятыми в условиях риска или неопределённости). Эти условия играют чрезвычайно важную роль при принятии решений, поэтому мы их рассмотрим более подробно.

3.3. Процесс принятия управленческих решений

Процесс и методы принятия управленческих решений являются предметом *теории принятия решений*. В этом разделе вводятся основные понятия этой теории.

3.3.1. Лицо, принимающее решение

В теории принятия решений есть специальный термин – *«Лицо, Принимающее Решения», сокращенно ЛПР*. Это тот, на ком лежит ответственность за принятое решение, тот, кто подписывает приказ или иной документ, в котором выражено решение. Обычно это генеральный директор или председатель правления фирмы, командир воинской части, мэр города и т.п., словом – ответственный работник. В случае проектов по разработке программного обеспечения ЛПР – это, как правило, менеджер проекта.

Но иногда может быть и коллегиальное ЛПР, как, например, в случае с Государственной Думой Российской Федерации. В больших проектах по разработке сложного программного обеспечения наиболее важные, стратегические решения принимает, как правило, коллективное ЛПР, которое называется *Технический совет*, иногда называемый также

Группой проекта (Project Board). В такой совет включают менеджера проекта, ведущих технических специалистов проекта и полномочных представителей всех заинтересованных сторон проекта. Ясно, что затраты на принятие коллективного решения существенно больше, чем на принятие единоличного, но эти затраты окупаются снижением риска принятия неверного стратегического решения на ранних стадиях проекта.

Группа проекта – это коллегиальное ЛПР, состоящее из полномочных представителей всех заинтересованных сторон проекта.

Проект решения готовят специалисты, как говорят, **«аппарат ЛПР»**. Если ЛПР доверяет своим помощникам, то может даже не читать текст решения, а просто подписать его. Но ответственность все равно лежит на ЛПР, а не на тех, кто участвовал в подготовке решения.

При практической работе на ЛПР лежит обязанность четко отделять этап дискуссий, когда рассматриваются различные варианты решения, от этапа реализации решения, когда надо принятое решение выполнять, а не обсуждать. Границей между этими этапами является момент принятия решения.

Часты конфликты между менеджерами по поводу сфер ответственности – кто за что отвечает, кто какие решения принимает. Поэтому очень важны регламенты, определяющие порядок работы. Недаром любое собрание принято начинать с утверждения председательствующего и повестки заседания, а работу любого предприятия или общественного объединения – с утверждения его устава. Это же относится и к управлению проектами по разработке программного обеспечения. Еще до начала работы по проекту необходимо четко определить, кто принимает решения по архитектуре, кто отвечает за тестирование, за управление конфигурацией и вообще, кто принимает решения и отвечает за последствия (т.е. является ЛПР) по каждому из видов деятельности, предусмотренной используемым процессом разработки.

3.3.2. Цели и ресурсы

Каждое решение направлено на достижение одной или нескольких целей. Иногда все цели можно достичь одновременно. Однако так бывает не всегда.

Часто встречающаяся формулировка «максимум прибыли при минимуме затрат» внутренне противоречива. Минимум затрат равен нулю, когда работа не проводится, но и прибыль тогда тоже равна нулю. Если же прибыль велика, то и затраты велики, поскольку и то, и другое связано с объемом производства. Можно либо максимизировать

прибыль при фиксированных затратах, либо минимизировать затраты при заданной прибыли.

Каждое решение при своей реализации предполагает использование тех или иных ресурсов. Так, принимая решение о выборе технологии для разработки программного продукта, ЛПР исходит из существования коллектива разработчиков, владеющих альтернативными технологиями и способных провести проект. Если бы коллектива не было, то и дискуссия не имела бы смысла. Конечно, можно было бы сначала обсудить вопрос о единовременном найме нужного числа разработчиков, о посильности таких затрат для организации... Кроме того, предполагается, что у организации достаточно средств для проведения начальных стадий разработки – ведь надо сначала создать продукт, и только потом получить прибыль как разность между доходами от продаж и расходами на разработку.

При подготовке решения менеджеру важно все время задавать себе вопросы: «Чего мы хотим достичь? Какие ресурсы мы готовы использовать для этого?».

3.3.3. Риски и неопределенности

Многие решения принимаются в условиях риска, т.е. при возможной опасности потерь. Связано это с разнообразными неопределенностями, окружающими нас. Кроме отрицательных неожиданностей бывают положительные – мы называем их удачами. Менеджеры стараются застраховаться от потерь и не пропустить удачу.

Внутренне противоречива формулировка: "Максимум прибыли и минимум риска". Обычно при возрастании прибыли возрастает и риск – возможность все потерять. Наиболее прибыльными в нашей стране были финансовые пирамиды типа МММ – для тех, кто успел вовремя продать «билеты МММ», «наварив» на них тысячи процентов прибыли. Подавляющее же большинство потеряло свои деньги, оставшись с «бесценными» (не имеющими цены) бумажками в руках.

Неопределенность порождает не только факторы, значения которых мы не знаем заранее и не можем на них повлиять (таковы, например, погодные условия). Неопределенности содержатся во всех данных, используемых для принятия решений. Например, данные о предполагаемой прибыли от разработки программного продукта не могут быть точными. Ведь для того, чтобы рассчитать прибыль, необходимо:

- оценить затраты на разработку (это можно сделать достаточно точно, только если есть надежные данные для достаточно большого числа аналогичных проектов);
- оценить число будущих продаж разрабатываемого продукта (а это уже труднее – если продукт новый, то ранее он не

продавался, и достоверной статистики нет). В таком случае можно, например, поручить социологам провести опрос потенциальных потребителей, но результаты любого опроса заведомо содержат значительные погрешности;

- оценить затраты, которые готов понести средний потенциальный потребитель продукта за право владения продуктом (а это зависит, в частности, от общего экономического положения к моменту выхода продукта на рынок, которое тем самым тоже необходимо спрогнозировать).

Следовательно, реальные значения прибыли будут иметь отклонения от прогнозируемых значений, которые используются при принятии решения. Необходимо изучить устойчивость выводов по отношению к допустимым отклонениям исходных данных, а также по отношению к малым изменениям предпосылок используемой математической модели.

Речь идет об общеинженерной идее – любое измерение проводится с некоторой погрешностью, и эту погрешность необходимо учитывать.

3.3.4. Системный подход при принятии решений

При обсуждении проблем принятия решений часто говорят о *системном подходе, системе, системном анализе*. Речь идет просто о том, что надо рассматривать проблему в целом, а не «выдергивать» для обсуждения какую-нибудь одну черту, хотя и важную. Так, при массовом жилищном строительстве можно «выдернуть» черту – стоимость квадратного метра в доме. Тогда наиболее дешевые дома – пятиэтажки. Если же взглянуть системно: учесть стоимость транспортных и инженерных коммуникаций (подводящих электроэнергию, воду, тепло и др.), то оптимальное решение может быть уже другим (в зависимости от выбранного критерия), например, девятиэтажные дома.

Так, например, менеджер компании по продаже программного обеспечения, отвечающий за распространение программного продукта, рассчитанного на автоматизацию бухгалтерского учета, может сосредоточиться на рекламе в средствах массовой информации. Между тем ему от системы «компания – пользователи» может оказаться лучше перейти к системе «компания – руководители организаций – пользователи». Договоренность с руководителями предприятий, давшим в итоге приказ бухгалтерам перейти на использование данного программного продукта, может дать менеджеру гораздо больший прирост численности пользователей, чем постоянная дорогая реклама.

Различных определений понятия системы весьма много. Общим в них является то, что о системе говорят как о множестве, между элементами которого имеются связи. Целостность системы и ее «отделенность» от окружающего мира обеспечиваются тем, что взаимосвязи внутри системы существенно сильнее, чем связь какого-либо ее элемента с любым элементом, лежащим вне системы.

По определению действительного члена Российской академии наук Н.Н.Моисеева:

Системный анализ – это дисциплина, занимающаяся проблемами принятия решений в условиях, когда выбор альтернативы требует анализа сложной информации различной физической природы.

Именно в таких условиях и приходится принимать решения менеджерам.

3.4. Методы принятия управленческих решений

При принятии решений применяют богатый арсенал методов современной прикладной математики. Они используются для оценки ситуации и прогнозирования при выборе целей, для генерирования множества возможных вариантов решений и выбора из них наилучшего.

Прежде всего, надо назвать **методы оптимизации** (математического программирования). Для борьбы с многокритериальностью используют различные методы свертки критериев, а также интерактивные компьютерные системы, позволяющие вырабатывать решение в процессе диалога человека и компьютера. Применяют имитационное моделирование, базирующееся на компьютерных системах, отвечающих на вопрос: "Что будет, если...?", метод статистических испытаний (Монте-Карло), модели надежности и массового обслуживания. Часто необходимы статистические (эконометрические) методы, в частности, методы выборочных обследований. При принятии решений применяют как вероятностно-статистические модели, так и методы анализа данных.

Особого внимания заслуживают проблемы неопределенности и риска, связанных как с природой, так и с поведением людей. Разработаны различные способы описания неопределенностей: вероятностные модели, теория нечеткости, интервальная математика. Для описания конфликтов (конкуренции) полезна теория игр. Для структуризации рисков используют деревья причин и последствий. Плата за риск и различные формы страхования также постоянно должны быть в его поле зрения.

3.4.1. Игры с природой

Типовой ситуацией в процессе принятия решений при разработке программного продукта является наличие неопределенности и риска. В этих условиях для принятия решений оказываются полезны некоторые методы теории игр, которые мы рассмотрим на примере.

Пусть менеджеру проекта по разработке программного продукта необходимо принять решение о выборе архитектуры разрабатываемого продукта.

Имеются две альтернативы:

1. Можно выбрать простую архитектуру клиент/сервер, причем известно, что в этом случае стоимость разработки составит 50 тыс. руб.
2. Можно выбрать более сложную трехзвенную архитектуру, и получить продукт с большими возможностями, но в этом случае стоимость разработки составит 150 тыс. руб.

На величину прибыли существенно влияет число продаж продукта. По предыдущему опыту известно, что число продаж может быть малым (менее 10), средним (от 10 до 20) или большим (более 20). В качестве представителей этих интервалов выберем средние значения: 5, 15 и 25, соответственно. При этом ценовая политика фирмы такова, что:

- * при малом числе продаж любой продукт продается по минимальной цене в 15 тыс. руб.;
- * при среднем числе продаж простой продукт можно продавать по 20 тыс. руб., а сложный – по 30 тыс. руб.;
- * при большом объеме продаж простой продукт продается по той же цене 20 тыс. руб., а сложный продукт можно продавать дороже – 40 тыс. руб.

Проведем простые расчеты ожидаемой прибыли для этих вариантов и сведем их в таблицу 3.2.

Таблица 3.2

Ожидаемая величина прибыли, тыс. руб.

Архитектура	Число продаж		
	Малое, 5	Среднее, 15	Большое, 25
Простая	$5 \times 15 - 50 = 25$	$15 \times 20 - 50 = 250$	$25 \times 20 - 50 = 450$
Сложная	$5 \times 15 - 150 = -75$	$15 \times 30 - 150 = 300$	$25 \times 40 - 150 = 850$

Ясно, что если бы число продаж было известно, то проблемы выбора не было бы. Но проблема есть, и состоит она в том, что число продаж неизвестно заранее. В данном случае рынок выступает в качестве «природы» – фактора, который решающим образом влияет на наши решения, причем на выбор значения этого фактора мы не можем повлиять. Такая ситуация в теории игр называется *игрой с природой* – мы делаем наш выбор, а природа делает свой и в результате мы проигрываем или выигрываем.

Матрица, приведенная в таблице 3.2, называется *платежной матрицей (payoff matrix) игры*. Традиционно индексы столбцов обозначают j , индексы строк – i , а элементы матрицы, которые соответствуют выигрышу при выборе нами варианта i в состоянии природы j , обозначаются a_{ij} .

Кроме платежной матрицы a_{ij} , рассматривается *матрица рисков (risk matrix) r_{ij}* . Риск определяется как разность между выигрышем, который мы бы получили, если бы точно знали состояние природы и выигрышем, который мы получим, не зная этой информации. Другими словами, $r_{ij} = \max_i a_{ij} - a_{ij}$. Для рассматриваемого случая матрица рисков приведена в таблице 3.3.

Таблица 1.

Матрица рисков, тыс. руб.

Архитектура	Число продаж		
	Малое, 5	Среднее, 15	Большое, 25
Простая	$25 - 25 = 0$	$300 - 250 = 50$	$850 - 450 = 400$
Сложная	$25 - (-75) = 100$	$300 - 300 = 0$	$850 - 850 = 0$

В игре с природой можно применить один из следующих критериев выбора решения:

- **Оптимистический критерий максимакса.** Максимизируются максимальные выигрыши в каждом состоянии природы: $\max_i \max_j a_{ij}$. В данном случае этот критерий дает ответ: $i=2$, $j=3$, архитектура сложная, ожидаемая прибыль 850 тыс. руб.
- **Пессимистический критерий максимина Вальда (Wald).** Природа рассматривается как враждебная среда и максимизируется выигрыш в наихудшем случае, т.е. $\max_i \min_j a_{ij}$. В данном случае этот критерий дает ответ: $i=1$, $j=1$, архитектура простая, ожидаемая прибыль 25 тыс. руб.

- **Критерий минимаксного риска Сэвиджа (Savage).** Критерий аналогичен предыдущему, но рассматривается не матрица выигрышей, а матрица рисков. Минимизируется проигрыш в наихудшем случае, т.е. $\min_i \max_j r_{ij}$. В данном случае этот критерий дает ответ: $i=1, j=2$, архитектура простая, ожидаемая прибыль 250 тыс. руб.
- **Смешанный критерий пессимизма – оптимизма Гурвица (Hurwitz).** Критерий рекомендует занимать некоторую промежуточную позицию между крайним пессимизмом и безудержным оптимизмом. При этом задается некоторый коэффициент пессимизма p и решение выбирается в соответствии со значением $\max_i \left(p \min_j a_{ij} + (1-p) \max_j a_{ij} \right)$. Значение p лежит в отрезке $[0,1]$. При $p=0$ критерий Гурвица совпадает с максимаксным критерием, а при $p=1$ – с критерием Вальда. Пусть $p=0,5$. Тогда имеем:
 $0,5 \times 25 + 0,5 \times 450 = 237,5$,
 $0,5 \times (-75) + 0,5 \times 850 = 387,5$, и этот критерий дает ответ: $i=2$, архитектура сложная, ожидаемая прибыль 387,5 тыс. руб.

Как видно, из четырех приведенных критериев два рекомендуют первый вариант, а два других – второй. Выбор критерия – субъективное дело менеджера и выбор все равно остается за ним. Однако сами введенные структуры – платежная матрица и матрица рисков делают этот выбор более осознанным и упорядоченным.

Мы видим, что в этом случае теория не дает однозначного математически обоснованного ответа. Дело здесь не в слабости теории, а в неопределенности самой ситуации. Неопределенность поведения природы называется «дурной», если вообще ничего нельзя сказать о ее состоянии. Как правило, в таких случаях для принятия обоснованных решений необходимо привлечь дополнительную информацию. Например, может быть известна вероятность пребывания природы в каждом из возможных состояний. В таком случае неопределенность называют *стохастической*.

Если заданы вероятности p_i пребывания природы в каждом из состояний, то наилучшим решением будет то, которое обеспечивает максимальный средний выигрыш: $\max_i \sum_j p_j a_{ij}$, если же рассматривать матрицу рисков, то наилучшим решением будет то, которое обеспечивает минимальный средний риск: $\min_i \sum_j p_j r_{ij}$. Нетрудно видеть, что оба критерия дают одно и то же решение. Действительно:

$$\min_i \sum_j p_j r_{ij} = \min_i \sum_j p_j (\max_i a_{ij} - a_{ij}) = \min_i \left(\sum_j p_j \max_i a_{ij} \right) - \min_i \sum_j p_j a_{ij} = C + \max_i \sum_j p_j a_{ij},$$

где C – константа.

Таким образом можно рассматривать любую матрицу. На практике обычно выбирают ту матрицу, которая задана с большей достоверностью.

Продолжим рассмотрение примера.

Пусть отдел маркетинга указал вероятности состояния рынка:

- * малое число продаж – 0,4;
- * среднее – 0,4;
- * большое – 0,2.

Тогда имеем:

$$0,4 \times 25 + 0,4 \times 250 + 0,2 \times 450 = 10 + 100 + 225 = 335 \text{ тыс. руб.},$$

$$0,4 \times (-75) + 0,4 \times 300 + 0,2 \times 850 = -30 + 120 + 425 = 515 \text{ тыс. руб.},$$

и критерий дает ответ $i=2$, архитектура сложная, средняя ожидаемая прибыль 515 тыс. руб.

Следует заметить, что использование средних значений носит в данном случае условный характер. Использование средних правомерно и оправдано при большом числе испытаний, здесь же речь идет об одном-единственном решении.

3.4.2. Деревья принятия решений

В практических ситуациях решения часто оказываются многошаговыми. В таких случаях целесообразно выбирать решение с помощью *дерева принятия решений*.

Рассмотрим этот метод на примере.

Пусть в условиях предыдущей задачи имеется возможность заказать консультационной компании исследование рынка с целью более точного предсказания числа продаж. Услуги консультационной компании стоят 50 тыс. руб. Эта компания давно работает на рынке и по предыдущему опыту известно, что точный прогноз будет дан с вероятностью 0,6 и с вероятностью 0,2 консультант ошибется в ту или другую сторону.

Возможности консультанта представлены в таблице 3.4.

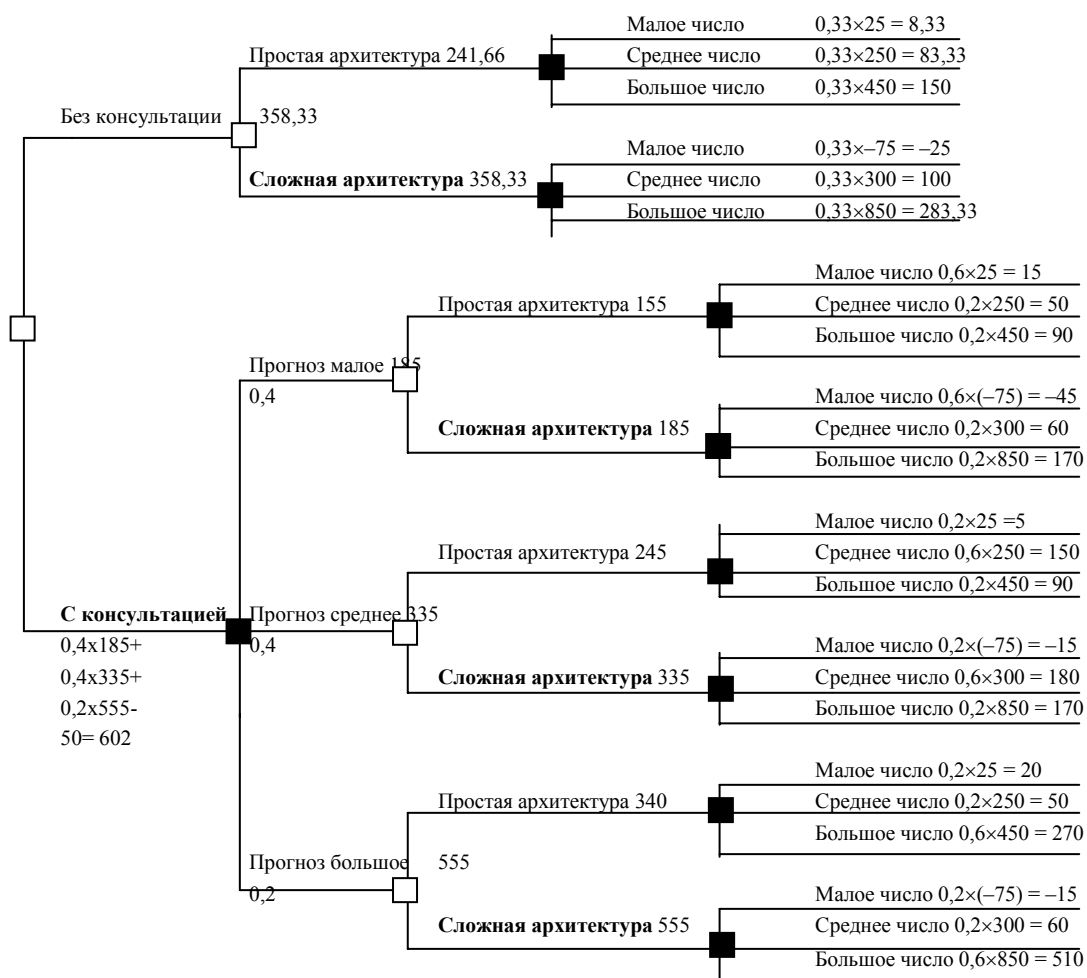
Таблица 3.4

Условные вероятности прогноза

Предсказанное число продаж	Фактическое число продаж		
	Малое, 5	Среднее, 15	Большое, 25
Малое, 5	0,6	0,2	0,2
Среднее, 15	0,2	0,6	0,2
Большое, 25	0,2	0,2	0,6

Допустим, что априори мы оцениваем вероятности того, что консультационная фирма выдаст прогноз малого, среднего и большого числа продаж 0,4, 0,4, 0,2, соответственно.

Результаты возможных решений можно теперь представить в виде следующей древовидной схемы. На этой схеме пустыми квадратами обозначены решения, принимаемые ЛПР, а закрашенными квадратами – «решения природы», на которые ЛПР не может влиять. Именно по этой причине консультационную фирму следует рассматривать как «природу».



Анализ дерева решений показывает, что в данном случае затраты на получение консультации оказываются оправданными, поскольку улучшают качество решения.

3.4.3. Методы экспертных оценок

Совершенно бесспорно, что для принятия обоснованных решений необходимо опираться на опыт, знания и интуицию специалистов. После второй мировой войны в рамках теории управления (менеджмента) стала развиваться самостоятельная дисциплина – **экспертные оценки**.

Методы экспертных оценок – это методы организации работы со специалистами-экспертами и обработки мнений экспертов, выраженных в количественной и/или качественной форме с целью подготовки информации для принятия решений ЛПР.

Для проведения работы по методу экспертных оценок создают (формально или по существу) **экспертную комиссию (ЭК)**. Существует множество методов получения экспертных оценок. В одних с каждым экспертом работают отдельно, он даже не знает, кто еще является экспертом, а потому высказывает свое мнение независимо от авторитетов. В других экспертов собирают вместе для подготовки материалов для ЛПР, при этом эксперты обсуждают проблему друг с другом, учатся друг у друга, и неверные мнения отбрасываются. Существуют методы, в которых число экспертов фиксировано и таково, чтобы статистические методы проверки согласованности мнений и затем их усреднения позволяли принимать обоснованные решения. В других – число экспертов растет в процессе проведения экспертизы.

Прежде всего, необходимо решить, что должна представить экспертная комиссия в результате своей работы – информацию для принятия решения ЛПР или проект самого решения? Рассмотрим каждый из вариантов.

- **Цель – сбор информации для ЛПР.** Тогда экспертная комиссия должна собрать возможно больше относящейся к делу информации, аргументов «за» и «против» определенных вариантов решений. Отметим, что эксперты в рассматриваемом методе только поставляют информацию, но не вырабатывают согласованного проекта решения. Поэтому нет никакой необходимости стремиться к тому, чтобы экспертные мнения были согласованы между собой.
- **Цель – подготовка проекта решения для ЛПР.** В этом случае для подготовки проекта решения обычно

применяются математические методы оценки согласованности мнений экспертов.

Математические методы оценки мнений экспертов следует применять, ясно понимая область их применимости.

Например, считается, что решение может быть принято лишь на основе согласованных мнений экспертов. Поэтому исключают из экспертной группы тех, чье мнение отличается от мнения большинства. При этом отсеиваются как неквалифицированные лица, попавшие в состав экспертной комиссии по недоразумению, так и наиболее оригинальные мыслители, глубже проникшие в проблему, чем большинство. Следовало бы выяснить их аргументы, предоставить им возможность для обоснования их точек зрения, вместо этого их мнением пренебрегают. Бывает и так, что эксперты делятся на две или более групп, имеющих единые групповые точки зрения. Иногда заявляют, что в случае обнаружения двух или нескольких групп экспертов (вместо одной согласованной во мнениях) опрос не достиг цели. Это не так! Цель достигнута – установлено, что единого мнения нет. ЛПР должен это учитывать. Стремление обеспечить согласованность мнений экспертов любой ценой может приводить к сознательному одностороннему подбору экспертов, игнорированию всех точек зрения, кроме одной, наиболее полюбившейся ЛПР.

Поскольку число экспертов обычно не превышает 20-30, то формальная статистическая согласованность мнений экспертов (установленная с помощью тех или иных критериев проверки статистических гипотез) может сочетаться с реально имеющимся разделением на группы, что делает дальнейшие расчеты не имеющими отношения к действительности. Другими словами, мы падаем жертвой заблуждений, вытекающих из своеобразного толкования слов: проверка согласованности в указанном статистическом смысле вовсе не является проверкой согласованности в смысле практики экспертных оценок.

С целью искусственно добиться согласованности стараются уменьшить влияние мнений экспертов-диссидентов. Жесткий способ борьбы с диссидентами состоит в их исключении из состава экспертной комиссии. Мягкий способ борьбы с диссидентами состоит в применении робастных (устойчивых) статистических процедур. Простейший пример: если ответ эксперта – действительное число, то резко выделяющееся мнение диссидента сильно влияет на среднее арифметическое ответов экспертов и не влияет на их медиану. Поэтому разумно в качестве согласованного мнения рассматривать медиану. Однако при этом игнорируются (не достигают ЛПР) аргументы диссидентов.

В любом из двух способов борьбы с диссидентами ЛПР лишается информации, идущей от диссидентов, а потому может принять необоснованное решение, которое приведет к отрицательным

последствиям. С другой стороны, представление ЛПР всего набора мнений снимает часть ответственности и труда по подготовке окончательного решения с комиссии экспертов и рабочей группы по проведению экспертного опроса и перекладывает ее на плечи ЛПР.

Распространен довольно примитивный подход так называемой «квалиметрии», согласно которому объект всегда можно оценить одним числом. Каждый объект можно оценивать по многим показателям качества. Например, программный продукт можно оценивать по таким показателям:

- требования к минимальным необходимым ресурсам;
- количество известных некритических ошибок;
- максимальное время реакции на событие и т.п.

Можно ли свести оценки по этим показателям вместе? Определяющей является конкретная ситуация, для которой выбирается продукт. Максимальное время реакции критически важно для программы управления ракетой, но, как нам представляется, не имеет большого практического значения для пользователя рядового настольного приложения. Для пользователя важнее, чтобы требования к ресурсам не выходили за пределы возможностей недорогого офисного компьютера.

Таким образом, важна конкретная (узкая) постановка задачи перед экспертами. Но такой постановки зачастую нет. А тогда искусственно введенный обобщенный показатель качества – например, в виде линейной функции от субъективно выбранных переменных – не может дать объективных выводов.

Как показывает опыт проведения экспертных исследований, целесообразно выделять следующие *стадии экспертного опроса*:

- 1) формулировка ЛПР, цели экспертного опроса;
- 2) разработка и утверждение у ЛПР технического задания на проведение экспертного опроса;
- 3) разработка подробного сценария проведения сбора и анализа экспертных мнений (оценок), включая конкретный вид экспертной информации и конкретные методы анализа этой информации;
- 4) подбор экспертов в соответствии с их компетентностью и формирование экспертной комиссии;
- 5) проведение сбора экспертной информации;
- 6) анализ экспертной информации;
- 7) при применении процедуры из нескольких туров – повторение двух предыдущих этапов;

- 8) интерпретация полученных результатов и подготовка заключения для ЛПР;
- 9) официальное окончание деятельности экспертной комиссии.

Проблема подбора экспертов является одной из наиболее сложных. Очевидно, в качестве экспертов необходимо использовать тех людей, чьи суждения наиболее помогут принятию адекватного решения. Но как выделить, найти, подобрать таких людей? Надо прямо сказать, что нет методов подбора экспертов, наверняка обеспечивающих успех экспертизы.

Часто предлагают использовать **методы взаимной оценки и самооценки компетентности экспертов**. С одной стороны, кто лучше может знать возможности эксперта, чем он сам? С другой стороны, при самооценке компетентности скорее оценивается степень самоуверенности эксперта, чем его реальная компетентность. Тем более, что само понятие «компетентность» строго не определено. Можно его уточнять, выделяя составляющие, но при этом усложняется предварительная часть деятельности экспертной комиссии.

При использовании метода взаимной оценки, помимо возможности проявления личностных и групповых симпатий и антипатий, играет роль неосведомленность экспертов о возможностях друг друга. В современных условиях достаточно хорошее знакомство с работами и возможностями друг друга может быть лишь у специалистов, много лет работающих совместно. Однако привлечение таких пар специалистов не очень-то целесообразно, поскольку они слишком похожи друг на друга.

Использование формальных показателей (должность, ученые степень и звание, стаж, число публикаций...), очевидно, может носить лишь вспомогательный характер. **Успешность участия в предыдущих экспертизах** – хороший критерий для деятельности дегустатора, врача, судьи в спортивных соревнованиях, т.е. таких экспертов, которые участвуют в длинных сериях однотипных экспертиз. Однако, увы, наиболее интересны и важны уникальные экспертизы больших проектов, не имеющих аналогов.

Необходимо подчеркнуть, что подбор экспертов, в конечном счете – функция ЛПР, и никакие методики подбора не снимают с ЛПР ответственности за результаты принятых решений.

3.4.4. Проблема горизонта планирования

Во многих ситуациях продолжительность проекта не определена, либо горизонт планирования инвестора не охватывает всю продолжительность реализации проекта до этапа утилизации. В таких

случаях важно изучить влияние горизонта планирования на принимаемые решения.

Рассмотрим условный пример.

Предположим, руководитель среднего предприятия по разработке программного обеспечения должен принять решение о вложении свободных средств, образовавшихся в результате предыдущей деятельности. Основным критерий при этом – увеличение прибыли.

- ✓ Если горизонт планирования – 1 месяц, то наибольший денежный доход можно получить, немедленно продав предприятие.
- ✓ Если же планирование ведется на год, то можно вложить средства в инициативную разработку собственного продукта, в расчете получить прибыль через год.
- ✓ Если горизонт планирования составляет 10 лет, то целесообразно пойти на крупные затраты, закупив лицензии на новейшие технологии и проведя массовое переобучение персонала, с целью увеличения дохода в дальнейшие годы.
- ✓ При планировании на 30 лет имеет смысл вложить средства в создание и развитие собственного научно-исследовательского центра, и т.д.

Таким образом, популярное утверждение «фирма работает ради максимизации прибыли» не имеет точного смысла. За какой период максимизировать прибыль – за месяц, год, 10 или 30 лет?

От горизонта планирования зависят принимаемые решения. Понимая это, ряд западных экономистов отказываются рассматривать фирмы как инструменты для извлечения прибыли, предпочитают смотреть на них как на живые существа, старающиеся обеспечить свое существование и развитие.

3.5. Заключение

Принятие решений при проведении проектов разработки программного продукта имеет все основные характеристики, присущие принятию решений вообще, а также некоторые специальные аспекты, связанные с особенностями процесса разработки программного обеспечения.

3.6. «Карта памяти» по теме 3



3.7. Список использованной и дополнительной литературы

1. Пашкус Ю. В. Мисько О. Н. Введение в бизнес. - Л.: "Северо-Запад", 1991.
2. Голубков Е.П. Какое принять решение? М.: Экономика, 1990.
3. Голубков Е.П. Технология принятия управленческих решений. М.: Издательская группа "Дело и сервис", 2005.
4. М. Эдоус, Р. Стэнсфилд. Методы принятия решения. М.: Издательство объединения "Юнити", 1997.
5. В. Н. Спицнадель. Теория и практика принятия оптимальных решений. М.: Бизнес-пресса, 2002.
6. Балабанов И.Т. Риск-менеджмент. М.: Финансы и статистика, 1996.

Тема 4. Проектный менеджмент

4.1. Введение

В этой теме рассматриваются основные основы проектного менеджмента в отличие от менеджмента вообще (последнее – существенно более широкое понятие, включающее в себя, в частности, и менеджмент компании).

Изучив учебный материал данной темы, Вы:

- узнаете или пополните свои знания о том, что такое проект, и что проектом не является;
- узнаете или пополните свои знания о типах проектов и областях эффективного применения проектного менеджмента;
- узнаете или пополните свои знания о жизненном цикле проекта в области разработки программного обеспечения и об основных стандартах в области управления проектами разработки программного обеспечения.

В рамках темы рассматриваются следующие учебные вопросы:

- Области эффективного применения проектного менеджмента.
- Типы проектов.
- Жизненный цикл проекта по созданию программного обеспечения.
- Основные стандарты в области управления проектами разработки программного обеспечения.

4.2. Области эффективного приложения проектного менеджмента

Обратимся к понятию «проект» и выделим четыре характеристики, делающих деятельность проектом.

- **Направленность на достижение конкретных целей.** Действительно проекты направлены на получение определенных результатов. Кстати, цели должны быть сформулированы так, чтобы всегда было ясно, что цель достигнута и проект можно заканчивать. Проект обычно предполагает наличие промежуточных целей, которые вместе образуют взаимосвязанный комплекс.
- **Координированное выполнение взаимосвязанных действий.** Здесь следует еще раз вспомнить сложность

разработки программного продукта. Взаимосвязи в проекте не всегда очевидны. Некоторые задания должны выполняться строго последовательно, а некоторые – строго параллельно.

- **Ограниченная протяженность во времени с определенным началом и концом.** Проект имеет определенный срок. Иногда этот срок плавающий, например, начинающийся от некоторой еще не определенной даты вступления договора в силу. Проект существует столько времени, сколько требуется для получения конечного результата.
- **Уникальность и важность.** Уникальность должна объяснять – почему надо создавать данный программный продукт, почему нельзя взять что-то готовое. Элемент важности должен демонстрировать, почему разрабатываемый продукт нужен и важен заказчику, какие нужды и потребности заказчика он покрывает.

Простое несение обязанностей, деятельность без четких границ или бесцельное времяпровождение проектами не являются, поскольку не имеют определенных сроков и конкретных целей!

Таким образом,

под **управлением проектом** будем понимать деятельность, направленную на реализацию проекта с максимально возможной эффективностью при заданных ограничениях по времени, денежным средствам и ресурсам, а также качеству конечных результатов проекта.

4.3. Типы проектов

Проекты в области информационных технологий могут быть классифицированы по различным признакам. Далее мы рассмотрим только проекты, имеющие отношение к программному обеспечению (ПО).

Проекты в области разработки или сопровождения ПО разделяются на категории следующим образом.

- По уровню сложности и области применения разрабатываемого или модифицируемого ПО.
 - * Проекты по разработке/модификации ответственного ПО (система управления атомным реактором). Такие проекты не обязательно требуют для выполнения большой команды исполнителей, однако, предъявляют

повышенные требования к качеству процесса разработки, квалификации персонала и качеству продукта.

- * Проекты, связанные со сложными многофункциональными системами (Enterprise Resource Planning (ERP) – Управление ресурсами предприятия). Сложность состоит, как правило, в управлении требованиями (сбор требований о функциональности) и координации усилий большой команды исполнителей.
- * Малые проекты (Интернет-магазины, сайты, простые настольные (standalone) приложения). Здесь критической, как правило, является скорость и стоимость разработки.
- * «Научеёмкие» проекты (численные расчеты, разработка новых специальных алгоритмов). Сложность – в творческом характере процесса разработки.
- * Проекты по сопровождению или модификации больших унаследованных приложений, систем или баз данных (системы продажи авиабилетов или железнодорожных билетов, существующие с 70х гг., в США – экономические или бухгалтерские системы, написанные на COBOLe для mainframe-систем в 50-60е гг.). Основные сложности: большой объем, как правило, плохо документированного кода, над которым трудилось несколько поколений разработчиков, отсутствие проектной документации, необходимость работы с устаревшими технологиями и недоступность соответствующих специалистов.

■ По способу применения разрабатываемого или модифицируемого ПО.

- * Проекты по разработке неотчуждаемого ПО («для себя»).
- * Разработка для заранее известного (не чрезмерно широкого) круга пользователей без дальнейшего тиражирования («внутреннее ПО»).
- * Разработка или модификация ПО для конкретного заказчика.
- * Разработка «коробочного продукта».

■ По характеру отношений с заказчиком/потребителем.

- * Аутсорсинговые проекты – постановка целей и частично управление проектом ведутся на стороне заказчика; на стороне исполнителя ведется, в основном, реализация и кодирование.

- * Заказные проекты – постановка целей частично происходит на стороне заказчика, но все управление проектом ведется на стороне исполнителя.
- * Проекты модификации или сопровождения ПО третьей стороны.
- * Инициативная разработка – все аспекты проекта: постановка целей, планирование, управление, разработка и сопровождение ведутся на стороне исполнителя.

Проекты разных категорий требуют применения различных методов управления.

4.4. Жизненный цикл проекта разработки программного продукта

Проект по разработке программного продукта, как и всякий проект, имеет начало и конец, то есть развивается во времени. Во время проведения проекта выполняются некоторые операции, причем не хаотически, а в определенной взаимосвязи и последовательности, образуя, тем самым, процессы. Замечено, что выполнение процессов распределено по времени выполнения всего проекта неравномерно. Одни процессы, например, управление проектом, выполняются все время, пока выполняется проект. Другие процессы, например анализ требований, выполняются только в начале проекта. Третьи процессы, например сопровождение продукта, выполняются только в конце.

Какие процессы выполняются и как они соотносятся друг с другом в конкретном проекте, зависит от многих факторов, из которых важнейшими являются следующие два:

- тип проекта и особенности продукта;
- потребности и возможности организации, проводящей проект.

Для того, чтобы было удобно управлять проектом и сравнивать различные способы проведения проектов, принято делить проект на некоторые периоды, называемые фазами (или стадиями), в которых соотношение и взаимосвязь процессов остаются примерно постоянными.

Жизненный цикл проекта (Project Life-Cycle) – набор последовательных фаз проекта, название и число которых определяется потребностями организации, выполняющей проект и типом проекта.

4.4.1. Модели жизненного цикла программных продуктов

Жизненный цикл программы – это весь период ее разработки и эксплуатации, начиная с момента возникновения замысла и заканчивая прекращением всех видов ее использования.

Давно замечено, что программа за время жизни претерпевает многочисленные изменения своей формы, зависящие от состояния процесса разработки и эксплуатации. Обычно совокупность и последовательность этих изменений и называется *жизненным циклом*. В разных парадигмах и технологиях программирования понятие жизненного цикла программы определяется и трактуется немного по-разному, но в общем близко к схеме, представленной на рис. 1. Важно подчеркнуть, что за время своей жизни программа проходит метаморфозы, как правило, несколько раз (чего, к сожалению, не случается с программистами), т.е. это именно цикл, причем не один, а несколько.

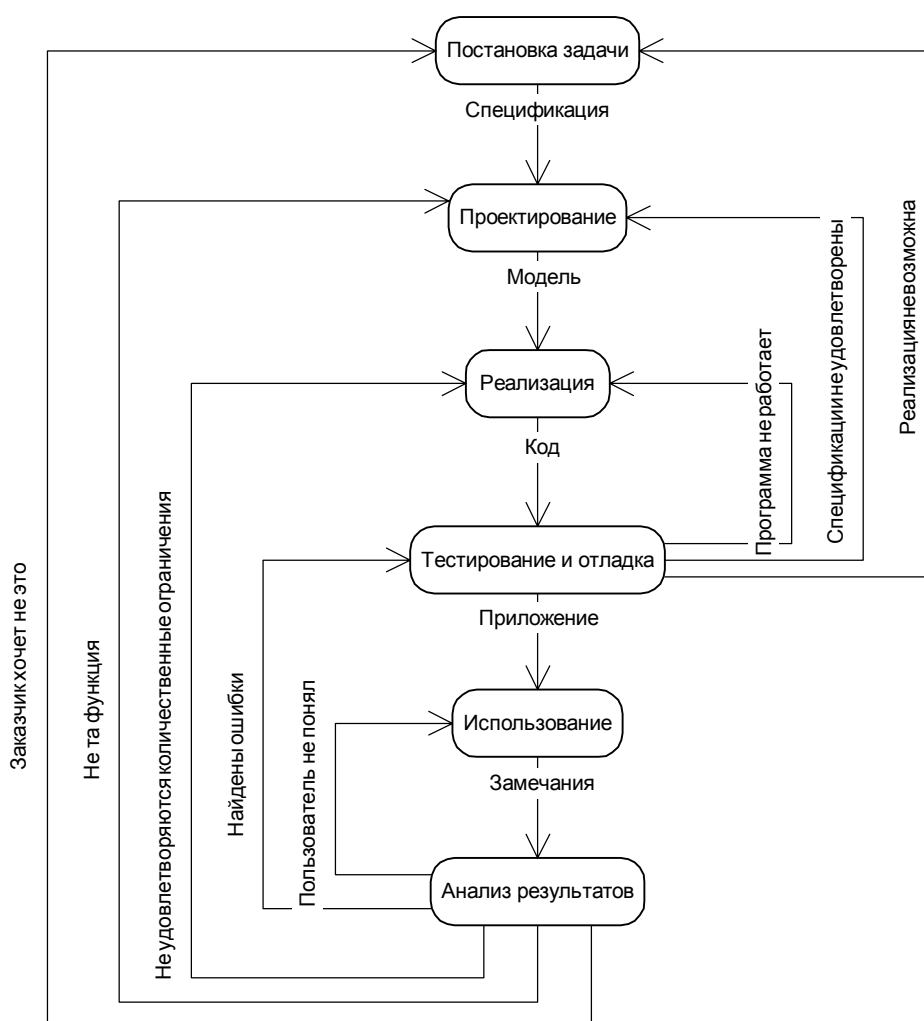


Рис. 1. Жизненный цикл программы

Жизненный цикл программы тесно связан с жизненным циклом проекта по разработке, особенно если в проект включают такие фазы, как сопровождение и модификация программы во время эксплуатации.

Технология программирования (Software Engineering) изучает отдельные технологические *процессы* и порядок их прохождения — *стадии* и *фазы*, а также более крупные структуры: жизненные циклы продуктов и проектов (с использованием определенных знаний, методов и средств). Результаты исследований обычно представляются для общего пользования в форме *моделей* жизненных циклов, то есть обобщенных описаний типовых способов проведения проектов разработки программных продуктов.

Модель жизненного цикла удобно характеризовать в двух измерениях — вертикальном (представляющем процессы) и горизонтальном (представляющем стадии).

Процесс — совокупность взаимосвязанных действий, преобразующих некоторые входные данные в выходные.

Процессы состоят из набора действий, а каждое действие из набора задач. Вертикальное измерение отражает статические аспекты процессов и оперирует такими понятиями, как рабочие процессы, действия, задачи, результаты деятельности и исполнители.

Стадия — часть действий по созданию программного обеспечения, ограниченная некоторыми временными рамками и заканчивающаяся выпуском конкретного продукта, определяемого заданными для данной стадии требованиями. Конкретный продукт называется артефактом стадии, момент его выпуска называется вехой или контрольной точкой.

Стадия — часть процесса работы над проектом. Каждая стадия характеризуется вехой, достижение которой знаменует завершение стадии.

Стадии состоят из *этапов*, которые обычно имеют итерационный характер. Иногда стадии объединяют в более крупные временные рамки, называемые *фазами*.

Следует подчеркнуть, что деление процесса на этапы, стадии и фазы носит объективный характер, поскольку определяется объективными событиями — вехами — выпуском тех или иных артефактов.

Веха — одномоментное идентифицируемое событие, сопровождающееся появлением и фиксацией некоторого отчуждаемого материала (документа, программы, протокола), который называется артефактом вехи.

Итак, горизонтальное измерение представляет время, отражает динамические аспекты процессов и оперирует такими понятиями, как фазы, стадии, этапы, итерации и контрольные точки.

Модель (или технологический подход) определяется спецификой комбинации стадий и процессов, ориентированной на разные классы программного обеспечения и на особенности коллектива разработчиков.

Простейшее представление жизненного цикла программы представлено на рис. 2.



Рис. 2 Каскадная модель жизненного цикла проекта.

Фактически, в данном случае на каждой стадии выполняется единственный процесс. Конечно, при разработке и создании больших программ такая схема недостаточно корректна (да и просто нереалистична). Однако мы можем взять ее за основу для многих других технологических подходов к ведению жизненного цикла.

4.4.2. История и эволюция

В истории технологии программирования можно выделить три этапа.

- **Осмысление опыта разработки больших систем.** Понимание того, что важно не только на каком языке программирования

разрабатывается программа, но и как это делается. Проведение первых международных и национальных конференций (конец 60-х — 70-е годы XX века).

- о 1968 г. — НАТО проводит первую конференцию по инженерии программирования (Software Engineering).
- о 1975 г. — 1-я международная конференция IEEE.
- о 1979 г. — 1-я Всесоюзная конференция по технологии программирования.
- **Разработка новых технологических подходов** (начало 70-х годов XX века — настоящее время).
 - о 1973 г. — Дагласом Россом (Douglas Ross) разработана технология проектирования сложных систем SADT (Structured Analysis and Design Technique). Стандартизована под названием IDEF (Integrated DEFinition).
 - о 1985 г. — Харланом Миллзом (Harlan Mills) сформулированы основные идеи технологии стерильного цеха.
 - о 1997 г. — в ноябре появилась первая версия Унифицированного языка моделирования UML (Unified Modeling Language). С точки зрения технологических подходов особый интерес представляет рациональный унифицированный процесс, описанный с использованием UML.
- **Принятие стандартов на состав процессов жизненного цикла программного обеспечения** (середина 80-х годов XX века — настоящее время). Попытки решить проблему качества программных продуктов.
 - о 1978 г. — В СССР утверждена Единая система программной документации (ЕСПД) — сборник стандартов, определяющий взаимоувязанные правила разработки, оформления и обращения программ и программной документации.
 - о 1985 г. — впервые утвержден стандарт жизненного цикла для проектирования программных систем (для систем военного назначения по заказам Министерства обороны США).
 - о 1994 г. — в Великобритании создан международный консорциум, разрабатывающий на постоянной основе проекты стандартов и технологии быстрого создания приложений DSDM (Dynamic Systems Development Method).
 - о 1995 г. — принят международный стандарт ISO 12207:1995 «Information Technology — Software Life Cycle

Processes», регламентирующий состав процессов жизненного цикла программного обеспечения.

4.4.3. Классификации моделей жизненного цикла

К настоящему времени было предложено множество моделей жизненного цикла программного продукта и процесса разработки, а также постоянно появляются новые предложения. Большинство этих моделей используются на практике в тех или иных типах проектов. Для обзора всего многообразия моделей целесообразно их классифицировать по подходам, процессам и стадиям.

Классификация технологических подходов

Выделим основные группы технологических подходов и укажем область применения для каждой из них.

- **Подходы со слабой формализацией.** Эти подходы не используют явных технологий и их можно применять только для очень маленьких проектов, как правило, завершающихся созданием демонстрационного прототипа. К подходам со слабой формализацией относятся так называемые ранние технологические подходы, например подход «кодирование и исправление».
- **Строгие (классические, жесткие, предсказуемые) подходы.** Данную группу подходов рекомендуется применять для средних, крупномасштабных и гигантских проектов с фиксированным объемом работ. Одно из основных требований к таким проектам — предсказуемость. В эту группу входят подходы, перечисленные ниже.
 - Каскадные технологические подходы.
 - Классический каскадный подход.
 - Каскадно-возвратный подход.
 - Каскадно-итерационный подход.
 - Каскадный подход с перекрывающимися процессами.
 - Каскадный подход с подпроцессами.
 - Спиральная модель.
 - Каркасные подходы.
 - Рациональный унифицированный процесс.
 - Генетические подходы.
 - Синтезирующее программирование.
 - Сборочное (расширяемое) программирование.
 - Конкретизирующее программирование.
 - Подходы на основе формальных преобразований.

- Технология стерильного цеха.
- Формальные генетические подходы.
- **Гибкие (адаптивные, легкие) подходы.** Подходы этой группы рекомендуется применять для небольших или средних проектов в случае неясных или изменяющихся требований к системе. Команда разработчиков должна быть ответственной и квалифицированной, а заказчики должны быть согласны принимать участие в разработке. В данную группу входят подходы, перечисленные ниже.
 - Ранние технологические подходы быстрой разработки.
 - Эволюционное прототипирование.
 - Итеративная разработка.
 - Постадийная разработка.
 - Адаптивные подходы.
 - Экстремальное программирование.
 - Адаптивная разработка.
 - Подходы исследовательского программирования.
 - Компьютерный дарвинизм.

Классификация технологических процессов

Мы будем рассматривать два набора (множества) технологических процессов. Первый набор — классический, включающий основные процессы, сложившиеся исторически в результате практического опыта разработки программного обеспечения. Второй набор — стандартный, т. е. основанный на стандарте ISO 12207:1995. Процессы классического набора фактически являются подмножеством стандартного, выступая там как процессы или действия процессов.

В классическом наборе выделим девять технологических процессов.

- Возникновение и исследование идеи.
- Управление.
- Анализ требований.
- Проектирование.
- Программирование.
- Тестирование и отладка.
- Ввод в действие.
- Эксплуатация и сопровождение.
- Завершение эксплуатации.

Процессы жизненного цикла, определяемые международным стандартом

ISO 12207 [ISO/IEC 12207:1995], делятся на три группы.

- Основные процессы.
 - Приобретение.
 - Поставка.

- o Разработка.
 - o Эксплуатация.
 - o Сопровождение.
- Вспомогательные процессы.
 - o Документирование.
 - o Управление конфигурацией.
 - o Обеспечение качества.
 - o Верификация.
 - o Аттестация.
 - o Совместная оценка.
 - o Аудит.
 - o Разрешение проблем.
- Организационные процессы.
 - o Управление.
 - o Создание инфраструктуры.
 - o Усовершенствование.
 - o Обучение.

Классификация технологических стадий

Технологические стадии выделяются исходя из соображений разумного и рационального планирования и организации работ. Существует два основных варианта формирования промежутков времени, поддерживаемых технологическими подходами. В первом – формируются фазы, отражающие крупные временные этапы. Например, начальная фаза, середина, кульминация, окончание. Во втором варианте – определяются стадии, отражающие названия классических процессов (или их подмножества или надмножества), большая часть времени которых проходит в данной стадии. Пример взаимосвязи между стандартными процессами и стадиями показан на рис. 3



Рис. 3 Взаимосвязь между стандартными процессами и стадиями

С 1980 года на территории Российской Федерации действует ГОСТ 19.102-77, входящий в систему ГОСТов ЕСПД, который устанавливает стадии разработки программ и программной документации для вычислительных машин, комплексов и систем независимо от их назначения и области применения. Стадии разработки, этапы и содержание работ должны соответствовать указанным в таблице 1.

Таблица 2.

Стадии разработки согласно ГОСТ 19.102-77

Стадии разработки	Этапы работ	Содержание работ
1. Техническое задание	Обоснование необходимости разработки программы	Постановка задачи Сбор исходных материалов Выбор и обоснование критериев эффективности и качества

		разрабатываемой программы. Обоснование необходимости проведения научно-исследовательских работ.
	Научно-исследовательские работы	Определение структуры входных и выходных данных. Предварительный выбор методов решения задач. Обоснование целесообразности применения ранее разработанных программ. Определение требований к техническим средствам. Обоснование принципиальной возможности решения поставленной задачи
	Разработка и утверждение технического задания	Определение требований к программе. Разработка технико-экономического обоснования разработки программы. Определение стадий, этапов и сроков разработки программы и документации на неё. Выбор языков программирования. Определение необходимости проведения научно-исследовательских работ на последующих стадиях. Согласование и утверждение технического задания.
2. Эскизный проект	Разработка эскизного проекта	Предварительная разработка структуры входных и выходных данных. Уточнение методов решения задачи. Разработка общего описания алгоритма решения задачи Разработка технико-экономического обоснования.
	Утверждение эскизного проекта	Разработка пояснительной записки. Согласование и утверждение эскизного проекта.
3. Технический проект	Разработка технического проекта	Уточнение структуры входных и выходных данных. Разработка алгоритма решения задачи. Определение формы представления входных и выходных данных. Определение семантики и синтаксиса языка.

		Разработка структуры программы. Окончательное определение конфигурации технических средств.
	Утверждение технического проекта	Разработка плана мероприятий по разработке и внедрению программ. Разработка пояснительной записки. Согласование и утверждение технического проекта.
4. Рабочий проект	Разработка программы	Программирование и отладка программы.
	Разработка программной документации	Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.
	Испытания программы	Разработка, согласование и утверждение порядка и методики испытаний. Проведение предварительных государственных, межведомственных, приёмо-сдаточных и других видов испытаний. Корректировка программы и программной документации по результатам испытаний.
5. Внедрение	Подготовка и передача программы.	Подготовка и передача программы и программной документации для сопровождения и (или) изготовления. Оформление и утверждение акта о передаче программы на сопровождение и (или) изготовление. Передача программы в фонд алгоритмов и программ.

Примечания:

1. Допускается исключать вторую стадию разработки, а в технически обоснованных случаях — вторую и третью стадии. Необходимость проведения этих стадий указывается в техническом задании.

2. Допускается объединять, исключать этапы работ и (или) их содержание, а также вводить другие этапы работ по согласованию с заказчиком.

4.5. Комплекс работ по внедрению технологии управления проектом в организации

При внедрении в организации новой технологии управления проектами или при радикальном изменении существующей технологии необходимо провести следующий комплекс работ.

Вовлечение руководства.

Для успешного внедрения в организации новой технологии управления проектами совершенно необходима заинтересованность и участие в этом процессе высшего руководства компании. Внедрение новой технологии управления проектами «на свой страх и риск» менеджерами среднего звена в большинстве случаев обречено на провал, и не даст ничего, кроме глубокого разочарования в инновациях. Более того, провал проекта по внедрению надолго отобьет охоту к нововведениям и замедлит прогресс организации в целом. Для успешного внедрения высшее руководство обязано выделить необходимые ресурсы: финансовые, материальные и кадровые. Выделение кадровых ресурсов является «лакмусовой бумажкой» вовлеченности руководства. Если руководство компании выделяет из своего состава менеджера высшего звена, возлагает на него ответственность за результаты внедрения и наделяет необходимыми полномочиями, то можно рассчитывать на положительный результат внедрения. В противном случае, когда неизвестно, кто отвечает за результат, положительного результата, скорее всего, не будет.

Разработка документации.

Новая технология управления проектом должна быть детально документирована. При этом старая технология (или отсутствие технологии) управления проектом могла не быть подробно документированной – менеджеры просто привыкли к старой технологии и делали все «как раньше», без письменных инструкций. Для новой технологии этот способ, очевидно, не подходит. Подчеркнем, что детальность документации не обязательно влечет необозримый объем. Использование современных методов документирования процессов позволяет очень компактно и в то же время с необходимыми подробностями описать всю последовательность и взаимосвязи необходимых действий и документов.

Ярким примером является Рациональный унифицированный процесс (Rational Unified Process (RUP)). Будучи весьма детальным описанием развитой системы взаимодействующих процессов, RUP относительно компактен, поскольку для описания процессов используется формализованная нотация Унифицированного языка моделирования (Unified Modeling Language (UML)).

Подготовка инфраструктуры.

Новая технология управления проектами часто требует совершенствования инфраструктуры организации. Например, переход на безбумажный документооборот требует наличия быстродействующей локальной сети, что не всегда легко обеспечить, если компания располагается в нескольких офисах или если разработчики работают вне офиса, дома. Другой пример: в случае внедрения технологии подобной экстремальному программированию, необходимо реорганизовать рабочее пространство и производственную среду, так, чтобы парное программирование и частные производственные совещания можно было проводить быстро и без помех. В любом случае до внедрения новой технологии должны быть определены и реализованы все необходимые мероприятия по совершенствованию инфраструктуры.

Обучение персонала.

Весь персонал должен пройти специальное обучение практическому применению новой технологии. Это касается в первую очередь менеджеров проектов, но распространяется и на другие категории: разработчиков, тестеров, продавцов. Каждый участник процесса должен точно знать, что от него требуется по новым правилам, и, самое главное, должен уметь это делать. Конечно, люди обучаемы и могут научиться чему угодно «в процессе производства», но издержки будут велики. Полезно помнить высказывание полководца Суворова: «тяжело в учении – легко в бою». Руководство компании должно зарезервировать время, необходимое для обучения.

Установка инструментов.

Современные технологии управления проектами по разработке программных продуктов глубоко автоматизированы и ориентированы на использование многочисленных компьютерных инструментов. Все инструменты, намечаемые к использованию, должны быть лицензированы, установлены и попасть под управление ответственных и уполномоченных лиц (системных администраторов). Если разработчики программного обеспечения сами выбирают систему управления конфигурациями «кто во что горазд», то трудно надеяться на успешное внедрение современной технологии управления проектами.

Проведение испытаний.

Новую технологию управления проектами нельзя считать успешно внедренной до тех пор, пока она не прошла апробацию на практике при проведении конкретных проектов. Для пробного внедрения рекомендуется использовать простой типовой проект из числа проводимых организацией. В организациях проектного типа целесообразно сам проект по внедрению новой технологии проводить по этой технологии.

Оценка результатов.

После проведения испытаний новой технологии на примере пилотного проекта необходимо провести анализ и оценить полученные результаты внедрения. Проведение анализа должно опираться на измерения, проводимые в процессе пилотного проекта. Руководство компании должно быть готово к беспристрастному и честному анализу результатов. В частности, анализ результатов может показать, что новая технология не подходит данной организации или что организация не готова к внедрению технологии. В таком случае дальновиднее отказаться от внедрения.

4.6. Функции менеджера проекта

Выделим две группы обязанностей менеджера проекта. Первая группа выполняется совместно с техническим лидером проекта, а вторая находится в исключительной ответственности менеджера.

Совместная деятельность менеджера и лидера проекта включает:

- планирование проекта;
- распределение работ;
- выбор наилучшей стратегии.

Исключительная ответственность менеджера заключается:

- в организации взаимосвязей внутри организации;
- в управлении сотрудниками. Менеджер ищет и привлекает лучших специалистов и экспертов для решения возникающих проблем;
- в руководстве проектом и контроле его выполнения. Менеджер отвечает за ежедневное руководство данным проектом. Хороший менеджер отводит различные проблемы от группы, он «гасит» нагоняи от заказчика или вышестоящего руководства;
- в том, чтобы проект отвечал требованиям заказчика. Руководитель должен держать заказчика в курсе всех событий проекта;
- в ответственности за поступление средств.

Менеджер не должен позволять вмешиваться кому-либо другому в руководство проектом. На ежедневное руководство, то есть на выполнение обязанностей второй группы в среднем у менеджера должно уходить 50% рабочего времени. Если менеджер тратит менее 50% времени на руководство, это означает, скорее всего, что он плохой руководитель и проект ждут серьезные проблемы.

4.7. Основные стандарты в области управления проектами разработки программного обеспечения

Стандарт – общепринятое определение компонента технических или программных средств, являющихся результатом соглашения.

Профиль – набор юридических и/или фактических стандартов, ориентированных на выполнение конкретной задачи.

4.7.1. Классификация стандартов

Стандарты можно классифицировать следующим образом:

- по типу установления требований:
 - устанавливающие требования к объекту;
 - устанавливающие требования к процессу;
- по масштабу:
 - международные;
 - государственные;
 - отраслевые;
 - предприятий;
- по степени юридического оформления:
 - принятые юридически;
 - действующие фактически.

4.7.2. Организации по стандартизации

Процесс стандартизации информационных технологий поддерживают три основные группы организаций (<http://www.citforum.ru/programming/prg96/sukhomlin.shtml>).

- Международные организации, входящие в структуру ООН.
 - International Organization for Standardization (ISO) – международная организация по стандартизации. В 1947 году представители 25 стран решили создать организацию, основной задачей которой стала бы координация разработок и унификация международных стандартов. Новая организация получила название International Organization for Standardization (ISO). Несоответствие полного названия и аббревиатуры объясняется тем, что «ISO» — это греческий префикс, означающий «равный».
 - International Electrotechnical Commission (IEC) – международная электротехническая комиссия.
 - International Telecommunication Union-Telecommunications (ITU-T) – международный союз по

телекоммуникации – телекоммуникация. До 1993 года эта организация называлась International Telegraph and Telephone Consultative Committee (ITTCC) – международный консультативный комитет по телефонии и телеграфии.

- Промышленные профессиональные или административные организации.
 - o Institute of Electrical and Electronic Engineers (IEEE) – институт инженеров по электротехнике и электронике.
 - o Internet Activity Board (IAB) – совет управления деятельностью Интернета.
- Промышленные консорциумы.
 - o Object Management Group (OMG) – группа управления объектами. Разрабатывает, в частности, стандарты CORBA, UML, XMI, MOF.
 - o X/Open – консорциум, организованный группой поставщиков компьютерной техники.
 - o Open Software Foundation (OSF) – фонд открытого программного обеспечения.

В 1987 году ISO и IEC объединили свою деятельность в области стандартизации информационных технологий и создали единый орган – Joint Technical Committee 1 (JTC1) – объединенный технический комитет 1. Этот комитет предназначен для формирования системы базовых стандартов в области информационных технологий.

4.7.3. Общие стандарты управления проектами

В настоящее время в мире существует большое количество стандартов и методологий управления проектами, которые имеют распространение как на международном, так и на национальном уровнях.

Стандарты по управлению *единичным проектом* представлены

- Руководством к своду знаний по управлению проектами – PMBOK (Project Management Body of Knowledge),
- Руководством по качеству при управлении проектами (Guidelines to Quality in Project Management) — ISO 10006,
- Системой знаний о процессах управления проектами — PRINCE 2 (PRojects IN Controlled Environments),

и являются наиболее ранней и достаточно проработанной по структуре и содержанию группой стандартов.

В группе стандартов по управлению *портфелями проектов* заслуживает внимания готовящийся к официальному принятию PMI драфт – Portfolio management, основанный на PMBOK и Модели организационной зрелости управления проектами — OPM3.

Среди стандартов, определяющих *требования к компетенции менеджера проекта*, в качестве основных можно выделить Международные требования к компетенции специалистов по управлению проектами (PM ICB), разработанных Международной ассоциацией управления проектами IPMA (Швейцария), а также основанный на них российский стандарт — Национальные требования к компетенции СОВНЕТ (Россия). В рамках данных стандартов профессионализм менеджера проекта определяется четырехуровневой системой оценки. По результатам работы инициативной группы Австралийского института управления проектами AIPM совместно с экспертами PMI подготовлены Основы развития компетенции менеджера проекта – PMCDF, согласованные с требованиями PMI к сертификации профессионалов по управлению проектами (PMR).

Комплексное представление о *системе знаний управления проектами в масштабах всей организации*, или управления масштабными *программами или портфелями проектов* можно получить, ознакомившись с группой стандартов, методологии которых позволяют разрабатывать модели корпоративных систем управления проектами. Наиболее известные из них – Модели организационной зрелости управления проектами OPM3 (PMI) и разработанный Ассоциацией инновационного развития и управления проектами Японии Program and Project Management for Innovation of Enterprises (P2M).

Кроме того, разработано множество национальных стандартов управления проектами, представленных APM (Великобритания), VZPM (Швейцария), GPM (Германия), AFITEP (Франция), CEPМ (Индия), PROMAT (Южная Корея) и другими. Вот некоторые из наиболее популярных национальных стандартов: BS 6079 (British Standards Board, 1996 г.), DIN 69 900 series x-50-100 series (German standards DIN 69 900 to 69 903 and 69 905), APM BOK (версия. 3.0) (APM Association for Project Managers: Body of Knowledge, пересм. март 1996 г. (версия 3), High Wycombe, 1996 г.), Australian National Competency Standards for Project Management (AIPM (Sponsor), 1996 г.).

Руководство к своду знаний по управлению проектами (PMBOK)

Разработчиком данного руководства является Американский Институт Управления Проектами (PMI – Project Management Institute).

PMBOK Guide является американским национальным стандартом управления проектами, содержащий сумму профессиональных знаний, основанных на лучших практиках (best practices) с использованием навыков, инструментов и методов, позволяющих успешно достичь целей проектов в различных сферах общественной и бизнес деятельности.

Целью стандарта является:

- унифицировать терминологическое пространство в области управления проектами;
- внедрить стандарт терминологии для обсуждения тем и написания статей;
- использовать данный документ как базовое справочное пособие для сертификации профессионалов по управлению проектами – PMP (Project Management Professionals);
- использовать свод знаний в образовательных целях и в обучении управлению проектами.

Это единственным на сегодняшний день стандарт в области управления проектами, который полностью соответствует ISO 9001:2000. Кроме того, он имеет весьма широкое международное распространение.

Методология управления проектами, изложенная в PMBoK, интегрирует как знания, выходящие собственно за рамки управления проектами (знания и навыки в области общего менеджмента, навыки межличностных отношений, знания, стандарты и нормативные акты, относящиеся к конкретной области приложения, понимание окружения проекта), так и знания, относящиеся исключительно к этой области: методы структурной декомпозиции работ, критического пути, освоенного объема и др. В PMBoK вошли как знания, уникальные для управления проектами, так и общие с другими дисциплинами управления: определение жизненного цикла проекта, пять групп процессов управления проектом, девять областей знаний.

В стандарте описаны разные жизненные циклы проекта и организационные структуры исполняющей организации, определены группы процессов (инициирования, планирования, исполнения, контроля, завершения) и их взаимодействие между собой, выделены основные и поддерживающие процессы, определены девять областей знаний (управление интеграцией, замыслом, временем, стоимостью, качеством, человеческими ресурсами, коммуникациями, рисками, контрактами и поставками). Стандарт базируется на процессном подходе. Для каждой области знаний определены входы, выходы и процедуры преобразования (tools and techniques) входных данных в выходные. Полностью определены взаимодействия между всеми процессами, которые включены в области знаний управления проектами.

В настоящее время PMI пошел по пути специализации и расширил PMBoK, выделив в нем следующие области:

- управление проектами со стороны правительств – Government extension to PMBoK,

- управление проектами в строительстве — Construction extension to PMBoK,
- управление стоимостью – Practice Standard for Earned Value Management,
- построение структур декомпозиции работ — Practice Standard for Work Breakdown Structures и др.

С 1999 года PMI PMBoK является национальным стандартом США как «Глоссарий терминов и сокращений» в области РМ. Третья редакция PMBoK Guide, датированная 2000 годом, подтверждена в качестве стандарта ANSI в марте 2001 года. Популярность PMBoK PMI объясняется простотой представления части знаний РМ в процессном виде и активной политикой PMI по распространению своего подхода за пределами США. Многие специалисты используют этот стандарт в качестве основы для своей деятельности и потому искренне считают его «де-факто» международным. Однако, как отмечают разработчики PMBoK, «... ни один документ не может вместить в себя всю сумму знаний». Методическая простота PMBoK PMI достигнута за счет описания упрощенной модели РМ в процессном виде, которая используется для управления одним обособленным проектом. То, что трудно или невозможно представить в виде процессов (например, стратегический менеджмент проектов, мультипроектное управление и многое другое), в этом документе должного отражения не нашло.

Руководство по качеству при управлении проектами (ISO 10006)

Во второй половине 1980-х гг. анализ накопленного обширного материала о многочисленных неудачах реализации тысяч проектов стран ЕС и США поставил под сомнение первоначально завышенные ожидания относительно возможностей управления проектами и выявить корни возникающих перед проектно-ориентированными организациями проблем. Обращение к вопросам эффективности управления выявило острую потребность в разработке системы управления качеством проекта. При этом особое значение наряду с требованиями к качеству конечного продукта стало придаваться качеству процессов проекта, отсутствие должного внимания к которым приводило к существенным отрицательным последствиям непосредственно для создаваемого продукта.

Порожденный многочисленными проблемами обеспечения качественного выполнения процессов проекта настоящий стандарт является первым (и основополагающим) из стандартов рассматриваемого профиля, где основной упор был сделан на процессы. В нем процессы по проекту сгруппированы в две категории: процессы управления проектом и процессы, связанные с продуктом проекта (то есть те процессы, которые касаются исключительно продукта проекта, такие как проектирование, производство, проверка). Описанию

последних посвящен отдельный стандарт ISO 9004-1. ISO 10006 представлен десятью группами процессов управления проектом. Первая группа представляет процесс разработки стратегии, который фокусирует проект на удовлетворение потребностей заказчика и определяет направление хода работ проекта. Вторая группа охватывает управление взаимосвязями процессов. Остальные восемь групп — это процессы, связанные с проектным заданием, сроками, затратами, ресурсами, кадрами, информационными потоками, риском и материально-техническим снабжением (закупками).

Международный стандарт ISO 10006 ориентирован на проекты самого широкого спектра — малые и крупные, краткосрочные и долгосрочные, для различных окружающих условий и безотносительны к типу проектируемого продукта (включая технические средства, программное обеспечение, полуфабрикаты, услуги или их сочетание). Реализованные в нем рамочные требования требуют последующей адаптации данного руководства к конкретным условиям разработки и реализации отдельного проекта.

Стандарт заимствует ключевые определения из ISO 8402, включая такие термины, как проект, продукт проекта, план проекта, участник проекта, процесс, оценка хода работ. Для всех процессов управления проектом (планирование, организация, мониторинг и контроль) применяются процессы и задачи менеджмента качества.

В большой мере стандарт по содержанию основан на PMBoK 1996, совпадение имеется вплоть до названий областей знаний управления проектами. Это первый из стандартов ISO серии 9000, в котором применен процессный подход.

Профессиональные международные и национальные квалификационные стандарты

Компетентность менеджеров проектов и специалистов в области РМ определяется следующими компонентами (рис. 4):

- знания;
- опыт;
- умения и навыки;
- этика;
- профессиональный образ мышления;
- профессиональный образ действий, включая использование методов и средств РМ.

Требования, нормы и стандарты, которые позволяют говорить о профессиональной состоятельности менеджера проекта и качестве его работы по проекту, устанавливаются в разном виде для различных компонентов.

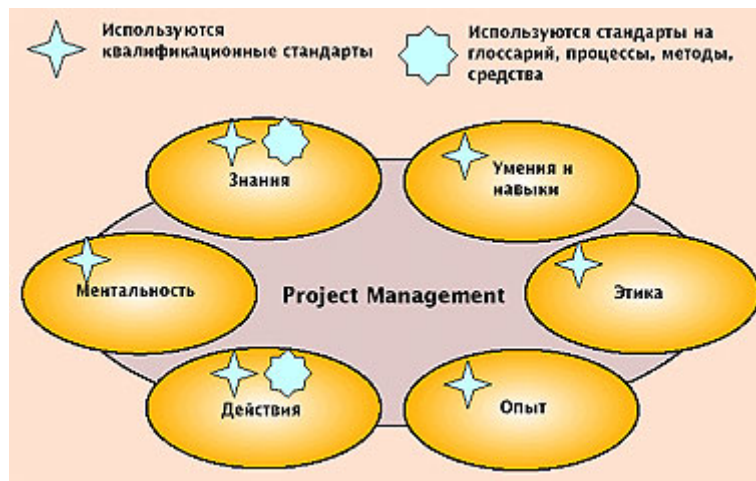


Рис. 4. Компоненты профессиональной компетентности менеджеров проектов и их нормирование посредством стандартов

Определение профессиональной компетентности происходит посредством сертификационных испытаний и в разных странах проводится по-разному. Скажем, международная сертификация IPMA предусматривает 4 уровня компетентности и проводится ассессорами, уполномоченными IPMA. Процедура испытаний длится от одного до трех дней, в зависимости от уровня притязаний кандидата, и предусматривает его обязательное личное участие. Таким же образом выстраиваются системы сертификации в странах, принявших стандарт IPMA в качестве базового. В Австралии AIRM предусматривает 7 уровней компетентности, и оценка проводится в несколько этапов. PMI (США) предусматривает один уровень компетентности, а экзамен проводится в течение нескольких часов одного дня. С 2000 года сертификационные испытания не требуют личного присутствия кандидата и осуществляются посредством дистанционной сдачи экзаменов через Internet в уполномоченной организации. Для допуска к экзамену надо пройти отбор на основании отправленных ранее документов; основной критерий отбора — наличие достаточного опыта профессиональной деятельности по РМ.

Ни одна из систем сертификационных испытаний не свободна от недостатков. Главное же их различие заключается в концептуальном подходе к проекту. При преобладании процессного подхода наиболее адекватная модель PMI, при главенстве системного подхода — модель AIRM, если же в основу положен «менеджерский» подход, целесообразно использование модели IPMA, APM, GPM и др.

IPMA ежегодно издает специальный сборник, в котором информирует о состоянии вопроса по сертификации, последних изменениях, приводит списки всех сертифицированных менеджеров проектов по международным и национальным стандартам, официальных международных и национальных ассессоров и проч.

Своды знаний

Требования к знаниям определяются «Сводами знаний» (Body of Knowledge). Они образуют систему требований к знаниям, опыту, мастерству менеджеров проектов и специалистов по РМ. Своды знаний поддерживаются и развиваются международными и национальными профессиональными ассоциациями. В настоящее время ассоциации более чем в 20 странах имеют официальные национальные Body of Knowledge on Project Management (PMBok) и национальные системы сертификации. Эти Своды знаний представлены в виде национальных систем требований к профессиональной компетентности или национальных стандартов по отдельным вопросам РМ.

В области РМ международным нормативным документом, определяющим систему международных требований к компетентности менеджеров проектов, является ICB IPMA. На его основе производится разработка национальных систем требований к компетентности специалистов в странах, являющихся членами IPMA. Национальные системы требований должны соответствовать ICB IPMA и официально утверждаться (ратифицироваться) соответствующими уполномоченными органами IPMA.

Ряд не входящих в IPMA стран (в том числе США, Австралия и Япония) имеет собственные Своды знаний и системы сертификации.

Международный Свод знаний

International Competence Baseline (ICB) — официальный международный Свод знаний в области РМ, который поддерживается и развивается IPMA. Для 32 стран-членов IPMA он является основой для разработки национальных Сводов знаний; в настоящее время утвержденные национальные Своды знаний, соответствующие ICB, имеют 16 стран.

ICB определяет области квалификации и компетентности в менеджменте проектов, а также принципы оценки кандидата на получение сертификата. ICB содержит 42 элемента (28 основных и 14 дополнительных), определяющих области требований к знаниям, мастерству и профессиональному опыту в менеджменте проектов.

ICB издан на английском, немецком и французском языках. Основой для него послужило несколько национальных разработок: Body of Knowledge of APM (Великобритания); Beurteilungsstruktur, VZPM (Швейцария); PM-Kanon, PM-ZERT/GPM (Германия); Criteres d'analyse, AFITEP (Франция).

Каждая входящая в IPMA национальная ассоциация ответственна за разработку и утверждение собственных Национальных требований по компетентности (National Competence Baseline — NCB) со ссылкой на ICB и в соответствии с ним, а также с учетом национальных особенностей и культуры. Национальные требования оцениваются

специальным Комитетом IPMA на соответствие ICB и основным критериям сертификации согласно стандарту EN 45013.

Национальные Сводоы знаний

ICB служит основой для разработки и использования в качестве национальных систем требований и стандартов национальных Сводоы знаний в странах, которые входят в IPMA. Национальные Сводоы знаний и процедуры сертификации имеются и в ряде стран, не являющихся членами IPMA, в частности, в США, Австралии, Южной Корее и в некоторых других странах. Из национальных стандартов наиболее распространенным документом в области PM, используемым специалистами многих стран, является PMI PMBoK Guide.

4.7.4. Стандарты управления проектами разработки программного обеспечения

Об управлении проектами в области разработки программного обеспечения прямо или косвенно говорят стандарты:

- ISO 12207 в части определения процессов жизненного цикла и их характеристик;
- серия стандартов ISO 9000:2000 в части управления качеством разработки;
- группа стандартов CMM/CMMI в части определения уровня зрелости организации-разработчика в отношении способности разработки качественного программного обеспечения;
- ISO/IEC15504 (SPICE) в части самооценки организации-разработчика.

Рассмотрим основные идеи, лежащие в основе этих стандартов.

Стандарт ISO 12207

Стандарт ISO 12207 в России принят как ГОСТ Р ИСО/МЭК 12207-99. Он разработан Всероссийским научно-исследовательским институтом стандартизации (ВНИИСтандарт) Госстандарта России; принят и введен в действие постановлением Госстандарта России от 23 декабря 1999 г. № 675-ст. Стандарт устанавливает, используя четко определенную терминологию, общую структуру процессов жизненного цикла программных средств, на которую можно ориентироваться в программной индустрии. Стандарт определяет процессы, работы и задачи, которые используются: при приобретении системы, содержащей программные средства, или отдельно поставляемого программного продукта; при оказании программной услуги, а также при поставке, разработке, эксплуатации и сопровождении программных продуктов.

Определения процессов, работ и задач носят описательный характер, указывается, что должно делаться, но не дается рекомендаций как это делать. Детальное решение вопросов реализации стандарта возлагается на организацию, использующую стандарт. На практике стандарт используется следующим образом: из общего перечня процессов, работ и задач выбираются те, которые адекватно соответствуют специфике конкретной организации. Выбранные элементы уточняются до уровня документированных процедур, и таким образом, регламентируются процессы, установленные в организации.

Группа стандартов ISO 9000

Серия стандартов ISO 9000:2000 устанавливает общие требования к системе менеджмента качества любой организации, желающей продемонстрировать свою способность стабильно давать продукцию, отвечающую требованиям потребителя и соответствующим нормативным требованиям, и способствующую повышению степени удовлетворенности потребителя. Стандарт основан на трех основополагающих идеях:

- ориентация на потребителя,
- процессный подход,
- постоянное улучшение.

Ориентация на потребителя означает, что качество (продукта, услуги) определяется как степень удовлетворенности потребителя. Вопросы организации взаимодействия с потребителем занимают важное место в стандарте.

Процессный подход подразумевает, что качественно работающая организация имеет установленные процессы, и что качество процесса предопределяет качество продукта. Требования стандарта определяют, что процессы в организации должны быть установлены (то есть надежно воспроизводиться, не быть случайными и рассчитанными на удачу) и должны быть определены (то есть должны быть детально документированы используемые операции, регламенты, формы документов и т.д.).

Принцип постоянного улучшения требует, чтобы все основные процессы были измеримыми и чтобы были установлены и определены процессы постоянного улучшения основных процессов.

Стандарт имеет развитую систему поддержки сертификации, то есть подтверждения соответствия стандарту. Сертификация производится независимыми специализированными органами по сертификации, имеющими международную признанную аккредитацию. Аудиторы органа по сертификации проверяют, что все процессы в организации установлены и документированы, и что процессы фактически выполняются в точном соответствии с тем, как это описано в документах организации. В обязанности аудиторов ни в коем случае не

входит определение того, какие процессы являются «правильными» для данной организации. Сертификация удостоверяет, что организация делает ровно то, что обещает, не более, но и не менее. Поэтому стандарты этой серии являются гибкими, динамичными и наиболее широко используемыми во всем мире.

Следует подчеркнуть, что применение стандартов серии ISO 9000 в организациях по разработке программного обеспечения является довольно трудоемким. Стандарты этой серии ориентированы на систему менеджмента качества любой организации в целом, и практически не касаются специальных вопросов управления разработкой программных продуктов. Организации надлежит самостоятельно детально проработать все вопросы разработки программного продукта в соответствии с буквой и духом стандарта. С другой стороны, сертификация по ISO 9000 действительно подтверждает не только то, что организация может однократно произвести качественный программный продукт, но и то, что организация постоянно прогрессирует, ориентирована не на себя, а на внешний мир и работает по строгим, прозрачным для внешнего наблюдателя правилам.

Группа стандартов CMM/CMMI

Группа стандартов CMM/CMMI была создана SEI (Software Engineering Institute), который финансируется за счет Министерства обороны США и является структурной единицей Университета Карнеги-Меллона. Первая официальная версия стандарта вышла в 1993 г., хотя работы над ним начались гораздо раньше — основные его положения были опубликованы еще в 1986 г. Основная идея стандарта состоит в использовании модели CMM (Capability Maturity Model — модель зрелости возможностей) для приписывания каждой организации определенного уровня, с тем, чтобы организации можно было бы сравнивать по уровням. Список уровней приведен в таблице 2.

Таблица 3.

Уровни модели CMM

<i>№ уровня</i>	<i>Название уровня</i>	<i>Ключевые области процесса</i>
1	Начальный	Если организация находится на этом уровне, то ключевых областей процессов для нее не предусмотрено
2	Повторяющийся	Управление программными конфигурациями. Обеспечение качества программных продуктов. Управление контрактами подрядчиков. Контроль за

		ходом проектов. Планирование программных проектов. Управление требованиями
3	Определенный	Экспертные оценки. Координация взаимодействий проектных групп. Инженерия программного продукта. Комплексный менеджмент ПО. Программа обучения персонала. Определение организационного процесса. Область действия организационного процесса
4	Управляемый	Менеджмент качества ПО. Управление процессом на основе количественных методов
5	Оптимизируемый	Управление изменением процесса. Управление технологическими изменениями. Предотвращение дефектов

Деление на уровни позволяет последовательно внедрять СММ, используя стандарт в качестве руководства, которое может обеспечить постоянное совершенствование процесса разработки.

Стандарт СММ оказался весьма успешным, и впоследствии на его основе была создана целая серия стандартов (таблица 3).

Таблица 4.

Развитие стандартов СММ

System Engineering CMM (SE-CMM)	Ориентирован на вопросы системного инжиниринга – разработку продуктов (анализ требований, проектирование систем продукта и их интеграция) и их производство (планирование производственных линий и функционирование)
Trusted CMM (T-CMM)	Предназначен для обслуживания чувствительных и закрытых программных систем, которые требуют гарантии высокого качества ПО
System Security Engineering CMM (SSE-CMM)	Сфокусирован на аспектах безопасности программной инженерии, обеспечивает безопасный процесс разработки, в том числе и безопасность членов команды создателей
People CMM (P-CMM)	Рассматривает вопросы развития персонала в софтверных организациях
Software Acquisition CMM (SA-CMM)	Охватывает вопросы приобретения программных продуктов у внешних организаций
Integrated Product	Служит средой для интеграции усилий по

Development CMM (IPD-CMM)	разработке на всех этапах жизненного цикла и со стороны каждого отдела компании
---------------------------	---

Однако практическое применение стандартов серии СММ показало, что они не обеспечивают безоговорочного успеха в разработке ПО. Эти стандарты не были хорошо согласованы между собой — одновременное внедрение различных модификаций СММ могло оказаться достаточно сложной задачей и приводило в недоумение специалистов организаций, которые с этим сталкивались.

Также существенная проблема СММ состоит в необходимости «выравнивания» всех процессов. Если организация пытается сертифицироваться на определенный уровень, то она должна обеспечить соответствующий уровень для всех своих процессов. Даже если специфика, методология или особенности разработки не располагают к выполнению определенных процессов, сертификация этого требует.

Еще одна проблема вызвана тем положением, которое заняли стандарты СММ в современной индустрии ПО. Поскольку организация, обладающая высоким уровнем в соответствии с СММ, должна обеспечивать более высокие показатели программных продуктов по сравнению с теми, кто сертифицирован на низших уровнях, то стандарт стал применяться в качестве критерия отбора для участия в тендерах на разработку ПО или в аутсорсинговых проектах. Спрос на сертифицированные организации породил предложение по «быстрой и безболезненной сертификации».

Подобная ситуация стала возможной благодаря недостаткам процесса сертификации. Сертификации по СММ подлежит не вся организация в целом, а только определенный проект. Ничто не мешает организации создать «образцово-показательный» проект, выполняемый с учетом всех требований высоких уровней стандарта СММ, получить соответствующий уровень сертификации и заявить о том, что все продукты отвечают такому-то уровню стандарта.

Разрешить большинство проблем СММ призван новый стандарт SEI — Capability Maturity Model Integrated (CMMI) — Интегрированная модель зрелости возможностей. Стандарт CMMI изначально создавался таким образом, чтобы объединить существующие варианты СММ и исключить какие-либо противоречия при его практическом применении в различных сферах деятельности высокотехнологичных компаний.

Для того чтобы устранить необходимость «выравнивания» процессов организации и быть более приспособленным к ее потребностям, стандарт CMMI имеет две формы представления — классическую, многоуровневую, соответствующую СММ, и новую, непрерывную. Непрерывная форма представления рассматривает не уровни зрелости (Maturity Levels), а уровни возможностей (Capability Levels), которые оцениваются для отдельных областей процессов. В

таблице 4 дано соответствие уровней зрелости стандарта CMM, а также уровней зрелости многоуровневого представления CMMI и уровней возможностей непрерывного представления CMMI.

Таблица 5.

Соответствие уровней CMM и CMMI

<i>№ уровня</i>	<i>Уровень зрелости CMM</i>	<i>Уровень зрелости многоуровневого представления CMMI</i>	<i>Уровень возможностей непрерывного представления CMMI</i>
0	–	–	Незавершенный
1	Начальный	Начальный	Выполнимый
2	Повторяющийся	Управляемый	Управляемый
3	Определенный	Определенный	Определенный
4	Управляемый	Управляемый количественно	Управляемый количественно
5	Оптимизируемый	Оптимизируемый	Оптимизируемый

Стандарт SPICE

Аббревиатура SPICE раскрывается как Software Process Improvement and Capability dEtermination. Основные цели SPICE:

- удовлетворение растущих потребностей в оценке возможностей процессов производства программного обеспечения в подразделениях;
- гармонизация методов и моделей, используемых для оценки процессов.

Проект SPICE был начат ISO в июле 1991 года и к настоящему времени объединил лучшие из существующих в мире практик. В основу SPICE легли такие стандарты, как:

- STD (Compita);
- CMM (SEI);
- TRILLIUM (Bell);
- SQPA (HP);
- BootStrap (Esprit);
- HealthCheck (BT);
- ISO 9001;
- SAM (BT).

В отличие от одномерной модели CMM архитектура SPICE двумерна и состоит из так называемых «уровней возможностей», их насчитывается 6 (плюс 9 атрибутов процессов и 32 правила менеджмента); категорий процессов (5) и типовых процессов (29), а

также наилучших известных практик (best known practices, их более 200). Центральной идеей SPICE является оценивание возможностей не одним числом, как в СММ, а путем построения так называемого *профиля организации*. Сама организация может проанализировать свою деятельность, выявить, какие из наилучших практик в ней применяются, а какие нет, и построить профиль — кривую, наглядно показывающую, в каких областях деятельности организация сильна, и где в ней узкие места. Наличие масштабируемого механизма самооценки является важнейшим достоинством SPICE. В таблице 5 проведено сравнение механизма самооценки SPICE и процедуры аудита в ISO 9000.

Таблица 6. Сравнение самооценки и аудита

Самооценка	Аудит
Детальные критерии	Абстрактные критерии
Внутреннее участие	Внешний, независимый
Постоянная	Краткий
Позитивное суждение	Негативное суждение
Поиск фактов	Поиск ошибок
Взаимодействие	Противостояние
Открытость	Защита
Общее обсуждение	Индивидуальные интервью

SPICE предлагает достаточно подробную модель, предоставляющую пользователям свободу в выборе путей к улучшению работы. Модель улучшения процессов в SPICE трехмерна, где по одной оси откладывается Эффективность работы (удовлетворенность заказчиков, качество продукции и продуктивность), по другой — Возможности персонала, и по третьей — Адекватность процесса. Таким образом, можно выбирать траекторию улучшения процесса в трехмерном пространстве, где улучшения по каждой из осей идут параллельно с улучшениями по другой. Собственно, параллельность не является требованием, это, скорее, рекомендация, позволяющая избежать серьезных перекосов в процессе производства.

Хотя, как уже говорилось, SPICE вобрал в себя все самое лучшее из целого ряда популярных стандартов, он не стал простым их объединением. Для того чтобы показать, чем же SPICE отличается от своих предшественников, целесообразно провести сопоставление SPICE и других наиболее известных стандартов.

Таблица 7.

Сравнение SPICE и ISO 9001

SPICE	ISO 9001
-------	----------

Развитая документация	Малая документация
Детальная модель	Абстрактная модель
Разработан для производства ПО	Разработан для производства в целом
Улучшение процессов и оценка возможностей	Сертификация
Требования к самооценке, руководство по применению	Не содержит
Дополняет ISO 9001	Дополняются SPICE

Таблица 8.

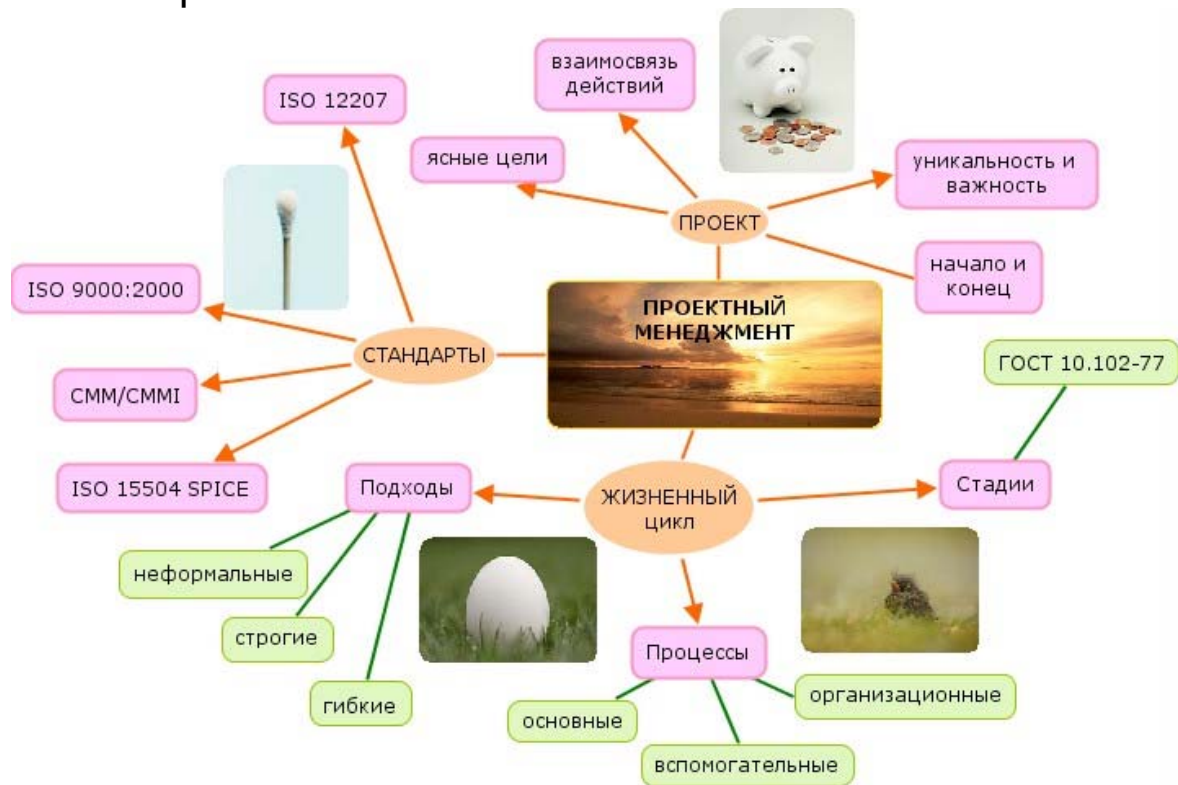
Сравнение SPICE и CMM

SPICE	CMM
Двумерная структура	Последовательная, одномерная структура
Допускает гибкость в выработке стратегии улучшения	Содержит predetermined путь развития
Уровни возможностей для каждого процесса	Единый уровень зрелости для всех процессов
Результаты требуют упрощения	Результаты легко понимаемы
Результаты очень подробные	Упрощенные результаты

4.8. Заключение

Менеджмент проектов по разработке программного продукта требует ясного осознания области эффективного применения, учета типа программного продукта, выбора подходящей модели жизненного цикла, знания соответствующих стандартов и проведения комплекса работ по внедрению.

4.9. «Карта памяти» по теме



4.10. Список использованной и рекомендованной литературы

1. Баранов С. Н. Управление программным проектом. Лекции по спецкурсу "Технология программирования". - СПб: Санкт-Петербургский государственный электротехнический университет, рукопись, 1998.
2. Брукс Ф. Мифический человек-месяц или как создаются программные системы. - СПб.: Символ-Плюс, 1999.
3. Эдвард Йордон. Путь камикадзе. Как разработчику программного обеспечения выжить в безнадежном проекте. - М.: ЛОРИ, 2001.
4. Conger, Sue A. The New Software Engineering. Wadsworth Publishing Company, 1994.
5. Pierre N. Robillard, Martin P. Robillard. Types of collaborative work in software engineering. // The Journal of Systems and Software 53, 2000, pp. 219-224.
6. Rob Thomsett. Effective Project Teams: A Dilemma, a Model, a Solution. American Programmer, July-August 1990.
7. Vliet H. V. Software Engineering: Principles and Practice. John Wiley and Sons, 2000.

Тема 5. Планирование проекта

5.1. Введение

Планирование – это первая функция менеджмента. Планирование проектов разработки программного продукта имеет как общие аспекты, присущие всем видам планирования, так и специальные особенности, характерные именно для процессов разработки программного продукта.

В этой теме рассматриваются основные понятия планирования проекта применительно к разработке программных продуктов.

Изучив учебный материал данной темы, Вы:

- узнаете или пополните свои знания о видах планов;
- узнаете или пополните свои знания о функциях, выполняемых менеджером в процессе планирования;
- сможете построить структуру декомпозиции работ для конкретного проекта;
- сможете спланировать организационную структуру для конкретного проекта.

В рамках темы рассматриваются следующие учебные вопросы:

- Фазы проекта и виды планов
- Определение целей
- Методика разработки и анализа планов
- Структура декомпозиции работ
- Разработка проектно-сметной документации
- Организационная структура исполнителей
- Прикладные программные средства для менеджера проекта

5.2. Предынвестиционная фаза проекта

Всякий проект по разработке программного обеспечения переживает особую фазу, которая называется ***предынвестиционной***, поскольку протекает до начала прямого финансирования проектных работ.

Прединвестиционная фаза проекта – это возникновение и исследование идеи проекта.

Прединвестиционная фаза содержит следующие действия:

1. Собственно ***возникновение и первичное исследование идеи***, носящее максимально творческий и неформальный характер.

2. **Детальное исследование идеи.** Выработка концепции. Постановка задачи. Создание «одностраничного описания проекта» и разработка его расширенной версии.
3. **Экспертиза идеи специалистами.** Принятие решения о начале процесса планирования.

5.2.1. Постановка задачи

Из идеи должно вырасти описание проекта. Все предложенное следует формализовать и документировать. Вклад в детальное исследование идеи могут делать любые заинтересованные люди. В результате должна быть выработана концепция создаваемого или развиваемого программного продукта. Необходимо прояснить цель проекта, сформулировать требования к проекту и его результатам и выявить ограничения, которые существуют для достижения этих целей и выполнения этих требований.

Одностраничное описание проекта включает несколько небольших разделов, кратко излагающих различные стороны предлагаемой идеи. Формально оно называется «одностраничным», подчеркивая краткость данной работы, но реально может занимать и две-три страницы.

5.2.2. Экспертиза идеи специалистами

Практически всегда перед принятием решения о запуске проекта проводится экспертиза идеи и проекта, который на ней основан.

Специалисты должны изучить и проанализировать идею (обычно на это отводится небольшой срок) и

- подтвердить или поправить все предположения, на которых базируется проект. На основании этих предположений будут делаться все дальнейшие построения.
- выявить и охарактеризовать все критические моменты в проекте. Специалисты должны указать, что не предусмотрено и к каким последствиям это может привести.

Завершиться процесс возникновения и исследования идеи должен **принятием решения о начале работы над проектом**. Как минимум, это означает, что будет исполнен процесс планирования. Решение о начале работы может приниматься совместно заказчиком и исполнителем (в случае наличия конкретного заказчика) или управляющим комитетом компании (если продукт предназначен для широкого использования). Вопрос финансирования проекта (полного или частичного) может рассматриваться уже здесь, но чаще он принимается при завершении планирования.

5.3. Планирование проекта: определение целей проекта

Зачем вообще нужен план проекта? Вот ответ на этот вопрос.

- ✓ План помогает создать ясное и четкое понимание – как будущие работы будут выполняться.
- ✓ План определяет роль каждого человека в исполнении проекта.
- ✓ План увязывает части работ вместе. План позволяет видеть все взаимозависимости между разными частями работы.
- ✓ План – это точка отсчета для любых последующих изменений.
- ✓ С помощью плана мы видим, когда цель, поставленная перед проектом, достигнута и проект заканчивается.

Заметим, что первое действие процесса планирования – это принятие решения о начале планирования.

5.3.1. Планирование целей

При постановке целей проекта по разработке программного продукта очень важно различать цель – создание программного обеспечения и цель, ради достижения которой создается программное обеспечение.

В очень редких случаях программное обеспечение создается «ради себя самого». В качестве исключительных примеров можно указать только учебные проекты, когда программное обеспечение разрабатывается ради демонстрации возможностей разработчиков или же академические проекты, когда программное обеспечение разрабатывается ради доказательства принципиальной возможности разработки.

В подавляющем большинстве случаев программные продукты разрабатываются с целью последующего использования. Очень важно не потерять из виду основную цель разработки, которая формулируется заказчиками и пользователями программного продукта и не подменить эту основную цель собственными целями разработчиков программного продукта.

Современное определение качества гласит, что качество – это степень удовлетворенности потребителей. Если цели проекта не отвечают потребностям потребителей, то программный продукт невозможно будет признать качественным, даже если при его разработке были использованы самые современные технологии и были задействованы самые квалифицированные разработчики.

Качество – это степень удовлетворенности потребителя.

5.3.2. Описание целей проекта

Цели проекта должны быть описаны явно и удовлетворять определенным критериям.

- **Конкретность.** Цель проекта должна быть конкретной, выраженной в терминах проекта, с указанием условий и сроков. Например, цель «добиться максимальной степени удовлетворенности потребителей продукта» не конкретна. В то же время цель «получить положительный отзыв на продукт от 90% потребителей за первые полгода продаж» достаточно конкретна для проекта по разработке «коробочного» продукта.
- **Реалистичность.** Цель должна быть реалистичной, то есть достижимой с учетом имеющихся ресурсов. Например, цель «разработать программу автоматического художественного перевода поэзии с английского языка на русский за год» нереалистична для компании среднего размера. В то же время цель «разработать программу подстрочного перевода технического текста с английского языка на русский за год» может быть вполне реалистична, если есть опыт и задел.
- **Измеримость.** Должен быть указан эффективно проверяемый критерий, который позволял бы определить, достигнута ли цель и в какой степени. Например, цель «разработать программу, существенно повышающую эффективность процесса продаж» неизмерима: неясно, как измеряется эффективность и какое повышение можно считать существенным. В то же время цель «разработать программу, ускоряющую процесс оформления заявки в среднем на 10%» измерима.
- **Непротиворечивость.** Цели не должны быть внутренне противоречивыми и взаимно исключаящими друг друга. Например, цели «разработать продукт с максимальным набором функций» и «затратить минимальное количество ресурсов на разработку» противоречат друг другу (и неконкретны, к тому же). Минимальное количество ресурсов (ноль) будет затрачено, если разработку не проводить, но тогда и возможностей у продукта не будет. В то же время цель «разработать продукт, имеющий не меньше функций, чем конкурент X и затратить на разработку не более Y человеко-часов» не является противоречивой (но может оказаться нереалистичной).

5.3.3. Подтверждение целей проекта

Сформулированные цели проекта необходимо подтвердить, то есть заручиться явным согласием с целями всех сторон, заинтересованных в успехе проекта. Сторонами, заинтересованными в успехе проекта, могут быть:

- разработчики, которые выполняют проект;
- инвесторы, которые его финансируют;
- пользователи, которые будут применять результаты на практике.

Заинтересованные стороны проекта – все лица и организации, прямо или косвенно участвующие в проекте и заинтересованные в его успехе.

Ясно, что цели, преследуемые сторонами, могут быть весьма различны. Например, разработчики могут преследовать цель освоения новой технологии при выполнении проекта. Инвесторы, скорее всего, заинтересованы в возврате средств и получении прибыли. Пользователи надеются на то, что разрабатываемое программное обеспечение сделает их труд более производительным. Подтверждение целей проекта – это поиск компромисса, который удовлетворял бы все заинтересованные стороны.

5.3.4. Контроль изменения целей

В процессе выполнения проекта его цели могут измениться. Причины изменения целей многообразны и вряд ли могут быть учтены заранее. Например, причинами изменения целей могут быть:

- изменения в бизнес-процессе, для автоматизации которого разрабатывается программное обеспечение;
- изменение конъюнктуры рынка, в частности, появление нового конкурирующего продукта;
- появление новой информационной технологии, влияющей на целевую предметную область.

В процессе выполнения проекта постоянно должен проводиться мониторинг целей для проверки их актуальности. В случае необходимости цели следует пересматривать. Иногда пересмотр целей влечет пересмотр плана всего проекта.

5.4. Виды планов

При проведении проектов по разработке программного продукта используются несколько видов планов. Планы различаются по планируемому периоду и аспекту процесса разработки.

По планируемому периоду различают планы:

- ◎ **Стратегические.** Стратегический план охватывает длительный период, и обычно относится ко всей организации в целом, а только к отдельному проекту.
- ◎ **Текущие.** Текущие планы относятся к текущему выполняемому проекту.
- ◎ **Оперативные.** Оперативные планы могут быть несколько раз созданы и выполнены на протяжении одного проекта. Например, при итеративном процессе разработки план отдельной итерации является оперативным.

Процесс разработки программного обеспечения имеет несколько аспектов, каждый из которых требует планирования. Каждому такому аспекту также может соответствовать свой тип плана.

Наиболее удобной является классификация планов в соответствии с согласованным множеством *стандартов ведения документации, разработанным IEEE*. Несмотря на то, что стандарты IEEE иногда требуют некоторой модернизации, они помогают справляться с большинством проблем, возникающих с документацией.

Ниже приводится краткое описание каждого плана из набора IEEE:

- ◎ **SVVP (Software verification and validation plan) – План экспертизы программного обеспечения.**

Этот план определяет, каким образом и в какой последовательности должны проверяться стадии проекта, а также сам продукт на соответствие поставленным требованиям. В терминах управления качеством такая проверка подразумевает верификацию и валидацию. Верификация – это процесс проверки на соответствие требованиям процесса разработки программного продукта; валидация проверяет тот факт, что разработан требуемый продукт. Зачастую валидацию и верификацию осуществляют сторонние организации, в этом случае экспертиза называется *независимой (Independent V&V, IV&V)*.

Верификация – проверка соответствия установленным требованиям процесса разработки продукта.

Валидация – проверка соответствия продукта установленным требованиям в условиях конкретного применения.

◎ **SQAP (Software quality assurance plan) – План контроля качества программного обеспечения.**

Этот план определяет, каким образом проект должен достигнуть соответствия установленному уровню качества. Например, в этот план включают такие мероприятия, как инспекция программного кода, тестирование и т.п.

◎ **SCMP (Software configuration management plan) – План управления конфигурацией программного обеспечения.**

Этот план определяет, как и где должны храниться документы, программный код и их версии, а также устанавливает их взаимное соответствие. Было бы крайне неразумным начинать работу без такого плана, так как самый первый созданный документ обречен на изменения, а мы должны знать, как управлять этими изменениями до того, как мы начнем составлять документ.

◎ **SPMP (Software project management plan): План управления программным проектом, или, в более сжатой форме, план проекта.**

Этот план определяет, каким образом управлять проектом. Обычно он соответствует известному процессу разработки, например стандартному процессу компании. Именно этот план является тем главным документом, посредством которого менеджер проекта ведет управление проектом. Другие виды планов либо включаются в план проекта в качестве разделов, либо план проекта содержит ссылки на них.

5.5. Методика разработки и анализа плана проекта

Планирование проекта включает **определение ресурсов** – человеческих, вычислительных и организационных и составление «карты» задач и времен их выполнения.

В стандартном наборе процессов планирование является одним из действий процесса управления.

Общий подход к планированию выглядит следующим образом.

- ◎ Построение списка задач.
- ◎ Составление графиков выполнения работ.
- ◎ Выделение требуемых ресурсов.
- ◎ Распределение ответственности.
- ◎ Определение зависимостей между задачами.

- ⊙ Проведение персональных назначений на задачи.
- ⊙ Определение времени выполнения задачи.
- ⊙ Оценка рисков, связанных с конкретными задачами.
- ⊙ Выявление критических путей.
- ⊙ Создание инфраструктуры управления.
- ⊙ Оценка затрат.

Следует отметить, что перечисленные этапы планирования не выполняются строго последовательно. Процесс планирования принципиально итеративен, но и внутри каждой итерации планов могут проводиться различные итерации последовательности этапов. Например, оценка затрат может начинаться на самом раннем этапе (и даже до начала собственно планирования), когда речь идет о грубых оценках, а затем уточняться по окончании последующих этапов формирования плана, причем в самом конце формируется уже подробная смета проекта. Однако всегда построение списка задач предшествует распределению ресурсов и назначению персонала.

В рамках планирования управляющий решает также следующие организационные задачи.

- Анализ документации на полноту, содержание, аккуратность.
- Назначение начальной и конечной дат работ.
- Определение интерфейсов приложения и планирование работ по детальному проектированию интерфейсов.

5.5.1. Выявление и анализ критического пути

Очень важным является анализ критического пути в плане. Понятие «критический путь» определяется на основе анализа зависимостей между работами.

Работы могут быть независимыми, а могут быть связаны так, что одна работа должна обязательно предшествовать другой. Например, такая зависимость возникает, если результаты одной работы используются как исходные данные для другой. Зависимости между работами удобно изображать в виде ориентированного графа: узлы графа соответствуют работам, и для каждой работы проводятся дуги от всех предшествующих работ. Для удобства вводятся два узла, начальный и заключительный, соответствующие началу и окончанию проекта.

Любая работа не может начаться раньше, чем будут выполнены все предшествующие ей работы. Продолжительность выполнения всего проекта не может быть меньше, чем сумма продолжительностей работ, входящих в любой путь от начального до заключительного узла. Длиной пути назовем сумму продолжительностей работ, входящих в путь.

Критическим путем будем называть путь наибольшей длины. Наибольший риск в выполнении проекта будет иметь задержка в работах на критическом пути.

Критический путь – это путь наибольшей длины в графе работ.

5.5.2. Методики оценок времени и затрат

Существует несколько методик оценок времени и затрат (трудоемкости, стоимости) для составления планов и расчетов. Перечислим основные из них:

- ✓ **Сверху вниз по крупным блокам аналогичного проекта.** Поскольку аналогичный проект пройден и мы знаем реальные затраты для него, то для нового проекта берем соответствующие данные с некоторыми поправками.
- ✓ **Снизу вверх по составленному графику работ.** Данные по трудозатратам и времени берем от исполнителей.
- ✓ **Методика норм работы.** Учет ведется по нормам, определяемым как среднее значение времени, уходящее на данную работу в данной компании. Например, время, уходящее на разработку одной строки программного кода.
- ✓ **Методика исторических соотношений.** Стоимость (трудоемкость, время) одного проекта относится к стоимости (трудоемкости, времени) другого пропорционально отношению их объемов в некоторой степени n . Степень n – число, которое надо знать или подобрать опытным путем.
- ✓ **Методика приближенных вычислений.** Существует несколько расчетных формул для времен (трудоемкостей, стоимостей) выполнения работ проекта. Введем три оценочных величины — оптимистическую (О), реальную (Р) и пессимистическую (П). Тогда среднее оценочное время (трудоемкость, стоимость) задач проекта можно вычислить по:
 - ◆ формуле трапеций: $(O+2P+P)/4$;
 - ◆ формуле Симпсона: $(O+4P+P)/6$.

5.5.3. Распределение работ

Распределение работ включает определение уровня квалификации для исполнителей задач, составление списка потенциальных участников проекта и «отображение» исполнителей на задачи. Типичной проблемой может быть сопоставление имеющимся рабочим местам тех потенциальных исполнителей, чей уровень квалификации не полностью

соответствует требуемому. Существует несколько эвристических правил для персональных назначений:

- ♦ Назначать опытных и квалифицированных людей на наиболее сложные задачи в критическом пути. Людей с меньшим количеством опыта – на менее сложные задачи и т.д.
- ♦ Определить последовательность работ для каждого исполнителя.
- ♦ Не перегружать исполнителей задачами, для исполнения которых потребуется больше времени, чем есть в реальности.
- ♦ Никогда не следует стремиться «загрузить исполнителей до предела». Точный предел никогда не известен, а менеджеры склонны его завышать. В реальности такой подход приведет к чрезмерной загрузке исполнителей. Следует всегда планировать некоторый разумный резерв загрузки.

5.6. Структура декомпозиции работ (WBS)

Структура декомпозиции работ – WBS (Work Breakdown Structure) или СДР – это широко распространенный, сравнительно простой и проверенный на практике метод составления списка работ, который может послужить основой дальнейшего планирования.

5.6.1. Основные термины и определения

Используемые в названии данного метода термины имеют следующие значения:

- Work (Работа) – непрерывное физическое или умственное усилие, направленное на преодоление препятствий и достижение целей или результатов; специфическая задача, обязанность, функция или задание, часто являющиеся частью фазы или другой, большей по объему работы; что-то, производимое или выполняемое в результате усилия или применения навыков (квалификации).
- Breakdown (Декомпозиция) – разделение на части или категории, выделение простых составляющих.
- Structure (Структура) – фиксированное упорядоченное множество объектов и отношений между ними, классификация чего-либо по заданному основанию.

Эти определения означают, что структура декомпозиции работ (WBS) имеет следующие характеристики:

- описывает с необходимой точностью содержание работ по проекту;
- определяет весь объем работ по проекту;

- формируется в виде иерархической структуры (проект декомпозируется на пакеты/субпакеты и т.д. вплоть до отдельных работ).

WBS является средством для разделения всех работ по проекту на управляемые, определяемые пакеты работ, позволяющие достичь уровня детализации предоставляемой информации, соответствующего потребностям руководства проекта в контроле. WBS позволяет определить работу по проекту с точки зрения жизненного цикла проекта.

WBS обеспечивает выявление работ, необходимых для достижения целей проекта. При таком подходе проект определяется в терминах иерархически взаимосвязанных ориентированных на результат элементов (пакетов, или комплексов работ, сгруппированных по заданным основаниям/критериям). Каждый следующий уровень декомпозиции обеспечивает последовательную детализацию содержания проекта, что позволяет производить оценку выполненных объемов работ, освоенных денег и выполнения по срокам. На нижних уровнях пакетам работ соответствуют сравнительно меньшие объемы работ. Это упрощает оценку процента выполнения и дает возможность более четко определять действия, необходимые для достижения целей проекта. Предложенный подход декомпозиции работ формирует необходимую основу для определения измеримых показателей (трудоемкости, стоимости), а также позволяет с высокой степенью достоверности говорить о том, что цели, связанные с данным пакетом работ могут и будут достигнуты.

WBS является одним из основных инструментов (средств) в механизме управления проектом, с помощью которого измеряется степень (в абсолютной или относительной шкале) достижения результатов проекта, представляется информация на соответствующие уровни детализации, в формате и структуре, доступный и принятой теми, кто выполняет и контролирует работы.

С помощью WBS можно связать определенный объем работ с элементом организационной структуры, субподрядчиками или отдельными исполнителями. Как только работы и область ответственности определяются, отдельные исполнители (включая субподрядчиков) назначаются ответственными за выполнение определенных элементов WBS в рамках назначенных бюджетов и определенных сроков выполнения.

Для проектов средней сложности рекомендуется использовать до 6 уровней WBS: 3 верхние уровня для предоставления информации уровня заказчика, 3 нижние уровня для детализации информации уровня исполнителя. Глубина детализации WBS зависит от размера и сложности проекта, поскольку должна обеспечивать четкую формализацию целей и результатов работы, которые необходимо

выполнить. Каждый пакет работ включает весь объем работ, выполняемый основной организацией, ответственной за данный пакет работ, так же, как и организациями, с которыми заключены подрядные договора.

Подводя итог, констатируем, что WBS:

- ♦ определяет иерархию результатов;
- ♦ описывает весь объем работ, необходимый для достижения конечных целей или результатов проекта;
- ♦ разрабатывается путем деления результатов на поддающиеся учету и измеряемые элементы;
- ♦ является механизмом для объединения и оценки выполнения, выполнения проекта по объему работ и стоимости.

WBS – это способ описания целей и задач проекта путем его декомпозиции в терминах иерархически взаимосвязанных результатов и пакетов работ, выполнение которых необходимо для реализации проекта.

5.6.2. Необходимость использования WBS

Стандарт PMBOK (Project Management Body of Knowledge) описывает WBS, как средство (инструмент) определения содержания проекта. Он определяет управление содержанием проекта как «процесс, направленный на гарантированное обеспечение того, что проект включает все необходимые работы, и только те работы, которые необходимы для успешного завершения проекта». Основываясь на этом определении, разработка WBS имеет две основные цели:

- ✓ обеспечение планирования всех необходимых работ проекта;
- ✓ обеспечение отсутствия ненужных (лишних) работ, работ, не связанных с реализацией проекта.

Для руководителя проекта важны обе эти цели. Если в плане отсутствуют все необходимые работы, проект будет задержан, бюджет, скорее всего, будет превышен. Если выполняются работы, не относящиеся к данному проекту – деньги заказчика тратятся не целевым образом. Если WBS не объединяет обе эти цели, проект может потерпеть неудачу.

Успешное управление проектом зависит от способностей руководителя проекта эффективно руководить командой проекта,

определяя содержание работ проекта в терминах его результатов. С помощью WBS работы структурируются и непосредственно связываются с графиком, а ресурсы распределяются и отслеживаются.

5.6.3. Разработка структуры декомпозиции работ

При разработке WBS необходимо принимать во внимание следующие основные правила.

- + **Результативность.** Каждый элемент WBS должен обеспечивать достижение осязаемого результата.
- + **Агрегация.** Каждый элемент WBS должен являться агрегатом всех подчиненных элементов, перечисленных непосредственно под ним.
- + **Логичность.** Результаты должны логически декомпонироваться до уровня, на котором можно определить, как они будут достигаться (проектирование, поставки, заключение договоров, производство). Декомпозиция результатов, начиная от верхнего уровня WBS (проекта) до нижнего уровня должно быть логически связана.
- + **Уникальность.** Результаты пакетов работ должны быть уникальными и отличаться от результатов других пакетов работ того же уровня. Они должны декомпонироваться до уровня детализации, обеспечивающей успешное планирование, координацию и контроль работ, связанных с достижением поставленных целей.
- + **Гибкость.** Процесс разработки WBS должен представлять собой гибкий механизм, позволяющий корректировать WBS, особенно когда объем работ по проекту может изменяться. Однако, для успешного управления проектом, необходимо тщательно обеспечить процесс контроля изменений для документирования и управления изменениями содержания проекта. При изменении содержания проекта WBS должна быть откорректирована.
- + **Согласованность.** Каждый элемент WBS (пакет работ), представляющий собой объем работ субподрядчика или других внешних организаций, должен быть согласован непосредственно с соответствующими элементами WBS подрядчика.
- + **Полнота.** Все ожидаемые результаты проекта в явном виде должны быть включены в WBS.
- + **Отчетность.** Для всех важных событий, связанных с отчетностью (например, ежемесячные отчеты, отчеты о

проведении испытаний и т.д.) должны быть включены и определены соответствующие пакеты работ.

- + **Совместимость.** Все пакеты работ должны быть совместимы с организационной структурой и структурой затрат.

При разработке WBS следует принимать во внимание определенные сложности этого метода и стараться избегать типичных ошибок. Некоторые сложности перечислены ниже.

- Нахождение баланса между задачами описания содержания проекта и требованиями к сбору фактической информации и отчетности. (Напомним, что основная функция WBS — это определение объема работ). WBS — это инструмент, позволяющий руководителю проекта декомпозировать проект до уровня, необходимого и достаточного для определения требований проекта. Чрезмерная детализация WBS требует излишнего уровня контроля и отчетности. Недостаточная детализация WBS ведет к потере управляемости проектом.
- Недостаточное внимание к разработке WBS и переход непосредственно к формированию плана-графика (диаграммы Ганта, расчету критического пути или сетевого графика). Это может привести к потере важных для проекта работ, и, следовательно, к задержкам проекта на поздних стадиях его реализации после выявления упущений.
- Разработка элементов WBS, учитывающих только стадии проекта, либо организационную структуру без учета промежуточных результатов проекта (объектов, продуктов, артефактов), может привести к перерасходу по проекту, поскольку при таком подходе трудно оценить плановые показатели и проконтролировать выполнение проекта.
- Упущение определения пакетов работ, описывающих начальные и завершающие элементы, такие как планирование, опытную эксплуатацию и сопровождение. В этом случае объем работ может быть оценен неверно.
- Назначение пакетов работ с несколькими ответственными за создание тех или иных результатов (артефактов) — «у семи нянек дитя без глаза».
- Упущение вспомогательных результатов управления проектами, таких как управление процессами, услуги, информационное обеспечение и коммуникации, регламенты, обучение и инструментальное программное обеспечение.

5.6.4. Определение приемлемого уровня детализации

Разработка WBS является итерационным процессом разбиения проекта на составные элементы с выделением последующих уровней до тех пор, пока не будет достигнут уровень, обеспечивающий необходимую и достаточную детализацию информации для эффективного управления.

Ниже приведены вопросы для определения необходимости в дальнейшей детализации WBS. Если ответы на большинство пунктов в данном опросном листе являются положительными, необходима дальнейшая декомпозиция WBS.

Чем больше количество положительных ответов, тем более обоснованным является дальнейшая детализация WBS.

Опросный лист: нужно ли дальше детализировать WBS?

№	Вопрос	Ответ	
1.	Есть ли необходимость в повышении точности оценки стоимости и длительности по пакету работ?	Да	Нет
2.	Для пакета работ определен более чем один ответственный? Для выполнения работ в рамках пакета могут использоваться различные ресурсы, однако должен быть назначен только один ответственный за каждый пакет работ.	Да	Нет
3.	Объем работ, выполняемый в рамках данного пакета, описывает больше, чем один тип процесса или больше, чем один результат (артефакт) проекта?	Да	Нет
4.	Есть ли необходимость в раздельном определении стоимости процессов или результатов, описанных в данном пакете работ?	Да	Нет
5.	Есть ли зависимость между частью работ внутри пакета работ и другими внешними пакетами?	Да	Нет
6.	Наблюдаются ли существенные перерывы в выполнении работ в рамках пакета?	Да	Нет
7.	Меняются ли требования к ресурсам в течение времени в рамках выполнения пакета работ?	Да	Нет
8.	Различаются ли исходные условия для работ внутри пакета работ?	Да	Нет
9.	Существуют ли четкие, объективные критерии измерения выполнения для пакета работ?	Да	Нет
10.	Существуют ли специфические риски, связанные с частью пакета работ и требующие дальнейшей детализации пакета для выделения этих рисков?	Да	Нет

11.	Может ли для части пакета работ отдельно пересчитываться расписание?	Да	Нет
-----	--	----	-----

Как было определено ранее, уровень детализации WBS зависит от размера проекта и баланса между сложностью, риском и требованиями руководителя проекта к контролю проекта. Уровень детализации может также изменяться в процессе жизненного цикла проекта.

Для краткосрочных проектов на начальной стадии можно разработать всю WBS до достаточного уровня детализации, в то время как долгосрочные проекты и проекты с высоким уровнем сложности могут не декомпонироваться полностью на начальной стадии. Полностью WBS для таких проектов можно описать в процессе их реализации. С другой стороны, это может означать, что для конкретного проекта, отдельные пакеты работ могут иметь различные уровни детализации. В частности, это верно при разработке «развертываемых» проектов, когда план детализируется для работ, которые должны непосредственно начаться, а работы будущих периодов определяются укрупнено, на верхнем уровне, до тех пор, пока на более поздней стадии жизненного цикла проекта можно будет прописать их более детально. Именно так проводятся проекты по разработке программного обеспечения при использовании итерационных моделей разработки.

5.6.5. Взаимосвязь между риском проекта и WBS

Для проектов с высоким уровнем риска настоятельно рекомендуется разработка более детальной структуры WBS. Рисковые случаи – ситуации, которые могут повлиять на достижение результатов проекта – необходимо оценивать для определения и квалификации рисков.

Риски проектов связаны с вероятностью возникновения событий, позитивно или неблагоприятно влияющих на цели проекта, включающие основные элементы такие, как технические характеристики, качество, стоимость и сроки реализации. Подход к декомпозиции WBS может помочь в определении и уменьшении рисков. Например, проекты, требующие получения разрешительных документов и лицензий от надзорных органов, могут иметь высокую степень риска. Так как рисковая ситуация может возникать не для всего проекта в целом, а только для некоторых пакетов работ, для руководителя проекта удобнее анализировать ее влияние на каждый пакет работ, обособляя таким образом риски, обеспечивая их обработку, что, в конечном счете, обеспечивает более эффективное управление рисками.

Первый шаг при использовании такого метода – анализ каждого из пакетов работ до уровня, на котором можно выделить рисковое событие. Такой анализ должен учитывать критические области (проектирование и конструкторские работы, технологию, логистику и т.д.) и элементы, которые могут помочь в описании рисковых событий. Используя информацию из различных источников таких, как предыдущие оценки рисков, анализ экспертов и тому подобное, обследуются рисковые случаи и определяются характерные риски для каждой критической области. Затем они анализируются для определения вероятности их наступления, степени влияния и взаимозависимости.

Риск, связанный с объемом работ, может также определить необходимый уровень детализации. Дополнительная детализация пакета работ с высоким уровнем риска, обеспечивает лучшую оценку рисковой ситуации, а также более точную оценку стоимости и сроков. Это вынужденное структурирование позволяет определить предполагаемые и ожидаемые показатели на контролируемом уровне.

5.6.6. Взаимосвязь планирования и контроля ресурсов и WBS

WBS декомпозируется до уровня, необходимого для планирования и контроля. В общем случае это будет детализация до уровня, по крайней мере, на один уровень ниже уровня требуемого для отчетности. Такая детализация позволяет осуществлять эффективное планирование, контроль и измерение выполнения отдельных работ с однозначно определяемыми ресурсами.

Чтобы соответствующим образом подготовиться к планированию ресурсов в соответствии с WBS, необходимо рассмотреть следующие вопросы при определении уровня детализации WBS.

- ✓ Все ли работы запланированы с достаточной степенью детализации, необходимой для формирования и соблюдения обязательств?
- ✓ Существует ли возможность установления и контроля индивидуальных назначений ресурсов на работы со структурой отчетности, определенной данной WBS?
- ✓ Можно ли определить назначения на работы при постепенном расширении WBS? Будут ли они обоснованы как при разворачивании WBS сверху-вниз, так и при сборе данных снизу-вверх?
- ✓ Будут ли назначения ресурсов на работы согласованы с формальной системой расчета расписания?
- ✓ Как будут распределяться бюджеты?
- ✓ Можно ли будет связать бюджет с предполагаемым увеличением работы?

- ✓ Можно ли измерить увеличение объема работы на приемлемом уровне (т.е. соответствует ли уровень детализации WBS эффективному планированию и контролю)?
- ✓ Можно ли логически собрать данные по индивидуальным рабочим заданиям (т.е. можно ли работы, определенные в WBS, сгруппировать логически)?
- ✓ Как будет определяться состояние работ в процессе выполнения проекта?

5.6.7. Разработка WBS

WBS разрабатывается путем итерационного рассмотрения целей и результатов проекта, объема работ, реализации технических требований и других атрибутов. Верхние уровни WBS могут быть разработаны на ранней, концептуальной стадии проекта. Дальнейшая детализация WBS возможна, как только будут определены цели проекта и подготовлены спецификации.

Основной процесс разработки WBS состоит из следующих шагов:

- 1. *Определение конечных результатов проекта*** – что должно быть произведено (поставлено) для обеспечения успешного завершения проекта.
- 2. *Определение основных пакетов работ, необходимых для получения конечных результатов (продуктов) проекта.*** Часто такие основные пакеты работ соответствуют целям проекта, но вместе с тем, сами по себе они не являются целями проекта (например, технические требования к разработке программного обеспечения).
- 3. *Объединение дополнительных уровней детализации в соответствии с внутренней системой управления и единой системой контроля.*** Такие элементы обычно связаны с четким и раздельным определением отдельных результатов (продуктов) проекта.
- 4. *Пересмотр (анализ) и усовершенствование WBS*** до тех пор, пока все заинтересованные стороны проекта не будут согласны, что планирование проекта может быть успешно завершено, и можно будет успешно управлять, контролировать и регулировать получаемые результаты.

Как только закончена разработка WBS, можно приступать к разработке календарных планов, назначению ресурсов и созданию проектно-сметной документации.

5.7. Разработка проектно–сметной документации

Проектно-сметная документация – нормативно установленный комплекс документов, обосновывающих целесообразность и реализуемость проекта, раскрывающих его сущность, позволяющих осуществить проект.

5.7.1. Материально-техническая подготовка проекта

Всякий проект требует материально-технической подготовки, и проект по разработке программного продукта не является исключением. Очень часто менеджеры проектов по разработке упускают материально-техническую подготовку из виду, поскольку им кажется, что для проведения проекта не понадобится ничего, кроме персональных компьютеров, а они и так всегда под рукой. Такое заблуждение частично обосновано тем, что проекты разных типов требуют различного материально-технического обеспечения и невозможно дать общие рекомендации на все случаи жизни. Менеджер проекта должен внимательно проанализировать конкретную ситуацию и предусмотреть в плане проекта все необходимые материально-технические ресурсы.

Ниже приведены несколько типовых элементов материально-технического обеспечения проекта по разработке программного продукта и указаны некоторые вопросы, от ответов на которые может серьезно зависеть материально-техническая подготовка проекта.

- **Компьютеры.** Персональный компьютер является основным инструментом разработчика. Сейчас общепринятой является практика один разработчик — один компьютер. В большинстве случаев все разработчики в проекте используют компьютеры близкой конфигурации. При этом следует принять во внимание следующие вопросы.
 - Является ли переменным число разработчиков на протяжении всего проекта? Если да, то предусмотрены ли способы оперативного изменения числа компьютеров, задействованных в проекте?
 - Зависит ли разрабатываемый продукт от аппаратной платформы? Если да, то будет ли применяться кросс-платформенная разработка или же во время разработки будут использоваться специальные компьютеры?

- Используются ли разрабатываемым продуктом специальные устройства (например, биометрические)? Если да, то будет ли проводиться их программная эмуляция или же во время разработки будут использоваться реальные устройства?
- Являются ли заявленные минимальные требования продукта к конфигурации аппаратных средств более низкими, чем типовая конфигурация компьютера тестера? Если да, то обеспечены ли тестеры компьютерами минимальной и типовой заявленной конфигурации?
- Являются ли заявленные максимальные требования продукта к производительности/масштабируемости более высокими, чем позволяет реализовать типовая конфигурация компьютера тестера? Если да, то обеспечены ли тестеры компьютерами для нагрузочного тестирования?
- **Инфраструктура.** Разработка программного обеспечения — это офисная, конторская работа, для которой достаточно стола с компьютером и относительной тишины. При этом следует принять во внимание следующие вопросы.
 - Является разработка распределенной? Если да, то как обеспечивается оперативная и защищенная связь между разработчиками (Интернет, телефонные конференции и т.д.)?
 - Есть ли необходимость в использовании нестандартного режима рабочего времени (например, общение с заказчиком за океаном)? Если да, то обеспечены ли необходимые рабочие условия для разработчиков (транспорт, горячее питание)?
 - Предусматривает ли процесс проведение коллективных действий (производственных совещаний, мозговых штурмов и т.д.)? Если да, то имеется ли подходящее помещение для этого?
 - Используют ли разработчики печатные материалы (книги, документы, инструкции)? Если, то где хранятся эти материалы?

5.7.2. Типовая смета расходов

Составление финансовых документов обычно не входит в сферу ответственности менеджеров проектов по разработке программных продуктов и не включается в план проекта. Тем не менее, при планировании проекта невозможно полностью отстраниться от

финансовых вопросов. Менеджер проекта должен понимать смысл основных статей сметы расходов проекта и уметь оценивать влияние, которые эти расходы оказывают на ход проекта.

Ниже перечислены некоторые типовые статьи сметы расходов на проведение проекта по разработке программного продукта с необходимыми пояснениями.

- **Зарплата.** Как показывает практика, это основная статья расходов. Разработанная программа — это почти на 100% овеществленный труд. Менеджеру проекта полезно помнить, что начисления на зарплату в настоящее время в России составляют около 40%.
- **Оборудование.** В большинстве организаций финансисты включают в эту статью почти все, что перечислено в предыдущем разделе. Менеджер проекта обязан проследить, чтобы ничего существенного не было забыто.
- **Расходные материалы.** Кажется, что разработка программ целиком ведется в электронном виде. Тем обиднее менеджеру проекта получить рекламацию от пользователя, которому поставили готовый продукт на носителе низкого качества.
- **Командировки.** Средства по этой статье наиболее интенсивно расходуются в начальной фазе проекта, когда ведется интенсивное взаимодействие с заинтересованными сторонами и на последней фазе проекта, когда ведется сопровождение разработанного продукта. Тем не менее, менеджер должен помнить про эти расходы.
- **Накладные расходы.** Накладные расходы – это плата за то, что проект выполняется в организации, а не «под открытым небом». Как правило, величина этих расходов регламентирована и от менеджера проекта не зависит.
- **Прочие расходы.** Если корпоративные стандарты позволяют включить в смету такую статью, то это стоит сделать. Забытые при планировании мелочи и неприятные неожиданности нельзя исключить. Если же «прочих расходов» не будет, то ничего страшного — экономия средств не вызовет нареканий.

Подробнее особенности управления финансами проекта будут обсуждаться в дальнейшем.

5.8. Организационная структура исполнителей



Организационные структуры проекта (Project organization) – наиболее соответствующая проекту временная организационная структура, включающая всех его участников и создаваемая для успешного достижения целей проекта.

Существуют и используются различные организационные структуры проекта. Ниже рассмотрены наиболее популярные. Прежде чем описывать конкретные структуры, следует определить основные термины, используемые в них. Таких терминов три:

- функция;
- роль;
- должность.

Функция — это вид деятельности, выполняемой в ходе проекта. Выполнение каждой функции требует наличия определенной специфической квалификации и способностей.

Приведем примеры функций:

- **Администрирование.** Ведение договоров; разговоры с заказчиком; составление внешних формальных документов, доклады начальству.
- **Проектирование.** Составление бумажных и/или электронных концепций, моделей, спецификаций и планов.
- **Кодирование.** Ручная и/или полуавтоматическая генерация кода на языке программирования. Автономная (поблочная) отладка кода. Рисование и тестирование интерфейсных элементов (форм).
- **Тестирование.** Пробное использование приложения с целью сломать его, а не решить задачу.
- **Сопровождение.** Развертывание, обучение пользователей, администрирование разработанного или установленного программного обеспечения в процессе опытной эксплуатации.

Роль – это временное назначение сотруднику набора функций в рамках конкретного проекта.

Получение роли означает делегирование полномочий для выполнения определенных функций и принятие ответственности за результаты выполнения этих функций. Роль может требовать выполнения разных функций; некоторые функции могут быть присущи нескольким ролям. Определяющим признаком роли является не

характер деятельности, а набор конкретных результатов (вех), ответственность за достижение которых налагает роль.

В проектах разных типов используются разные наборы ролей из следующего множества:

- Аналитик. Функции: планирование, администрирование.
- Администратор. Функции: администрирование, планирование.
- Программист. Функции: проектирование, кодирование, тестирование, сопровождение.
- Тестер. Функции: тестирование.
- Эксплуатационник. Функции: сопровождение, тестирование.

Должность – это сертифицированная способность играть определенные роли и выполнять определенные функции.

В отличие от роли и функции должность не привязана в конкретному проекту и носит (почти) постоянный характер по времени. В большинстве организаций предусматривается по меньшей мере три уровня иерархии должностей:

- Начальник (отдела, подразделения).
- Руководитель (группы, проекта, направления).
- Исполнитель (инженер, технический специалист).

5.8.1. Иерархическая модель

Иерархическая модель (или модель дерева субординации (см. рис. 1)) являются самой распространенной и известной организационной моделью.

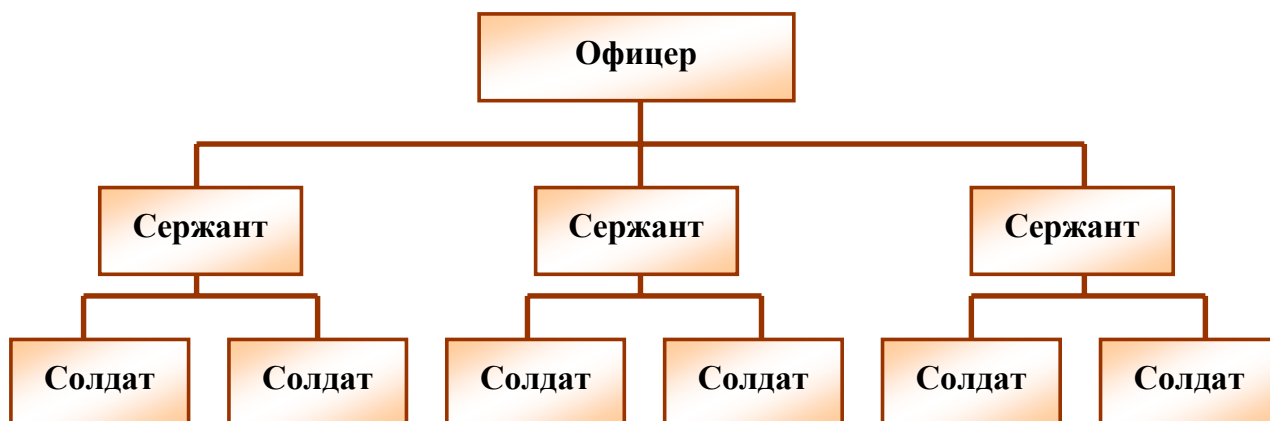


Рис. 1. Иерархическая модель команды (дерево)

Эта модель обладает целым рядом достоинств:

- + **Единоначалие.** Принцип единоначалия обеспечивает очень высокую степень надежности, устойчивости и управляемости команды. В критических ситуациях всегда используется именно эта модель.
- + **Известность.** Иерархическая модель привычна и известна абсолютно всем. Ее использование не требует дополнительных мероприятий по внедрению и обучению персонала.
- + **Масштабируемость.** Иерархическая модель обладает высокой степенью масштабируемости. Она с успехом применяется в масштабах от десятков до десятков миллионов исполнителей.

В то же время иерархической модели присущи некоторые принципиальные недостатки:

- Иерархическая модель экстенсивна. Наращивание функциональности обеспечивается увеличением состава.
- Иерархическая модель жесткая, т.е. практически не допускает перестройки «на ходу» (в процессе). Она ориентирована на выполнение строго определенных функций. Изменение функций требует болезненной перестройки дерева субординации.
- Иерархическая модель консервативна. При ее использовании имеется тенденция к жесткому закреплению за каждым исполнителем его ролевой функции. Она плохо приспособлена для быстрой смены технологий и парадигм.
- Иерархическая модель не устойчива по отношению к личным качествам руководителей. Отрицательные личные качества руководителей оказывают отрицательное воздействие на эффективность команды, причем это воздействие непропорционально увеличивается с ростом уровня в дереве субординации.

5.8.2. Модель «Бригада главного программиста»

Модель бригады главного программиста появилась во время первой технологической революции в программировании на рубеже 60–70-х годов. Долгое время модель главного программиста (модель хирургической бригады, или модель звезды (см. рис. 2)) являлась доминирующей моделью при разработке программного обеспечения.

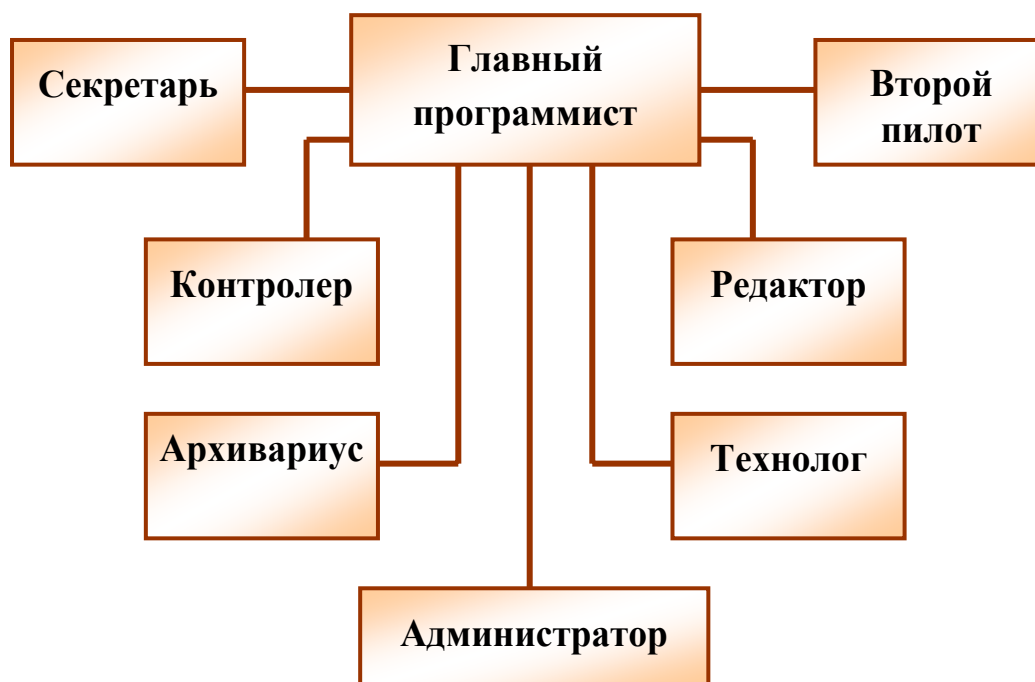


Рис. 2 Модель бригады главного программиста (звезда)

В этой модели главный программист выполняет весь проект сам, а прочие члены бригады ассистируют в predetermined рамках своих ролей и функций.

Модель главного программиста имеет следующие достоинства:

- + **Предсказуемость.** Бригада главного программиста обладает высокой предсказуемостью. Если главный программист плох, то это выявляется на ранних стадиях проекта. Проект может быть прекращен или реорганизован практически без убытков. Если главный программист хорош, то вероятность успешного завершения проекта высока даже при наличии серьезных внешних факторов риска.
- + **Автономность.** Бригада главного программиста обладает высокой автономностью. Она успешно функционирует даже в изменяющейся и неблагоприятной внешней среде.
- + **Гибкость.** Бригада главного программиста обладает достаточной функциональной гибкостью. За счет изменения набора лучей в звезде ее легко можно ориентировать на различные типы программных проектов.
- + **Единоначалие.** Бригада главного программиста наследует все достоинства принципа единоначалия (поскольку главный программист единолично принимает все принципиальные решения по проекту).

Метод бригады главного программиста допускает различные модификации при сохранении своей сути.

- Первая модификация: ***изменение количества и качества лучей в звезде***. В классической модели Брукса в звезде еще присутствовали второй Секретарь и Языковед. В практических случаях количество лучей сокращают, например, объединяя Секретаря, Редактора и Архивариуса. Характеристическими ролями в бригаде являются Главный программист, Второй пилот и Администратор, остальные роли меняются по мере развития технологий программирования.
- Вторая модификация: ***на роль главного программиста назначается не кодировщик, а, например, аналитик или менеджер продукта***. В этом случае кодирование ведет Второй пилот, но все принципиальные решения принимает Главный программист. В современных условиях наблюдается тенденция перемещения принятия ключевых решений со стадии кодирования на более ранние стадии, поэтому вторая модификация является фактически стандартной.
- Третья модификация: ***«сдвоенный центр»*** (рис. 3). Эта модификация позволяет хотя бы в некоторых пределах масштабировать бригаду. Имеются два Главных программиста, которые делят проект пополам, все решения принимают консенсусом и являются вторыми пилотами друг для друга.

Замечание. При использовании большего количества главных программистов модель перестает работать, потому что коммуникации, достижение консенсуса и взаимное дублирование работы требует слишком больших накладных расходов.



Рис. 3 Модификация модели бригады главного программиста (сдвоенный центр)

Модель бригады главного программиста имеет определенные недостатки.

- Бригада главного программиста не является масштабируемым решением. Она отлично работает на проектах объема 6-8 человек × 1-2 года. Если проект требует более коротких сроков или существенно больших объемов, то использование бригады главного программиста затруднено. Замечание. Если формально разрезать крупный проект на несколько частей и запустить несколько бригад параллельно, то результаты их работы будет очень трудно синхронизировать и интегрировать в одно приложение. Дело в том, что главный программист держит очень много в голове, часто опуская этапы формального документирования и спецификации. За счет этого повышается производительность, но затрудняется совместимость. Очень короткий проект бригаде главного программиста трудно провести потому, что главный программист последовательно выполняет всю основную работу, и его личные возможности ограничивают производительность бригады.
- Бригада главного программиста не является распараллеливаемой структурой. Она действует по принципу: один проект — одна команда. Практически невозможно выполнять бригадой одновременно разные фазы разных проектов.

- Бригада главного программиста имеет уязвимое центральное звено. Очень велик управленческий риск мгновенной аннигиляции бригады, если что-то случается с главным программистом. Замечание. Второй пилот в бригаде главного программиста должен тщательно отслеживать все действия главного программиста. Это несколько снижает риск аннигиляции.

5.8.3. Модель «Команда равных»

Модель команды равных является составной частью Microsoft Solutions Framework (MSF) – методологии разработки программных проектов фирмы Microsoft. Это наиболее демократичная модель, поскольку в ней нет явно выделенного центра. Схематически ее принято изображать в виде цикла (рис. 4), где все роли равноправны и связаны друг с другом.

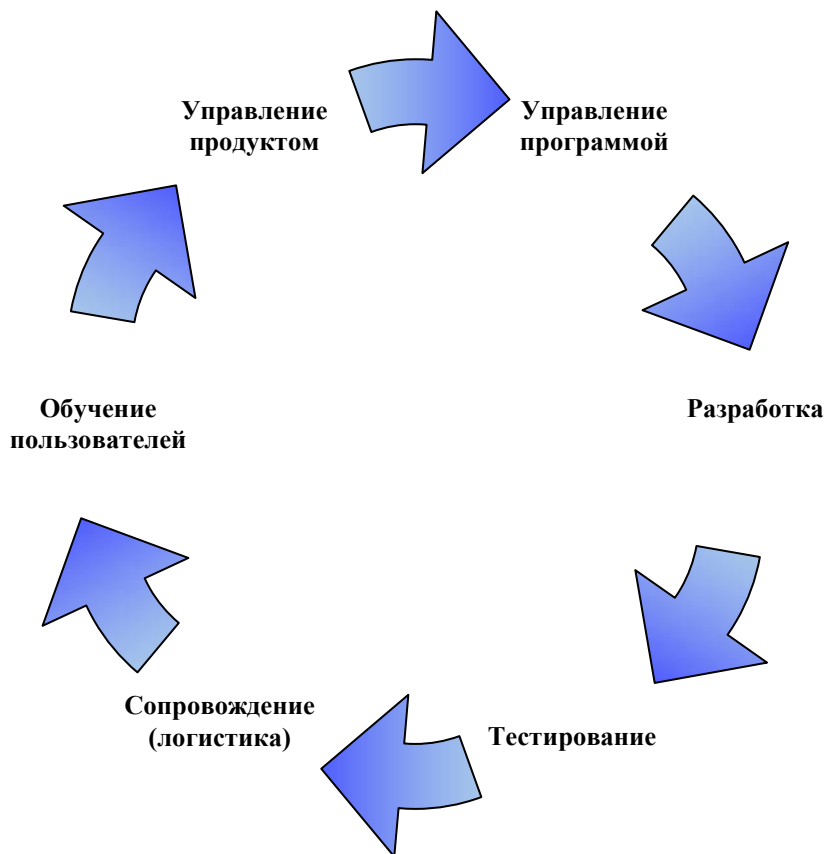


Рис. 4 Модель команды MSF (цикл)

Замечание. Чтобы подчеркнуть отсутствие иерархии в команде MSF, роли в команде обозначаются не названиями должностей, а названиями функций.

Преимущества модели команды MSF.

- + Высокая производительность, поскольку непроизводительные трудозатраты на поддержание формальных и субординационных связей сведены к минимуму.
- + Сравнительно легкая масштабируемость. Каждый элемент в схеме команды может быть в свою очередь циклом.
- + Сильная положительная мотивация труда и равно высокая заинтересованность всех участников в конечном успехе.

Основные недостатки модели MSF являются продолжением ее достоинств.

- Для формирования команды MSF нужны равные (равно квалифицированные и равно заинтересованные) участники.
- Критическое значение имеет коммуникабельность (большая часть коммуникаций неформальны), умение и готовность работать в коллективе (артельный дух).
- Демократичная модель команды MSF плохо сопрягается с жесткой иерархической моделью подразделения (предприятия).

5.8.4. Матрица ответственности

Матрица распределения ответственности (Responsibility Assignment Matrix) – структура, которая ставит в соответствие организационной структуре проекта структуру декомпозиции работ (WBS) для назначения ответственных лиц за каждую работу.

Приведенное определение полностью объясняет основное назначение матрицы ответственности. Для эффективной организации работ по проекту необходимо точно определить, кто за что отвечает, и кто что делает. Матрица распределения ответственности – это двумерная таблица, строкам которой назначены все работы проекта (взятые, например, из структуры декомпозиции работ), а столбцам назначены все роли используемые в проекте. Если в проекте есть несколько сотрудников, играющих одну и ту же роль, то заводится несколько столбцов. Другими словами, в матрице распределения ответственности столько строк, сколько всего работ в проекте и столько столбцов, сколько всего исполнителей в проекте. Если какой-то сотрудник, играющий определенную роль, назначен ответственным за выполнение определенной работы, то на пересечении соответствующих строки и столбца делается отметка.

Матрица распределения ответственности является удобным инструментом управления. Она позволяет быстро и в наглядной форме ответить на важные для менеджера вопросы.

- ◆ Кто отвечает за данную работу?
- ◆ Есть ли работы, за которые никто не отвечает (а значит, никто и не сделает)?
- ◆ Есть ли работы, за которые отвечают несколько исполнителей? (Это ошибка планирования)
- ◆ За что отвечает данный сотрудник?
- ◆ Есть ли сотрудники, которые ни за что не отвечают? (Их можно вывести из проекта)
- ◆ Есть ли сотрудники, которые отвечают за слишком большое количество работ? (Есть риск, что они не справятся со всеми делами)

Матрица распределения ответственности является неотъемлемым элементом плана проекта.

5.9. Прикладные программные средства для менеджера проекта

Для управления проектами разработано большое количество специализированного программного обеспечения (например, Time Line, Microsoft Project, Guide Line, Project Expert, Primavera Project Planner, Open Plan, Spider Project). Использование такого рода инструментария особенно важно в организациях, где необходима строгая стандартизация и координация ведущейся проектной деятельности, представление целостной картины состояния портфеля проектов, централизованное управление проектами и ресурсами, а также отчетность по проектам и ресурсам более высокого уровня.

Как правило, такие инструменты позволяют:

- ◎ осуществлять планирование проекта, в т.ч. в удобном виде (диаграмм Ганта, сетевых диаграмм, календарей проекта) формировать календарные планы проекта;
- ◎ осуществлять управление ресурсами проекта;
- ◎ оценивать эффективность планов и оптимизировать планы, в т.ч. работая с различными версиями плана;
- ◎ отслеживать выполнение проекта.

Наиболее развитые из систем подобного рода (MS Project, Primavera Project Planner) поддерживают полномасштабное управление корпоративными проектами, т.е. позволяют получать сведения о портфеле проектов в масштабе целой организации для более

эффективного анализа и оптимизации процесса принятия решений. Управление корпоративными проектами, в частности, обеспечивает следующие возможности:

- ⊙ Просмотр согласованной бизнес-статистики по всем проектам портфеля, с возможностью получения более подробных сведений.
- ⊙ Оценка и моделирование календарных планов, данных о ресурсах и затратах по определенному периоду времени и в масштабе нескольких проектов для выявления тенденций и выявления проблемных областей.
- ⊙ Эффективный выбор сотрудников для участия в проектах, а также отслеживание и управление этим процессом в масштабе организации с помощью средств назначения с учетом квалификации.
- ⊙ Отслеживание наличия сотрудников и производственных возможностей, необходимых для реализации будущих проектов.
- ⊙ Оптимизация процессов управления проектами за счет определения стандартов и рекомендаций для всей организации.
- ⊙ Определение общих для всех проектов процессов и правил создания отчетов и утверждения времени работы над проектом в целях обеспечения точности данных.

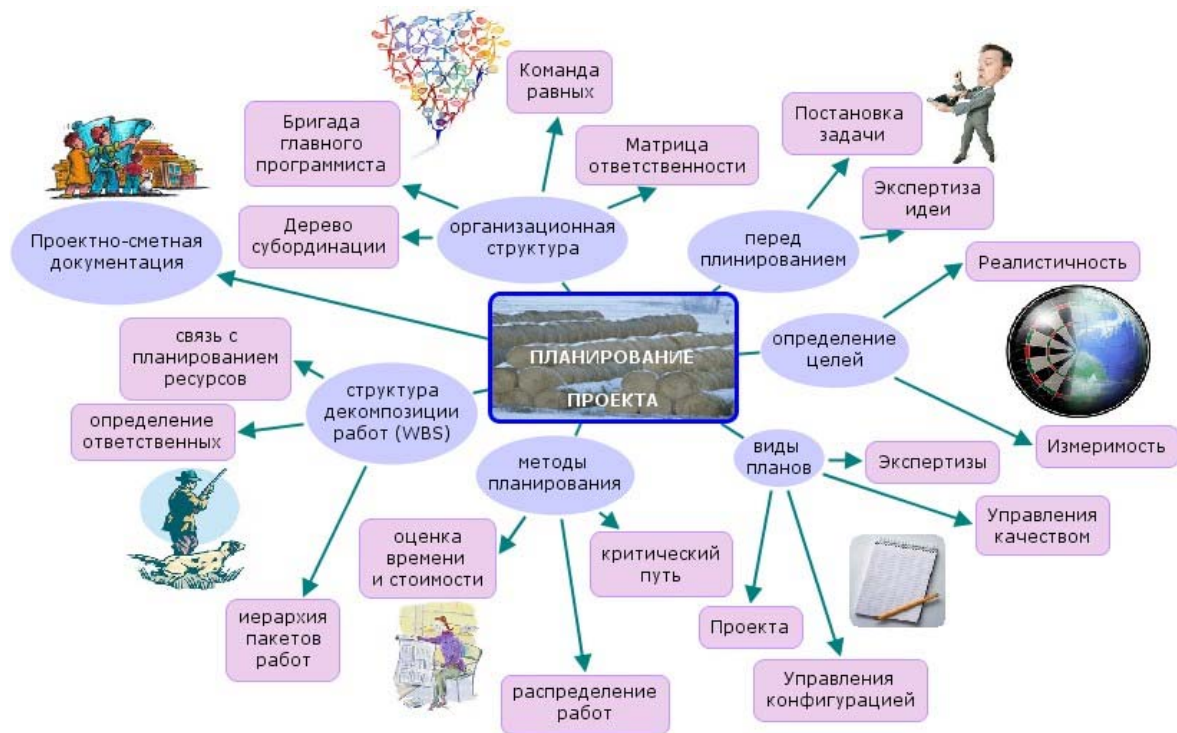
Управление корпоративными проектами позволяет более эффективно управлять производственным процессом, налаживать воспроизводимые процессы и оптимизировать ресурсы в масштабе предприятия для сокращения затрат, повышения качества и сокращения времени производства. Кроме того, соответствующее программное обеспечение оптимизирует координирование и совместное использование данных, предоставляя членам групп развитые возможности просмотра данных, относящихся к «своим» работам, и посылки отчетов о заданиях, создания отчетов о ходе выполнения задач и совместной работы над проектами. При этом также обеспечивается:

- ✓ Снижение объемов административной работы и получение точных обновленных статистических данных о календарных планах проекта и бюджете.
- ✓ Координация работы групп за счет использования порталов и автоматических уведомлений, повышающих эффективность и производительность работы сотрудников.
- ✓ Централизованное хранение, связывание и совместное использование данных, таких как документы, вопросы и риски, относящейся к планам проектов.
- ✓ Эффективная совместная работа с использованием функций управления документами, включая управление версиями.

5.10. Заключение

Планирование проектов разработки программного продукта имеет все основные характеристики, присущие планированию вообще, а также некоторые специальные аспекты, связанные с особенностями процесса разработки программного обеспечения.

5.11. «Карта памяти» по теме



5.12. Список использованной и рекомендованной литературы

1. Баранов С. Н. Управление программным проектом. Лекции по спецкурсу "Технология программирования". - СПб: Санкт-Петербургский государственный электротехнический университет, рукопись, 1998.
2. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. - СПб.: Символ-Плюс, 1999.
3. Эдвард Йордон. Путь камикадзе. Как разработчику программного обеспечения выжить в безнадежном проекте. - М.: ЛОРИ, 2001.
4. Пашкус Ю. В., Мисько О. Н. Введение в бизнес. - Л.: "Северо-Запад", 1991.
5. Conger, Sue A. The New Software Engineering. Wadsworth Publishing Company, 1994.

6. Pierre N. Robillard, Martin P. Robillard. Types of collaborative work in software engineering. // The Journal of Systems and Software 53, 2000, pp. 219-224.
7. Rob Thomsett. Effective Project Teams: A Dilemma, a Model, a Solution. American Programmer, July-August 1990.
8. Vliet H. V. Software Engineering: Principles and Practice. John Wiley and Sons, 2000.

Тема 6. Бизнес-анализ и управление проектом

6.1. Введение

Значительную часть современных проектов по разработке программного обеспечения составляют проекты разработки или внедрения различного рода корпоративных информационных или управляющих систем. Важным этапом проектирования таких систем является исследование и описание бизнес-процессов организации-заказчика, называемая обычно **бизнес-анализом**. Хотя бизнес-анализ и не является функцией менеджера проекта (этим занимаются специалисты – бизнес-аналитики), менеджеру таких проектов необходимо тесно сотрудничать с бизнес-аналитиком и, в частности, уметь разбираться в предлагаемых последним моделях бизнес-процессов предприятия заказчика.

Изучив учебный материал данной темы, Вы:

- ⊙ узнаете о том, как соотносится между собой деятельность бизнес-аналитика и менеджера проекта в проектах по разработке и внедрению ERP систем;
- ⊙ получите общее представление об основных методологиях структурного анализа и проектирования, основанных на стандартах IDEF0, IDEF3, а также на методологии моделирования систем, основанных на данных, с использованием диаграмм «сущность-связь».

В рамках темы рассматриваются следующие учебные вопросы:

- ⊙ управление проектом и бизнес-анализ.

6.2. Проекты разработки и внедрения корпоративных информационных и управляющих систем

Значительную часть современных проектов по разработке программного обеспечения составляют проекты разработки или внедрения различного рода корпоративных информационных или управляющих систем – например, ERP (Enterprise Resource Planning) или CRM (Customer Relationship Management) систем. К таким системам относятся системы управления производством на заводе, системы автоматизации торговли и учета для сети гипермаркетов, банковские системы автоматизации документооборота и т.п. Несмотря на то, что значительная часть такого рода систем автоматизации строится не «с нуля», а на основе уже разработанных специальных сложных программных комплексов (например, SAP/R3 или Oracle Business Suite

для сложных систем управления производством, Axapta Retail для систем управления в магазинах и сетях оптовой или розничной торговли), любой такого рода проект строго индивидуален, т.е. делается под конкретное предприятие. Как это ни покажется странным, даже для предприятий торговли в силу сложности и индивидуальности бизнес-процессов не существует ни одного приемлемого для всех «стандартного» решения автоматизации. В такого рода работах можно выделить следующие этапы, предшествующих собственно разработке или интеграции системы.

1. **Функциональное и информационное обследование предприятия-заказчика.** Цель этого этапа состоит в сборе подробной информации о направлениях работы, решаемых задачах, приоритетах, показателях развития, показателях эффективности, организационной структуре, выполняемых функциях, сопутствующих им материальных, финансовых и информационных потоках, взаимодействии с внешними абонентами, используемых средствах автоматизации и др.
2. **Определение миссии предприятия, иерархии целей, разработка функциональных моделей существующей («как есть», as is) организации бизнес-процессов.** Цель этого этапа состоит в формальном представлении собранной информации в соответствии с согласованной целью проекта.
3. **Анализ организации бизнеса.** Оценка эффективности существующей организации бизнес-процессов. На этом этапе выделяются показатели эффективности выполнения бизнес-процессов, выполняется анализ существующего уровня и методов их информационной поддержки, функционально-стоимостной анализ, выделение центров затрат, динамический анализ загрузки и распределения ресурсов и др.
4. **Разработка функциональных и информационных моделей рациональной («как должно быть», to be) организации бизнес-процессов,** в соответствии с установленной иерархией целей предприятия, на основе согласованных принципов и возможностей информационных технологий. На этом этапе проводится также формирование предложений по совершенствованию системы управления предприятия и системы информационной поддержки бизнес-процессов.
5. **Оценка эффективности предлагаемых решений.** На этом этапе проводится расчет значений, выделенных ранее показателей эффективности бизнес-процессов, соответствующих моделям «как должно быть», их сопоставление с значениями, полученными для моделей «как есть», анализ различий.

6. Разработка поэтапного плана реорганизации и проекта корпоративной информационной системы. На этом выполняется также анализ окупаемости инвестиций.

Более детальное содержание и трудоемкость выполнения каждого из этапов определяется особенностями конкретного проекта и конкретного предприятия. Наибольшее значение имеют следующие факторы:

- выбор приоритетного направления исследований: управление предприятием, финансами, материальными потоками, информацией, документами, обеспечение конкурентоспособности, снижение издержек и др.;
- существующее количество уровней управления, структурных подразделений, отделений, филиалов;
- уровень типизации, т.е. возможность использования типовых решений в различных автономно работающих подразделениях;
- степень детализации проработки проекта: уровень управления предприятием, основными и/или вспомогательными подразделениями, спецификация типовых рабочих мест, уровень конкретного документа и др.;
- уровень развития информационной системы предприятия.

В связи со сложностью таких проектов для их выполнения привлекаются специалисты по анализу и моделированию бизнес-процессов предприятия (бизнес-аналитики). Это специально обученные специалисты, которые в ряде случаев должны обладать кроме общих знаний о структурном анализе и моделировании бизнес-процессов различных предприятий, еще и специальными знаниями в конкретных предметных областях.

В помощь бизнес-аналитику на сегодняшний день разработан целый ряд методологий в структурном анализе и моделировании бизнес-процессов, а также проектирования основанных на них корпоративных информационных и управляющих систем. Все эти методологии, как правило, формализуют процесс бизнес-анализа и моделирования и представляют соответствующие модели в виде согласованных наборов специальных диаграмм. К таким методологиям относятся, в частности, SADT (Structured Analysis and Design Technique) и входящие в него стандарты IDEF0, IDEF3 и IDEF1X, методология объектного и процессного моделирования RUP (Rational Unified Process), основанная на использовании диаграмм UML, моделирование систем, управляемых данными при помощи диаграмм DFD (Data Flow Diagram) потоков данных и диаграмм «сущность-связь» ERD (Entity-Relationship Diagram),

и т.п. Они многократно апробированы в разнообразных проектах по всему миру и утверждены в виде международных и национальных стандартов ряда стран. Для их использования разработано и специальное программное обеспечение. Например, для построения полноценных функциональных моделей бизнес-процесса на основе диаграмм IDEF0, IDEF3, DFD весьма популярен AllFusion Process Modeler (BPWin). Для этих же целей предназначен Design/IDEF. Для моделирования систем, управляемых данными, в частности, на основе диаграмм «сущность-связь» ERD, популярен AllFusion Data Modeler (ERWin). Для объектного моделирования можно использовать AllFusion Component Modeler или Rational Rose. Использование этих методологий и программных продуктов позволяет разработать всесторонне проработанные проекты реорганизации предприятия на основе количественных показателей, сформированных в результате анализа всех наиболее важных аспектов функционирования предприятия.

Все описанное, как правило, не является прерогативой менеджера проекта — этим занимается бизнес-аналитик. Но в такого рода проектах весь процесс планирования существенным образом зависит от построенной бизнес-модели. Таким образом, менеджеру поневоле приходится вникать в деятельность бизнес-аналитика, и в частности, читать, построенные последним бизнес-модели.

6.3. SADT — методология структурного анализа и проектирования

SADT (Structured Analysis and Design Technique) является одной из самых известных и широко используемых методологий структурного анализа и проектирования сложных систем. Разработана она была Дугласом Т. Россом (D. T. Ross) в конце 1960х-начале 1970х годов изначально для аэрокосмической и военной промышленности и впоследствии получила широчайшее распространение во многих областях.



В основе этой технологии лежит описание бизнес-моделей при помощи специальных диаграмм. В связи с широким распространением SADT отдельные ее части перешли в стандарты. Такими стандартами стали IDEF0 (функциональное моделирование), IDEF3 (моделирование процессов) и IDEF1X (описание информационной модели, т.е. отображение структуры и содержания информационных потоков, необходимых для поддержки функций системы).

SADT не является единственной возможной методологией бизнес-анализа и проектирования. Например, в последние годы на роль успешно конкурирующей альтернативы выходит RUP (Rational Unified

Process) — технология, разработанная Гради Бучем (G. Booch). RUP базируется на UML и фактически представляет собой методологию применения UML для проектирования и моделирования приложений. При этом RUP претендует на сквозной цикл проектирования, однако предлагает довольно слабые средства для построения бизнес-модели задачи на ранних стадиях проектирования. Взамен построения полноценной функциональной модели предлагает ограничиться объектной моделью и моделью процессов. Однако для крупных проектов со сложными задачами одного RUP может оказаться недостаточно, так что лучше все-таки начать с SADT. Чем ближе проект к завершению, тем большую роль в нем будет играть RUP. Впрочем, тяжеловесность и формализм RUP делают невыгодным его применение для относительно небольших проектов, поскольку на этот случай у RUP имеется младший брат - ICONIX. ICONIX гораздо компактнее и требует куда меньших накладных расходов, поэтому его целесообразно применять, когда коллектив разработчиков невелик, а задача обзрима.

6.3.1. Основные принципы SADT

В SADT вводятся следующие основные понятия.

Система – совокупность взаимодействующих компонентов и взаимосвязей между ними.

Моделирование – процесс создания точного описания системы.

Модель (SADT-модель) – полное и точное описание системы с помощью SADT, которое может быть использовано для получения ответов на вопросы относительно системы с заданной точностью.

Иначе говоря, целью моделирования является получение ответов на некоторую совокупность вопросов. Эти вопросы всегда неявно присутствуют в процессе анализа системы и руководят созданием модели. Если модель отвечает не на все вопросы или ее ответы недостаточно точны, то говорят, что модель не достигла своей цели.

Поскольку моделируемая система никогда не существует изолированно, в методологии SADT подчеркивается необходимость точного определения границ системы. SADT-модель всегда ограничивает свой предмет, то есть точно определяет, что является и что не является предметом моделирования, описывая то, что входит в систему, и подразумевая то, что лежит за ее пределами. Ограничивая предмет моделирования, SADT-модель помогает сконцентрировать

внимание именно на описываемой системе и позволяет избежать включения посторонних предметов.

В SADT-моделях используются как естественный (русский, английский) язык, так и специальная графическая нотация для диаграмм. Модель может быть сосредоточена либо на функциях системы, либо на ее объектах. SADT-модели, ориентированные на функции, принято называть **функциональными моделями**, а ориентированные на объекты системы – **моделями данных**. Функциональная модель представляет с требуемой степенью детализации функций систему, которые отражают свои взаимоотношения через объекты системы.

С определением модели тесно связана позиция, с которой наблюдается система и создается ее модель. Поскольку качество описания системы резко снижается, если оно не сфокусировано ни на чем, SADT требует, чтобы модель рассматривалась все время с одной и той же позиции. Эта позиция называется **«точкой зрения»** данной модели. Точку зрения лучше всего представлять себе как место человека или объекта, в которое надо встать, чтобы увидеть систему в действии. С этой фиксированной точки зрения можно создать согласованное описание системы так, чтобы в ней не смешивались не связанные описания. Очевидно, что функциональные модели одного и того же предприятия с точек зрения начальника производства и главного бухгалтера будут существенно различаться по направленности их детализации. Это связано с тем, что по роду служебных обязанностей главного бухгалтера в меньшей степени интересуют вопросы обслуживания станочного парка предприятия, а начальника производства мало волнуют тонкости исчисления налогооблагаемой базы. Точка зрения определяет основное направление развития модели и уровень необходимой детализации. Четкое фиксирование точки зрения позволяет упростить модель, отказавшись от детализации и исследования отдельных компонентов, не являющихся для данной модели важными. Правильный выбор точки зрения существенно сокращает временные затраты на построение конечной модели.

На рис. 1 показано, как автор модели подразделения разработки программного обеспечения, которое называется «Лаборатория АВР», перечисляет претендентов, с точки зрения которых можно было бы описывать подразделение. Если в модели работы программирующего подразделения не зафиксировать определенную точку зрения, то легко можно смешать проблему инженерного обслуживания компьютеров с тем, какие будут применяться отступы при кодировании на C++. Если это произойдет, то читатель модели столкнется с трудностями при определении конкретных обязанностей персонала.

Вопросы: Каковы обязанности архитектора? Каковы обязанности программиста? Каковы обязанности тестера? Кто контролирует выдачу и прохождение заданий? Как обеспечивается управление конфигурацией? На каких стадиях требуются диаграммы? Как производится обработка дефектов?		Цель: Определить обязанности каждого сотрудника лаборатории АВР и понять, как эти обязанности связаны между собой, с тем чтобы написать должностные инструкции	
Претенденты: Главный архитектор Технический лидер Уполномоченный по качеству Заведующий лабораторией		Точка зрения: Заведующий лабораторией	
NODE:	ABPA-0	TITLE:	Цель и точка зрения модели АВР
		NO.:	ФАН 001

Рис. 1 Цель и точка зрения модели

Очень часто только с позиции одной точки зрения можно получить описание, удовлетворяющее цели моделирования. Например, создание согласованной модели Лаборатории АВР можно осуществлять с точки зрения, как архитектора, так и технического лидера или уполномоченного по качеству, но ни одна из них не даст модели, которая позволила бы написать должностные инструкции для всего персонала. Только с позиции заведующего лабораторией можно увидеть все виды работ, выполняемых в ней. Именно с его точки зрения можно проследить взаимосвязи обязанностей различных работников. Точка зрения заведующего лабораторией позволяет создателю модели определить роль каждого работника в изготовлении программного продукта и описать обязанности персонала.

После того как определены предмет, цель и точка зрения модели, начинается первая итерация процесса моделирования по методологии SADT. Автор определяет, что включить в модель, а что исключить из нее. Точка зрения диктует автору модели выбор нужной информации о предмете и форму ее подачи. Цель становится критерием окончания моделирования. Конечным результатом этого процесса является набор взаимоувязанных описаний, начиная с описания самого верхнего уровня всей системы и кончая подробным описанием деталей или операций. Каждое из таких описаний называется **диаграммой**. Диаграмма является основным рабочим элементом при создании модели. SADT-модель объединяет диаграммы в иерархическую древовидную структуру, в

которой верхняя диаграмма является наиболее общей, а самые нижние наиболее детализированы.

На рис. 2 верхняя диаграмма описывает работу лаборатории как функцию, в основе которой лежит преобразование входящих рабочих материалов (требований) в готовый результат (программный продукт). Диаграмма на рис.3 детализирует диаграмму на рис. 2, указывая на три главные функции: подготовка к проведению разработки, выполнение разработки, проверка результатов и передача их заказчику. Таким образом, общая функция, указанная на верхней диаграмме, детализируется с помощью трех функций на нижней диаграмме.

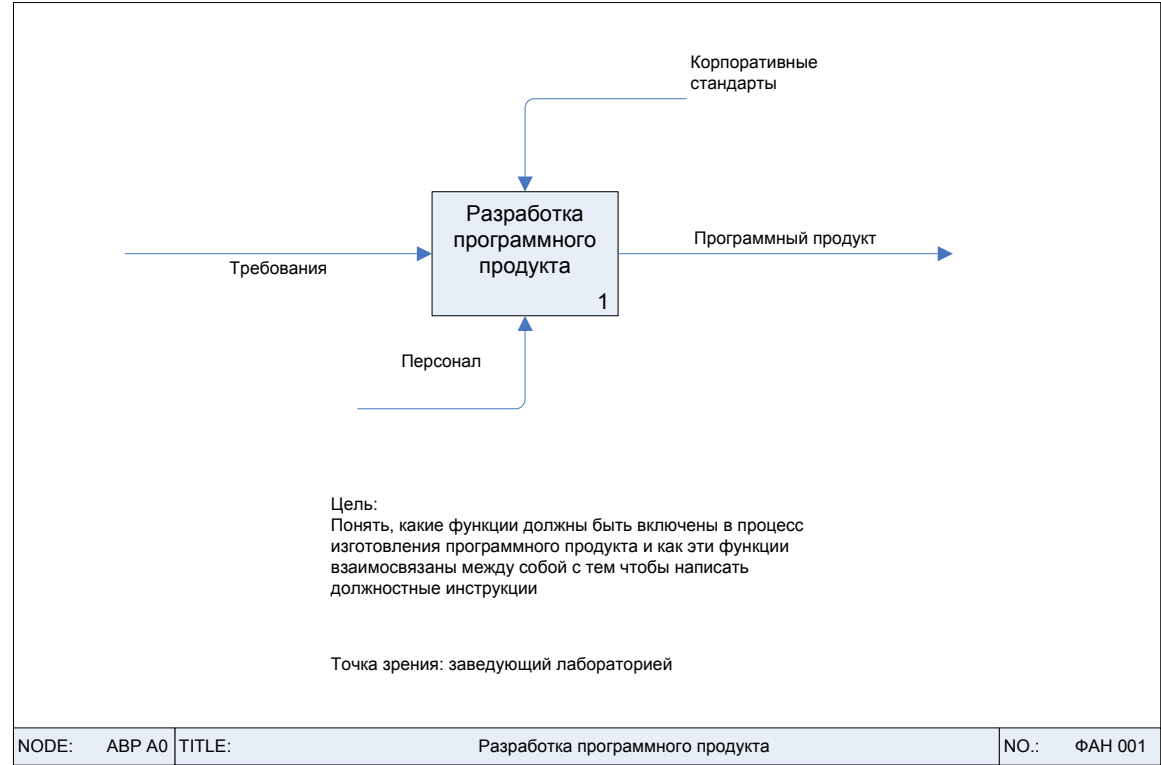


Рис. 2 Представление функции верхнего уровня

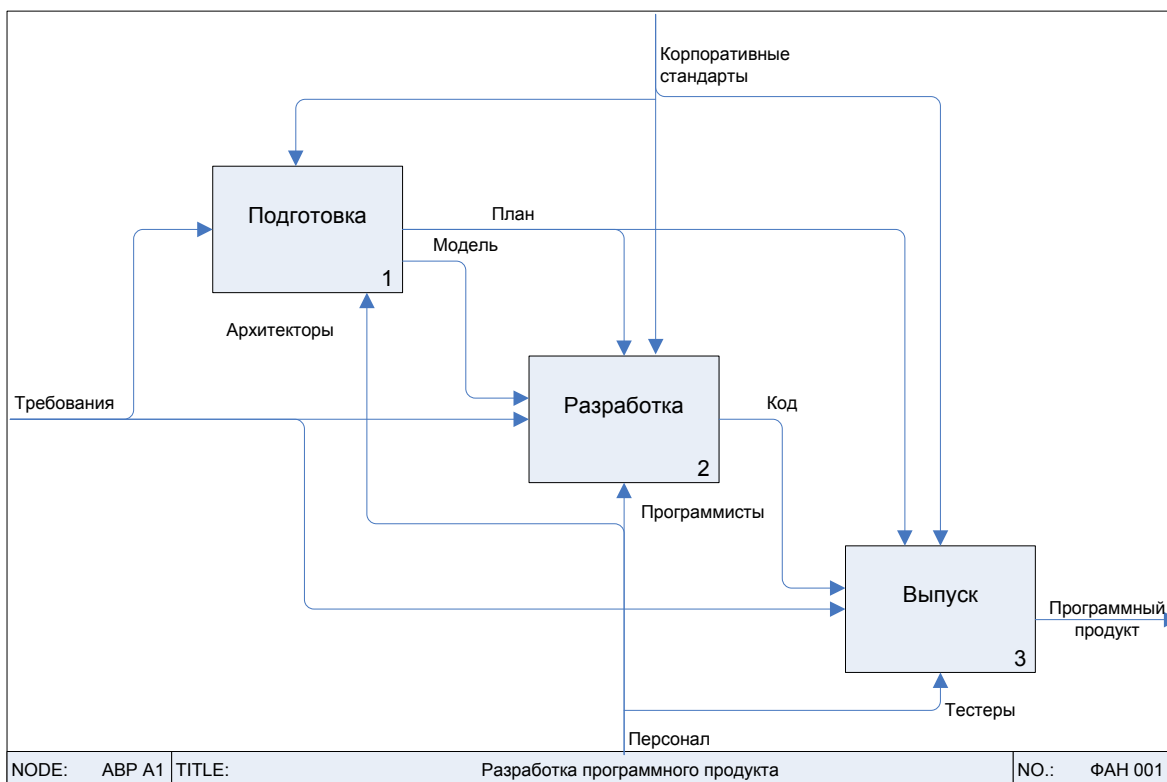


Рис. 3 Детализация функции

На рис. 3 показано также взаимное влияние трех функций нижней диаграммы, обозначенное дугами, которые символизируют объекты. Видно, что некоторые дуги доходят до границы диаграммы и что имена этих дуг совпадают с теми, что указаны на дугах верхней диаграммы. Это пример того, как SADT соединяет диаграммы в модели через объекты системы.

Процесс создания SADT-модели итеративный: модели создаются и проходят серию последовательных улучшений до тех пор, пока они в точности не будут представлять объект моделирования. В частности, одна и та же диаграмма может рассматриваться неоднократно, что приводит к появлению различных ее вариантов. Чтобы различать разные версии одной и той же диаграммы, в SADT используется схема контроля конфигурации диаграмм, основанная на хронологических или **С-номерах** (от Chronological Number). С-номерные коды обычно образуются из инициалов автора и последовательных номеров. Эти коды ставятся в нижнем правом углу SADT-бланка. Например, ФАН 001 – это С-номер для диаграммы «Разработка программного продукта» на рис. 3. Если диаграмма заменяет более ранний вариант, то автор помещает предыдущий С-номер в скобках, чтобы указать на связь с предыдущей работой. Каждый автор проекта SADT ведет реестр созданных им диаграмм, нумеруя их последовательными целыми числами.

Помимо итерационной разработки моделей, одной из основных компонентов методологии SADT является **итерационное рецензирование**, в процессе которого автор и эксперт неоднократно

проверяют достоверность создаваемой модели. Итерационное рецензирование называется **циклом автор-читатель**. Цикл автор-читатель начинается в тот момент, когда автор принимает решение распространить информацию о какой-либо части своей работы с целью получения отзыва о ней. Организация своевременной обратной связи имеет важнейшее значение для эффективного моделирования, потому что даже малая доля неверной или неверно интерпретированной аналитиком информации, заложенной в SADT-диаграмму, может привести к недостоверному описанию моделируемой системы.

6.3.2. Стандарт IDEF0. Синтаксис диаграмм, применяемых для функциональных моделей

Составной частью SADT является стандарт IDEF0, определяющий синтаксис диаграмм, предназначенных для функционального моделирования. В основном стандарт IDEF0 строго следует принципам структурного анализа, изложенным в предыдущем разделе, а также добавляет и уточняет детали представления SADT-моделей.

IDEF0 используется для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, преобразуемые этими функциями. Процесс использования IDEF0 приблизительно такой:

- ⊙ составляется модели «как есть» с точек зрения участников бизнес-процесса (например, менеджер, бухгалтер, руководитель..),
- ⊙ со точки зрения бизнес-аналитика составляется обобщенная модель,
- ⊙ на основе задачи автоматизации, поставленной заказчиком, составляется модель «как должно быть».

Бланк диаграммы

Диаграммы IDEF0 изображаются на специальных бланках. Вся диаграмма располагается в средней части бланка, а в верхней части («Заголовок») и в нижней части («Подвал») располагается служебная информация. Диаграмме дается название, которое располагается в центре нижней части ее бланка. В верхней части бланка диаграммы располагается стандартная идентифицирующая диаграмму информация: автор диаграммы, частью какого проекта является работа, дата создания или последнего пересмотра диаграммы, статус диаграммы. Все элементы бланка диаграммы перечислены в табл. 1. Пример использования бланка см. ниже на рис. 4.

Таблица 1.

Элементы «Заголовка» и «Подвала» диаграммы IDEF0.

Поле	Перевод	Назначение
Элементы заголовка диаграммы		
Used At	Используется В	Используется для внешних ссылок на диаграмму
Author, project, date, revised	Автор, проект, дата, пересмотр	Содержит имя автора диаграммы, даты создания и последнего внесения изменений, название проекта
Notes 1...10	Замечания	При ручном редактировании очередная цифра зачеркивается при внесении изменений
Status:	Состояние:	Отражает состояние разработки или утверждения диаграммы
Working	Рабочая версия	Новая диаграмма, глобальные изменения или новый автор
Draft	Эскиз	Диаграмма достигла уровня готовности представления на утверждение
Recommended	Рекомендовано	Диаграмма одобрена и утверждена
Publication	Публикация	Диаграмма готова для печати и публикации
Reader	Читатель	Имя читателя
Date	Дата	Дата знакомства читателя с диаграммой
Context	Контекст	Схематическое изображение функциональных блоков на родительской диаграмме
Элементы подвала диаграммы		
Node	Узел	
Title	Название	
C-Number	Номер	

Функциональная декомпозиция

Каждая IDEF0-диаграмма содержит **блоки** и **дуги**. Блоки изображают функции моделируемой системы. Дуги связывают блоки и отображают взаимодействия и взаимосвязи между ними. Функциональные блоки на диаграммах изображаются прямоугольниками. Блок представляет функцию или активную часть системы, поэтому названиями блоков служат глаголы или глаголы с пояснительными словами. SADT рекомендует, чтобы в диаграмме было не менее трех и не более шести блоков.

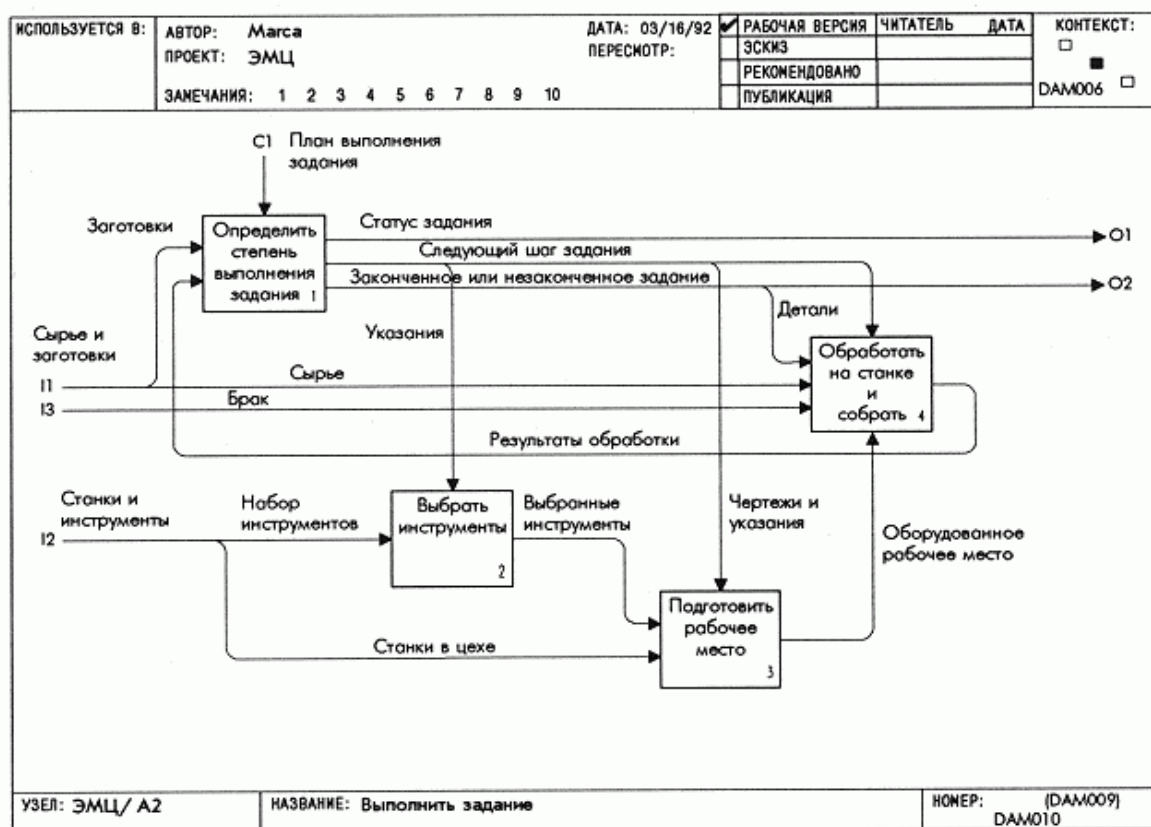


Рис. 4 Типичная IDEF0-диаграмма.

Блоки должны размещаться на диаграмме по степени важности (с точки зрения автора диаграммы). В SADT этот относительный порядок называется **доминированием**. Доминирование понимается как влияние, которое один блок оказывает на другие блоки диаграммы. Например, самым доминирующим блоком диаграммы может быть либо первая из требуемой последовательности функций, либо планирующая или контролирующая функция, влияющая на все другие функции. Наиболее доминирующий блок обычно размещается в верхнем левом углу диаграммы, а наименее доминирующий — в правом нижнем углу.

Блоки должны быть перенумерованы. Номера блоков служат однозначными идентификаторами для системных функций и автоматически организуют эти функции в иерархию модели.

Дуги на диаграмме изображаются одинарными линиями со стрелками на концах. Для IDEF0-диаграмм дуга представляет множество объектов и описывается существительным или существительным с определениями. Здесь используется общее понятие «объекты», поскольку дуги в SADT могут представлять планы, данные в компьютерах, машины, информацию или что-то другое — материальное или нематериальное. Поскольку дуги представляют наборы объектов, они могут иметь множество начальных и конечных точек. Поэтому дуги могут различными способами разветвляться и соединяться.

Между объектами и функциями возможны четыре отношения: **вход (input)**, **управление (control)**, **выход (output)**, **механизм (mechanism)**. Каждое из этих отношений изображается дугой, связанной с определенной стороной блока: левая сторона блока предназначена для входов, верхняя – для управления, правая – для выходов, нижняя – для механизмов. Такое обозначение отражает определенные системные принципы: входы преобразуются в выходы, управление ограничивает или предписывает условия выполнения преобразований, механизмы показывают, какие ресурсы необходимы для выполнения функции.

Входные дуги изображают объекты, используемые и преобразуемые функциями. Управленческие дуги представляют информацию, управляющую действиями функций. Выходные дуги изображают объекты, в которые преобразуются входы. Дуги механизмов отражают то, как функции реализуются. Таким образом, IDEF0-диаграммы не являются ни блок-схемами, ни просто диаграммами потоков данных. Это предписывающие диаграммы, представляющие преобразования входов в выходы и указывающие правила этих преобразований. Дуги на них изображают интерфейсы между функциями системы и между системой и окружающей средой.

В методологии SADT определены пять типов взаимосвязей между блоками для описания их отношений:

- ⊙ управление,
- ⊙ вход,
- ⊙ обратная связь по управлению,
- ⊙ обратная связь по входу,
- ⊙ выход-механизм (встречается редко и отражают ситуацию, при которой выход одной функции становится средством достижения цели для другой; в этом случае дуга механизма обозначает строго последовательную взаимосвязь: приготовления должны быть завершены до начала работы).

Взаимосвязь диаграмм в модели

Для описания системы в целом, как правило, требуется несколько диаграмм. Диаграммы, собранные и связанные вместе, становятся моделью. В SADT дополнительно к правилам синтаксиса диаграмм существуют правила синтаксиса моделей, позволяющие определить границу модели, связать диаграммы в одно целое и обеспечить точное согласование между диаграммами.

SADT-модель является иерархически организованной совокупностью диаграмм. Диаграммы состоят из блоков, каждый из которых может быть детализирован на другой диаграмме. Каждый блок может рассматриваться как отдельный строго определенный объект. Разделение такого объекта на его структурные части (блоки и дуги, составляющие диаграмму) называется **декомпозицией**.

Декомпозиция формирует границы новой диаграммы. Декомпозируемый блок и касающиеся его дуги определяют точную границу диаграммы, представляющей декомпозицию этого блока. Эта диаграмма, называемая **диаграммой-потомком**, описывает все связанное с этим блоком и его дугами и не описывает ничего вне этой границы. Декомпозируемый блок называется родительским блоком, а содержащая его диаграмма — родительской диаграммой.

Принцип ограничения объекта встречается на каждом уровне. Один блок и несколько дуг на самом верхнем уровне используются для определения границы всей системы. Этот блок описывает общую функцию, выполняемую системой. Дуги, касающиеся этого блока, описывают главные управления, входы, выходы и механизмы этой системы. Диаграмма, состоящая из одного блока и его дуг, определяет границу системы и называется **контекстной диаграммой** модели (рис. 2), при этом все, что лежит внутри этого, считается частью описываемой системы, а все, лежащее вне его, образует среду системы.

SADT-модели развиваются в процессе структурной декомпозиции сверху вниз. Сначала декомпозируется один блок контекстной диаграммы на диаграмму, содержащую от трех до шести блоков, затем декомпозируется один или несколько из этих блоков и так далее. Название диаграммы совпадает с названием декомпозируемого блока. Результатом этого процесса является модель, диаграмма верхнего уровня которой описывает систему в общих терминах «черного ящика», а диаграммы нижнего уровня описывают детализированные аспекты и операции системы.

В методологии SADT идентифицируется каждая диаграмма модели посредством номера узла. Номер узла для контекстной диаграммы имеет следующий вид: название модели или аббревиатура, заглавная буква А (Activity в функциональных диаграммах), ноль. Например, номером узла для контекстной диаграммы модели лаборатории АВР «ABP A0». Все другие номера узлов образуются посредством добавления к номеру узла родительской диаграммы номера декомпозируемого блока. Так диаграмма на рис. 3 декомпозирует блок 1 диаграммы на рис. 2 (первый ноль при образовании номера узла принято опускать, поэтому вместо АВР A01 пишется АВР A1).

Как только в результате декомпозиции образуется направленная вниз связь, на диаграмме-потомке формируется ссылка на родительскую диаграмму. В области контекста SADT-бланка (правый верхний угол) автор изображает каждый блок родительской диаграммы маленькими квадратами, заштриховывает квадратик декомпозируемого блока и размещает С-номер родительской диаграммы возле заштрихованного квадрата. Это образует «направленную вверх» (к родительской диаграмме) связь. Метод соединения диаграмм посредством однозначно определенных номеров гарантирует, что именно нужная версия диаграммы станет частью модели.

В SADT принята система обозначений, позволяющая идентифицировать и проверять связи по дугам между диаграммами. Эта схема кодирования дуг – «ICOM» – получила название по первым буквам английских эквивалентов слов вход – Input, управление – Control, выход – Output, механизм – Mechanism. При построении диаграммы следующего уровня, дуги, касающиеся декомпозируемого блока, используются в качестве источников и приемников для дуг, которые создаются на новой диаграмме. После завершения диаграммы ее внешние дуги стыкуются с родительской диаграммой для обеспечения согласованности. Одним из способов такой стыковки может служить присваивание кодов ICOM внешним дугам новой диаграммы согласно согласованным правилам.

Другие диаграммы IDEF0

В дополнение к диаграммам декомпозиции IDEF0 предусматривает использование других видов диаграмм:

- дерево модели
- презентационные диаграммы.

Дерево модели — это обзорная диаграмма древовидной структуры, на которой отражается подчиненность всех диаграмм модели.

Презентационная диаграмма (For Exposition Only) позволяет нарушение любых правил синтаксиса диаграмм IDEF0 в целях выделения важных с точки зрения аналитика частей модели. Например, можно отразить стрелки, относящиеся только к одному функциональному блоку, чтобы прояснить его связь с другими блоками.

6.3.3. Стандарт и методология моделирования IDEF3

Стандарт **IDEF3** был создан для описания последовательностей и логики взаимодействия операций и событий в анализируемой системе. IDEF3 обеспечивает аналитика методологией структурированного подхода и графическим языком для наглядного представления с необходимой степенью детализации знаний об очередности событий и действий описываемого процесса.

Для описания процесса в IDEF3 определены две стратегии и, соответственно, два типа диаграмм:

process-centered strategy — стратегия описания процесса как последовательности выполняемых действий. Диаграммы этого типа получили название Process Flow Description Diagrams (PFDD) — диаграммы потокового описания процесса;

object-centered strategy — стратегия описания процесса как последовательности изменений состояний объекта, над которым

выполняются действия. Диаграммы такого типа получили название Object State Transition Network (OSTN) – диаграммы последовательности изменений состояний объекта.

Описание процесса в IDEF3 может содержать диаграммы PFDD и OSTN или диаграммы какого-либо одного типа. Наиболее известными и широко используемыми являются потоковые диаграммы PFDD. Поэтому в дальнейшем мы ограничимся описанием только этих диаграмм.

Синтаксис диаграмм PFDD моделирования процессов

Для диаграмм PFDD в IDEF3 используется понятие **сценария** в качестве базовой структурной единицы описания процесса.

Сценарий – повторяющаяся ситуация или набор ситуаций, описывающих типичный класс проблем в системе или организации, обстановка или среда, в которой происходит рассматриваемый процесс.

Описание процесса может состоять из одного или нескольких сценариев. Основным назначением сценария является определение контекста описания через присвоение сценарию имени. Именем сценария может быть глагол с поясняющими словами («Оформить заказ на товары», «Проверить пригодность товара») или название совокупности характерных действий («Выполнение последовательности проверок»). Таким образом, сценарий устанавливает ориентацию и границы описания.

Графические элементы, используемые в диаграммах PFDD, включают

- ⊙ модули единицы поведения UOB (Unit Of Behavior),
- ⊙ связи старшинства,
- ⊙ узлы,
- ⊙ модули ссылок и примечаний.

Модули UOB имеют вид прямоугольника со специальными полями как показано на рис.5.

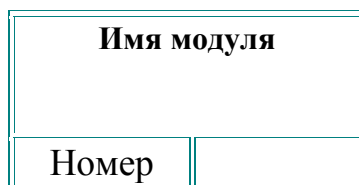


Рис. 5 Модуль UOB

Центральное поле предназначено для уникального (в рамках данного описания) имени модуля, а левое нижнее поле для его порядкового номера. Имя модуля выражается глаголом с поясняющими словами. Порядковый номер модуля определяет его место в диаграммах сценария.

В IDEF3, также как и в IDEF0, для более детального представления действий в описываемом процессе используется декомпозиция функциональных модулей. Декомпозиции модулей представляются на отдельных диаграммах. На рис. 6 представлен пример декомпозиции модулей и принцип формирования их номеров. Для наглядности все модули представлены на одном рисунке, но в IDEF3 описании они будут представлены на трех диаграммах.

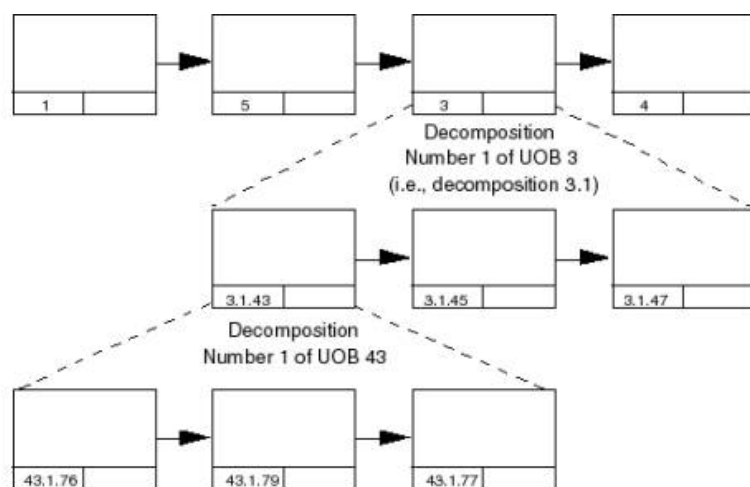


Рис. 6 Декомпозиция модулей в IDEF3 диаграммах.

С точки зрения стандарта IDEF3 каждый функциональный модуль – это обобщенное представление действия или события, которое в различных ситуациях может иметь различные характеристики и свойства. Например, действие «Выписать счет за товар» может выполняться по-разному:

- просто выписать счет;
- проверить наличие нужного количества товара на складе, зарезервировать часть товара, уточнить в отделе маркетинга текущую цену на товар, уточнить в транспортном отделе тарифы на доставку товара, уточнить в бухгалтерии ставки налогов на данный вид товара, согласовать цену и количество товара с отделом продаж, выполнить какие-то другие действия и выписать счет;
- представить процесс выписки счета как набор действий, которые могут выполняться в различном порядке, часть из которых может выполняться или не выполняться в конкретном случае.

В IDEF3 возможна многократная декомпозиция каждого функционального модуля. Как видно на рис. 6, в диаграмме декомпозиции первая цифра номера модуля указывает номер родительского модуля, а вторая цифра — порядковый номер декомпозиции. Третья цифра указывает порядковый номер модуля в диаграммах описания процесса. На рис. 7 представлен пример двух вариантов декомпозиции родительского модуля.

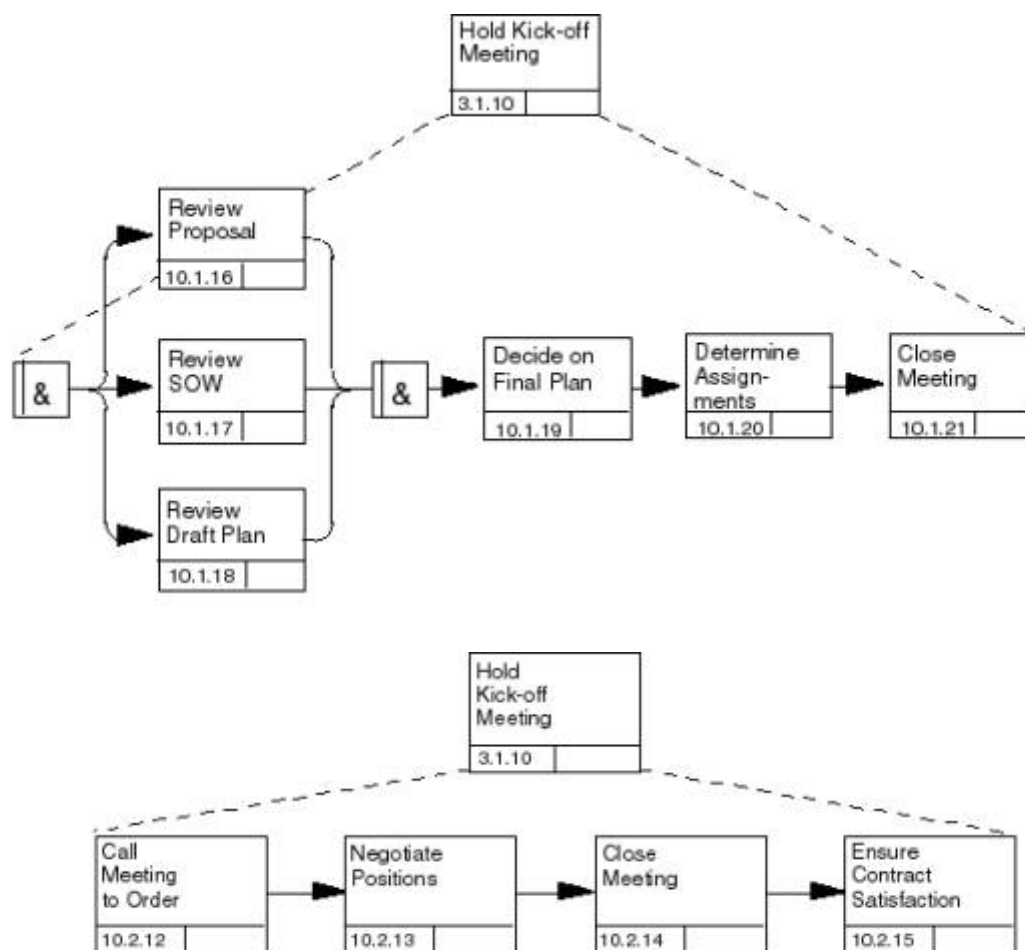


Рис. 7 Два варианта декомпозиции модуля

Связи между блоками

Для представления очередности взаимодействия модулей одной диаграммы используются **связи**, изображаемые соответствующими стрелками. Связи используются, чтобы непосредственно описать отношения между функциональными блоками. Эти отношения, в зависимости от описываемого процесса, могут быть постоянными, временными, логическими, причинными, условными, природными и другими. В большинстве случаев для описания процессов достаточно т.н. «простых связей старшинства», но в IDEF3 аналитику предоставляется возможность самому определить особенности связи,

описав её в соответствующей форме. В качестве базовых в IDEF3 используются два вида связей:

- связи старшинства,
- связи неопределенного типа (тип может быть определен пользователем).

В свою очередь, связи старшинства подразделяются на

- **простые связи старшинства** (Simple Precedence Links) и
- **принуждающие связи старшинства** (Constrained Precedence Links).

Все связи нумеруются. При этом связи старшинства обозначаются символом **PL** (от «precedence link») и порядковым номером, а связи неопределенного типа символом **DL** (от «dashed link») и своим порядковым номером.

Связи неопределенного типа между модулями UOB изображаются пунктирной линией. Они указывают, что между модулями существуют некоторые отношения, но на момент описания процесса они не определены.

Простые связи старшинства обозначаются сплошной однонаправленной стрелкой или дугой (рис. 8.).

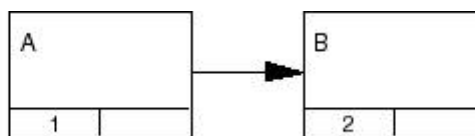
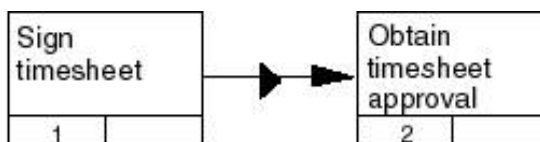


Рисунок 1. Простые связи старшинства (модуль A — предшественник модуля B, а модуль B — приемник модуля A).

Принуждающие связи старшинства добавляют смысловую нагрузку простым связям, определяя порядок предшествования действий. Двойные однонаправленные стрелки на верхней схеме рисунка 9 задают правило – действие модуля 1 должно предшествовать действию модуля 2.



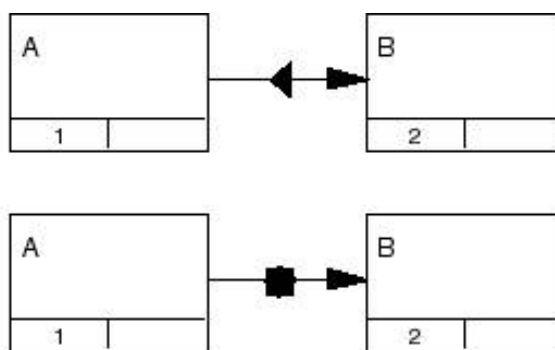


Рис. 9 Принуждающие связи старшинства

Средняя схема на рис. 9 задает правило предшествования действий модуля **В** действиям модуля **А**. Последняя схема определяет, что модули **А** и **В** должны предшествовать друг другу. Таким образом, принуждающие связи старшинства определяют не то, как выполняются действия модулями, а то, как они должны выполняться.

Узлы (Junctions) обеспечивают аналитика инструментом, описывающим возможные ветвления и параллельность выполнения ряда действий в описываемом процессе. Дополнительно узлы усиливают описание временных отношений и отношений очередности выполнения различных частей процесса. В IDEF3 различают **узлы слияния** (Fan-in Junction) и **узлы разветвления** (Fan-out Junction). Примеры изображения узлов приведены на рис. 10.

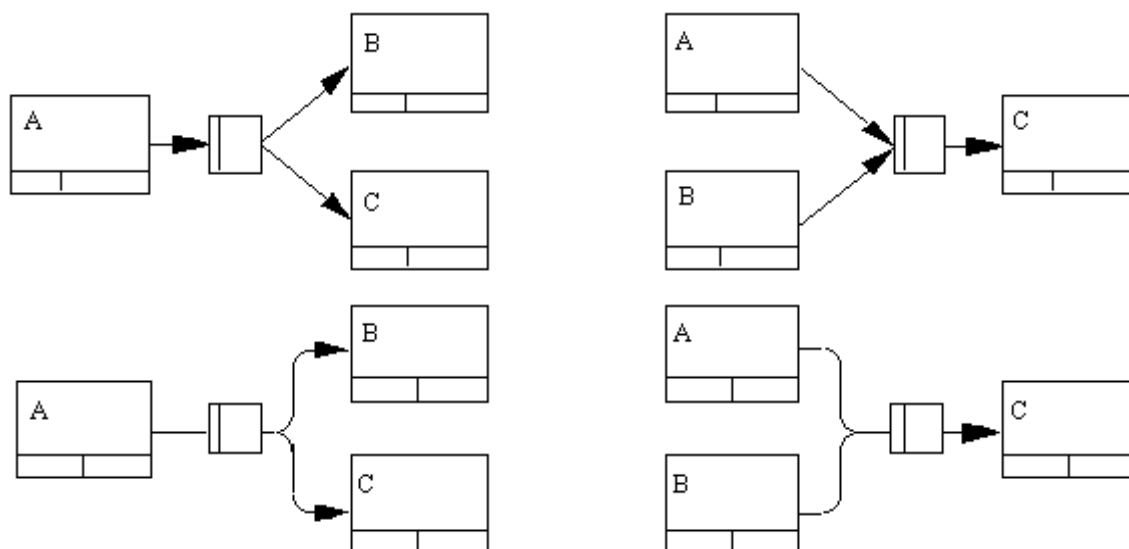


Рис. 10 Узлы слияния и разветвления

Кроме этого, каждый узел может выполнять одну из логических функций **И**, **ИЛИ**, **ИСКЛЮЧАЮЩЕЕ ИЛИ**, быть синхронным или асинхронным. Узлы на диаграммах обозначаются символом **J** (от Junction) и порядковым номером.

В таблице 2 приведены Обозначения и описание описания смыслового значения узлов.

Таблица 2.

Обозначения и описание описания смыслового значения узлов.

Условное обозначение	Наименование логической функции	Смысловое значение при слиянии стрелок	Смысловое значение при разветвлении стрелок
	Асинхронное И	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
	Синхронное И	Все предшествующие процессы завершены одновременно	Все следующие процессы запускаются одновременно
	Асинхронное ИЛИ	Один или несколько предшествующих процессов должны быть завершены	Один или нескольких следующих процессов должны быть запущены
	Синхронное ИЛИ	Один или несколько предшествующих процессов завершаются одновременно	Один или нескольких следующих процессов запускаются одновременно
	ИСКЛЮЧАЮЩЕЕ ИЛИ	Только один предшествующий процесс завершен	Только один следующий процесс запускается

Для описания отношений между модулями UOB в IDEF3 используется термин **активизация** (activation). Под активизациями понимается возможный набор состояний части или всех модулей UOB, удовлетворяющих временным или логическим условиям, заданным схемой, при которых активизируется один или несколько изображенных на схеме модулей. Для простого примера на рисунке 8 условием

активизации модуля В будет завершение действия модулем А. Для наглядного представления во времени очередности включения и выключения модулей UOB сложных схем рекомендуются так называемые **планы активизации** (activation plots), на которых во времени изображается очередность выполнения действий. На рисунке 11–12 приведены примеры схем IDEF3 и соответствующих им планов активизации. Следует особенно подчеркнуть, что замена асинхронных узлов (рис. 11) на синхронные (рис. 12) изменяет временную диаграмму выполнения операций модулями UOB.

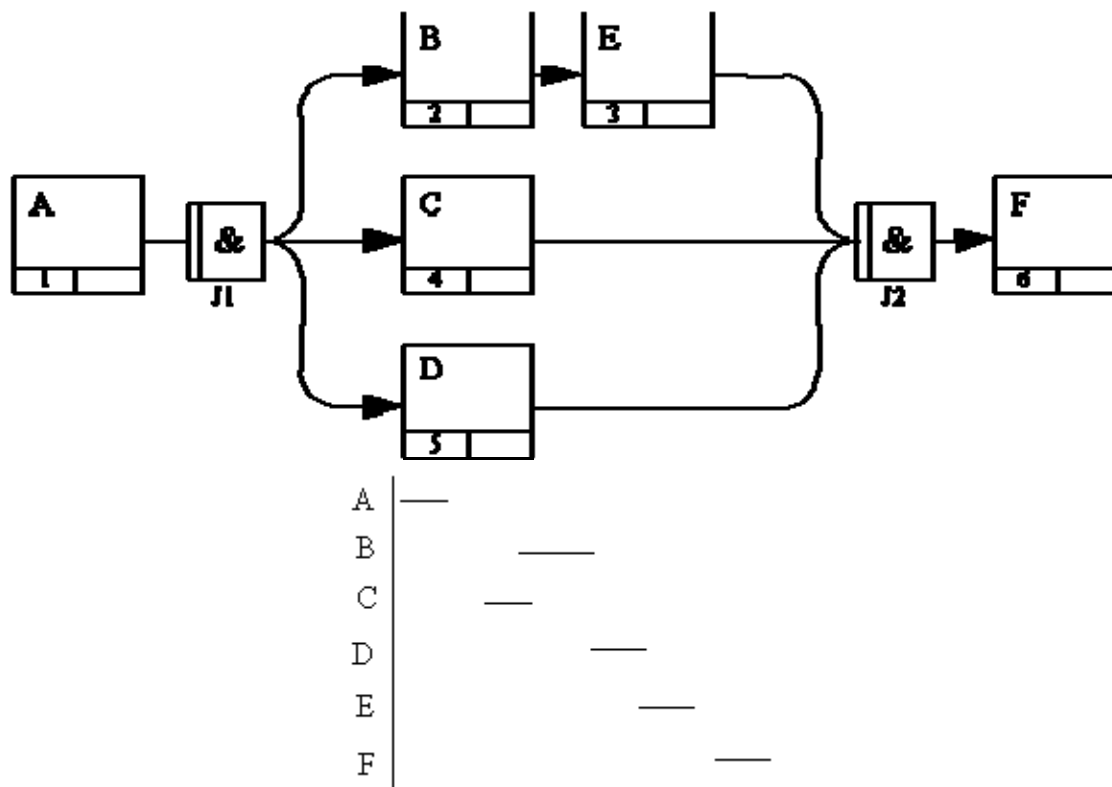
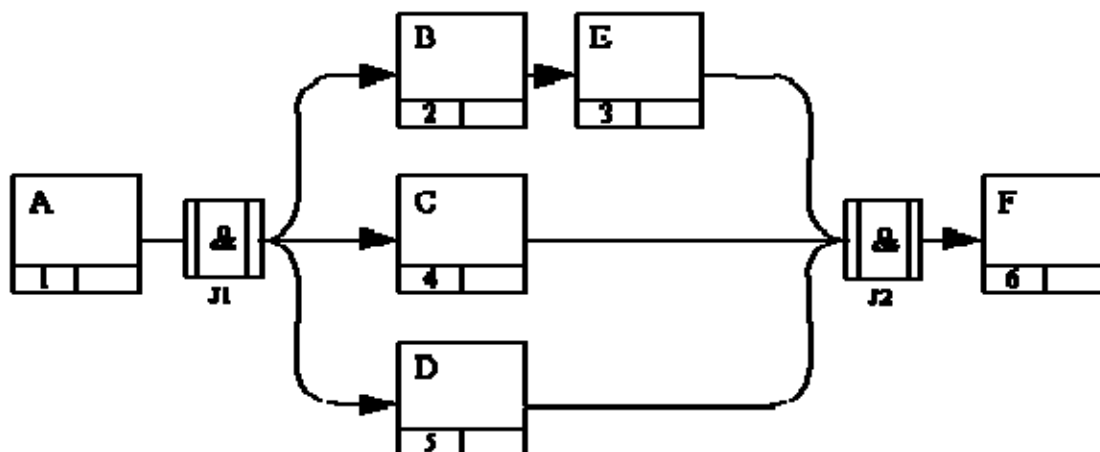


Рис. 11 Схема с асинхронными узлами J1 и J2.+



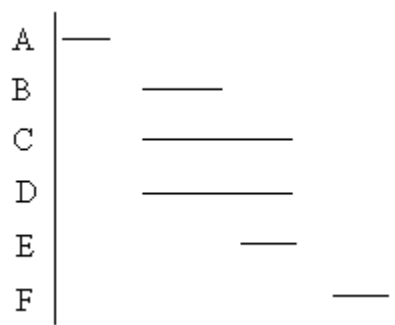


Рис. 12 Схема с синхронными узлами

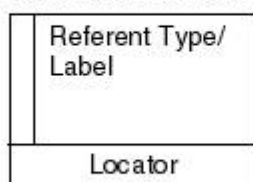
Ссылки

Для обозначения отношений и связей между модулями различных уровней PFDD и OSTN диаграмм и разных сценариев в IDEF3 используются специальные **ссылки** (Referents). Ссылки могут использоваться:

- для обращения к ранее определенному функциональному модулю UOB без повторения его описания;
- для передачи управления или индикации наличия циклических действий при выполнении процесса;
- для формирования ссылок или связей между потоковыми PFDD и объектными OSTN диаграммами.

На рис. 13 представлены условные обозначения двух видов базовых ссылок стандарта IDEF3.

Call and Continue Referent



Call and Wait Referent

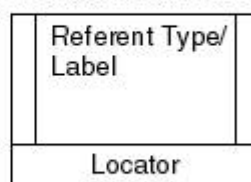


Рис. 13 Условное обозначение ссылок

Использование ссылки «Вызвать и продолжить» (Call and Continue Referent) указывает, что элемент, указанный в ссылке, должен быть активизирован до завершения выполнения действия модулем, к которому относится ссылка. Использование ссылки «Вызвать и ждать» (Call and Wait Referent), указывает, что элемент, указанный в ссылке, должен начать и закончить выполнение действия до завершения действия модулем, к которому относится ссылка.

Каждый из представленных видов ссылки может быть типа «UOB», «SCENARIO», «TS» или «GO TO». Очевидно, что ссылки типа UOB ссылаются на функциональный модуль «UOB», типа «SCENARIO» ссылаются на соответствующий сценарий, типа «TS» (Transition Schematic) – на соответствующую схему, типа «GO TO» – на любой из структурных элементов IDEF3: функциональный модуль, сценарий или узел. При этом в основном поле символа ссылки указывается её тип и через дробь уникальное наименование блока, сценария, схемы или функции узла. В поле «Locator» указывается уникальный номер идентификатор элемента, указанного в ссылке. Пример использования ссылок показан на рис. 14.

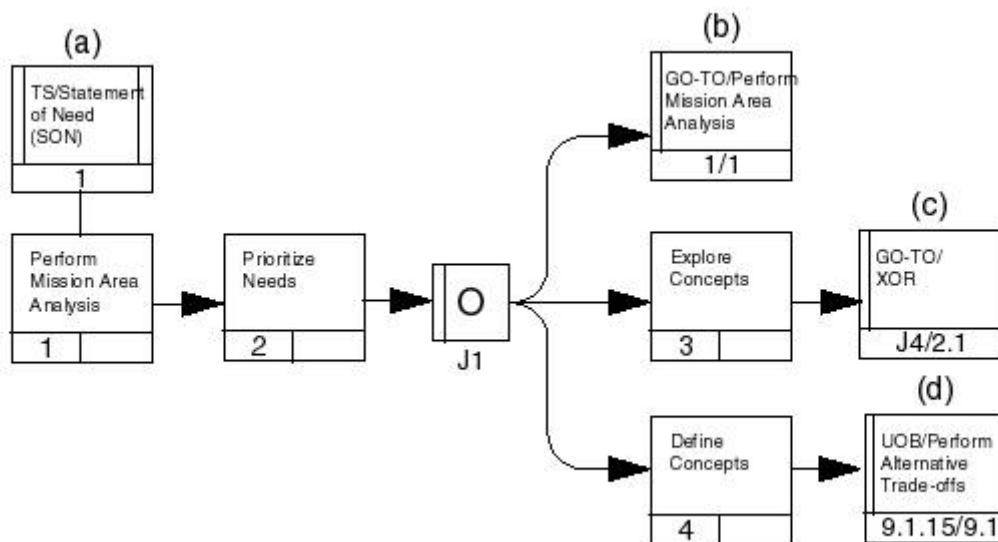


Рис. 14 Пример использования ссылок

6.4. Моделирование данных

Рассмотренная в предыдущем разделе методология SADT в центр рассмотрения ставит *процессы* (обработки данных). Существует и успешно используется двойственный подход, когда в центр рассмотрения ставятся *данные*, обрабатываемые процессами.

В современных информационных технологиях данные хранятся в базах данных и извлекаются оттуда для обработки. Ясно, что база данных — это модель. Возникает соблазн считать, что данные в базе — это модель объектов реального мира. На самом деле это не так, поскольку упускается из вида очень важный и существенный шаг. База данных является не моделью объектов реального мира, а моделью той модели реального мира, которая имеется в голове у пользователя базы данных. Вот этот промежуточный, но очень важный процесс, процесс создания модели той модели данных, которая имеется в голове у пользователя, называется моделированием данных.

Рассмотрим пример – информационную систему отдела кадров. Эта система применяется в реальном мире, и в ней фигурируют такие сущности, как сотрудники, должности, подразделения и т.п. Однако сказать, что запись в таблице «Сотрудники» является моделью конкретного человека было бы явным преувеличением. Такая запись является скорее моделью объекта, который называется «личное дело», причем при моделировании личных дел в базе данных применяется абстрагирование и отбрасывается множество деталей, которые несущественны для информационной системы отдела кадров. Но личное дело, в свою очередь, – это тоже абстрактная модель, отражающая только некоторые свойства конкретного человека. Личное дело может служить исходным материалом для моделирования информационной системы отдела кадров, но практически бесполезно для моделирования человеческого мышления.

Моделирование данных – это мета моделирование объектов реального мира.

Модель данных занимает промежуточное место между несистематизированными представлениями пользователей об объектах реального мира и формальным описанием структур данных в схемах СУБД. Было предложено и используется несколько способов моделирования данных. Наиболее известными являются:

- модель «сущность–связь», предложенная П.Ченом;
- семантическая объектная модель, предложенная Е. Коддом.

Здесь рассматривается модель «сущность–связь», как наиболее популярная в настоящее время.

6.4.1. Диаграммы «сущность–связь»



Модель «сущность–связь» была предложена Питером Ченом (Peter Chen) в 1976 г. в статье «The Entity-Relationship Model – Towards a Unified View of Data». В дальнейшем эта модель многократно раширялась и модифицировалась самим Ченом и многими другими.

Ключевыми элементами модели «Сущность–связь» являются:

- сущности (entity);

- атрибуты (attribute);
- идентификаторы (identifier);
- связи (relationship).

Сущность – это объект, видимый пользователю.

Как правило, рассматриваются не отдельные объекты, а классы (множества) однородных объектов, именно они и называются *сущностями*. Отдельные элементы этих множеств называются *экземплярами* сущностей.

У сущностей есть *атрибуты*, или свойства, которые описывают характеристики сущностей. В модели «сущность–связь считается», что все экземпляры одной сущности обладают одинаковым набором атрибутов (с разными значениями).

Атрибут – это свойство сущности.

Исходное определение модели «сущность–связь» предусматривает использование композитных и многозначных атрибутов.

Экземпляры сущностей имеют *идентификаторы*, то есть наборы атрибутов, идентифицирующих экземпляр или экземпляры сущности.

Идентификатор – это набор атрибутов, идентифицирующих экземпляр.

Различаются *уникальные* идентификаторы, которые идентифицируют один и только один экземпляр, и *неуникальные*, идентифицирующие множества экземпляров.

Взаимоотношения сущностей выражаются *связями*. Связь может иметь атрибуты.

Связь – это отношение между сущностями.

Особое значение в модели имеют три типа бинарных связей, которые называются «один к одному» (рис.), «один ко многим» (рис.) «многие ко многим» (рис.).

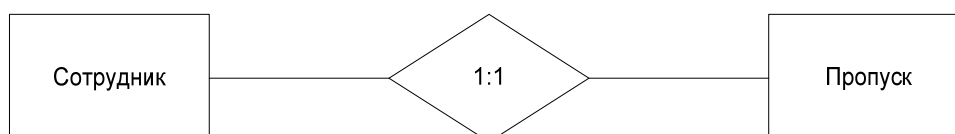


Рис. 15 Связь один к одному.

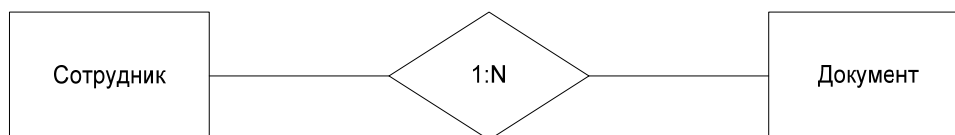


Рис. 16 Связь один ко многим

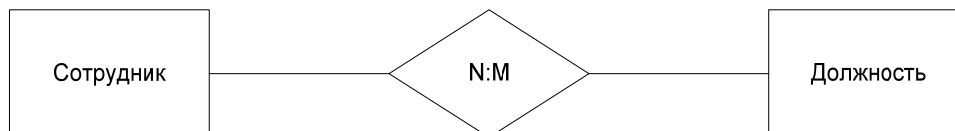


Рис. 17 Связь многие ко многим

Число экземпляров сущности, участвующих в связи, называется кардинальным числом. В типе связи, указывается, фактически верхняя граница кардинального числа.

В целом модель «сущность–связь» представляется в виде совокупности диаграмм, которые называются диаграммы «сущность–связь», или ERD (entity-relation diagram). Такие диаграммы стандартизованы, но не слишком жестко. Здесь используется оригинальная нотация, предложенная Ченом.

Сущность обозначается прямоугольником, внутри которого указывается имя сущности, а связь – ромбом, в котором указываются тип связи.

Различаются

- ⊙ слабая (weak) сущность, которая может существовать, только если есть некоторая другая сущность;
- ⊙ сильная (strong) сущность, которая может существовать, независимо от других сущностей.

Слабые сущности принято обозначать прямоугольниками со скругленными углами.

Чтобы показать нижнюю границу кардинального числа, используются разные способы. Например, если нужно подчеркнуть, что экземпляр обязан существовать, то есть минимальное кардинальное число равно 1, ставят черточку на линии связи возле сущности. Если же нужно показать, что экземпляр может, но не обязан существовать ставят овал. Введенные обозначения иллюстрированы на рис. 18.

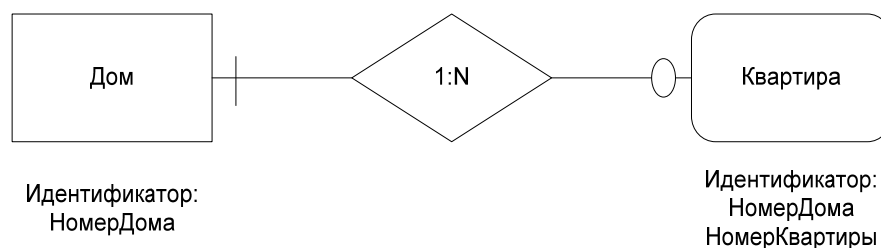


Рис2. 18 Слабые и сильные сущности

Атрибуты сущностей показывают по-разному, например, в виде отдельного списка или помещая их в овалы, связанные с прямоугольниками сущностей.

Для передачи отношения обобщения между сущностями, применяют так называемые подтипы сущностей, являющиеся частным случаем механизма наследования в объектно-ориентированном программировании. На рис. 19 приведен пример, иллюстрирующий введенные понятия.

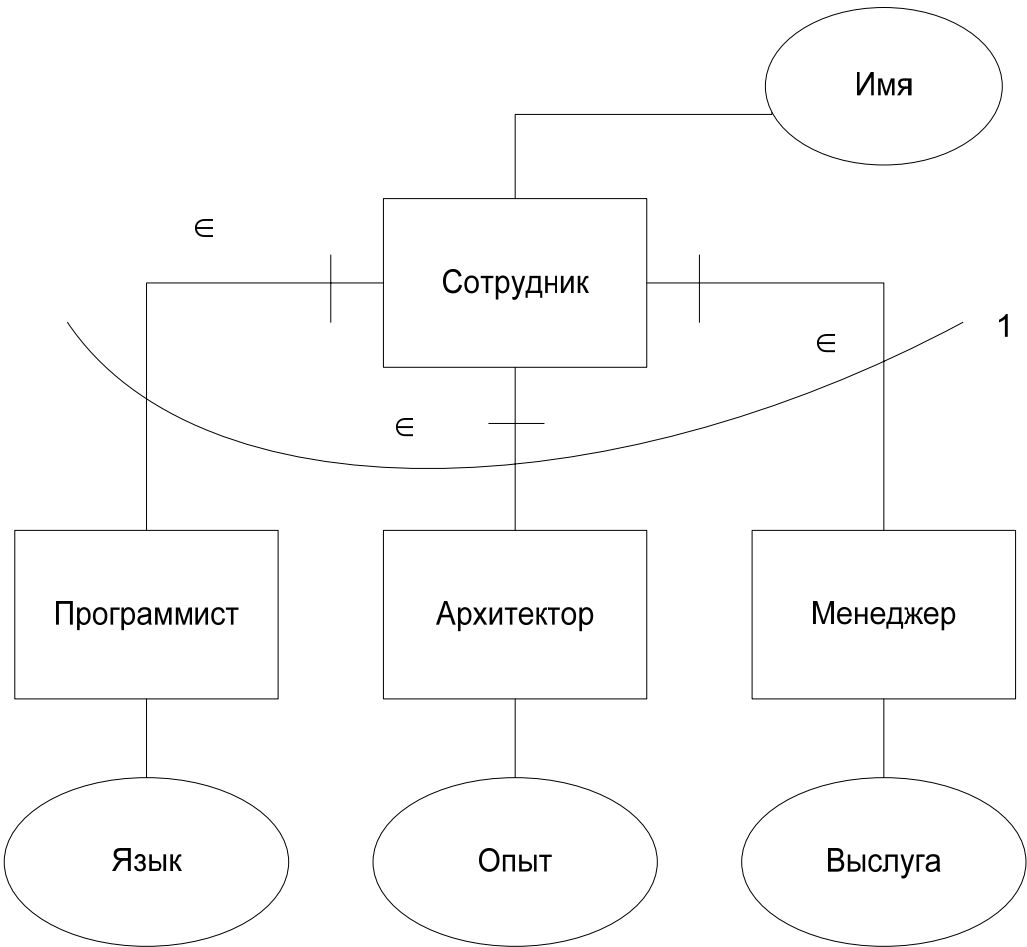


Рис. 19 Подтипы сущностей.

Закончим обсуждение модели «сущность-связь» приведя пример, простой, но законченной модели.

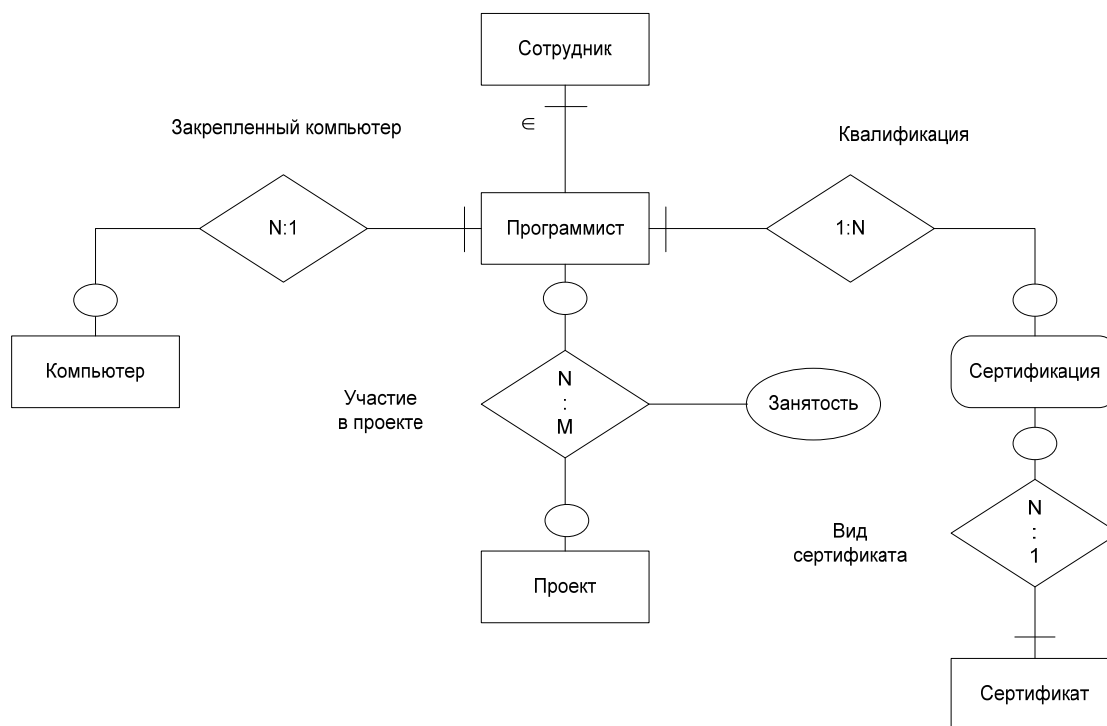


Рис. 20 Пример диаграммы «сущность-связь».

В этой модели указано, что сущность «программист» является частным случаем сущности «сотрудник»; за каждым программистом закреплено некоторое количество компьютеров (может быть и 0), но не бывает «бесхозных» компьютеров, программисты участвуют в проектах, причем учитывается занятость каждого программиста в каждом проекте; каждый программист квалифицирован в некоторых областях, что подтверждается сертификатами.

6.4.2. Выражение модели сущность-связь средствами UML

Модель «сущность-связь» имеет долгую историю и используется во множестве CASE-средств и СУБД. Ввиду столь широкой распространенности, при разработке Унифицированного языка моделирования UML были приняты специальные меры, чтобы модели UML позволяли сделать все, что можно сделать в ERD.

На рис. 21 приведена диаграмма классов UML, в которой отражается в точности та же модель данных, что и на диаграмме «сущность-связь» на рис. 20.

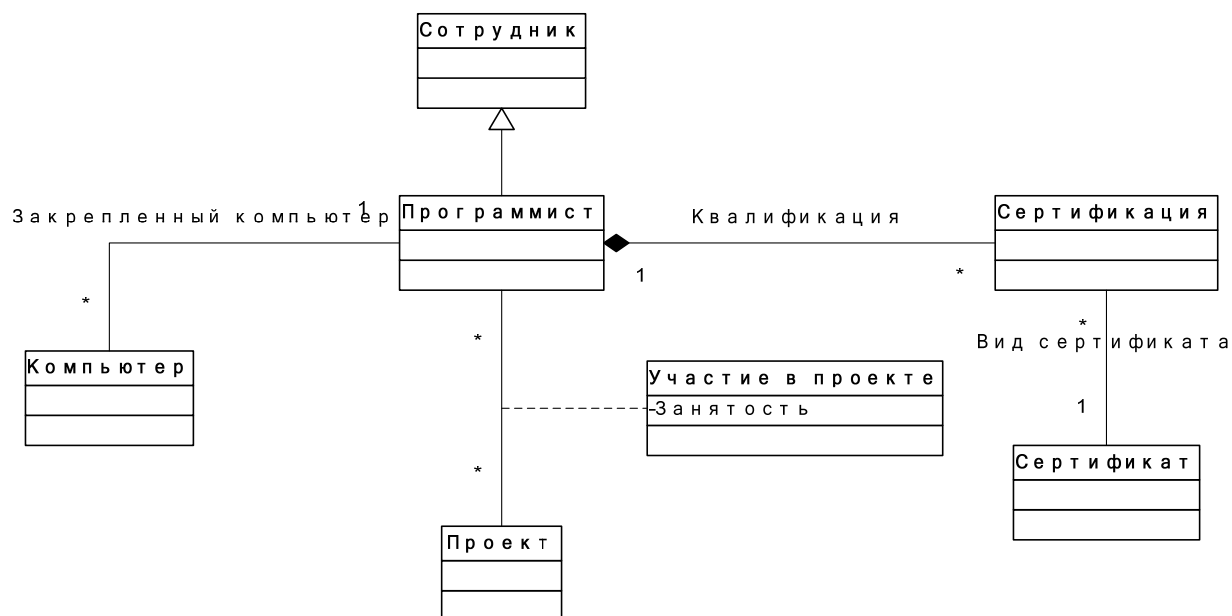
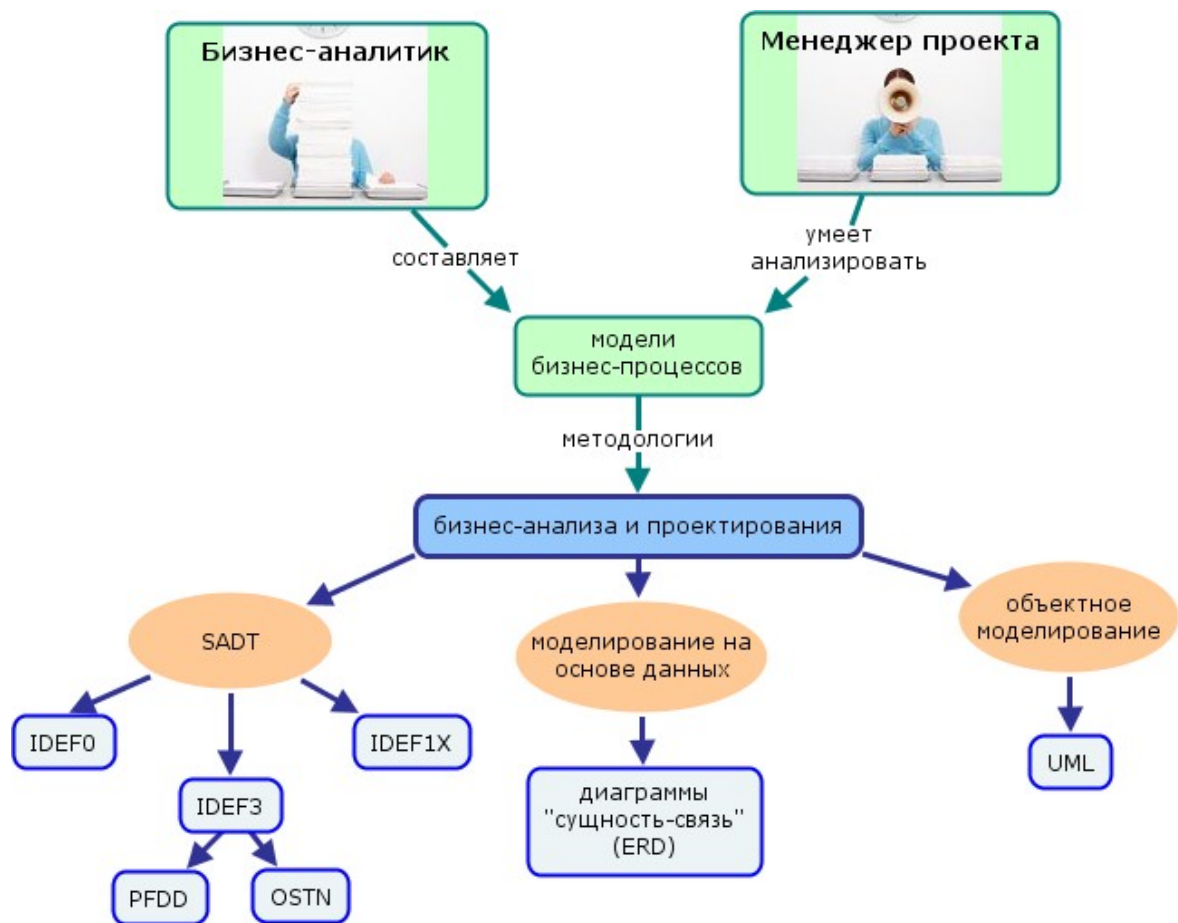


Рис. 21 Эквивалентная модель на UML

6.5. Заключение

В настоящее время разработано значительное количество стандартов, позволяющих представить в виде удобной модели бизнес-процессы предприятия с целью дальнейшего проектирования корпоративной информационной или управляющей системы. К наиболее потребительным из таких стандартов являются стандарты IDEF0, IDEF3, являющиеся частью методологии структурного анализа и проектирования SADT. Все стандарты такого рода, как правило, позволяют представить модель предприятия в виде удобоиспользуемого набора диаграмм, представляющей деятельность предприятия с определенной точки зрения. Диаграммы IDEF0, IDEF3 являются далеко не единственными типами диаграмм, применяемых для бизнес-анализа и проектирования. Например, для систем, основанных на данных, популярно использование диаграмм «сущность-связь». В последнее время также все большее распространение приобретает объектное моделирование, основанное на диаграммах UML. Менеджеру проекта по разработке корпоративных информационных или управляющих систем (например, систем управления производством, систем автоматизации документооборота) следует разбираться в используемых бизнес-аналитиках методологиях и стандартах бизнес-анализа и проектирования, и по меньшей мере уметь читать и анализировать соответствующие диаграммы.

6.6. «Карта памяти» по теме



6.7. Список использованной и дополнительной литературы

1. Д. Марка, Р. Макгоуэн. Методология структурного проектирования SADT. М.: Метатехнология, 1993.
2. Черемных С.В. и др. Структурный анализ систем: IDEF-технологии. М.: Финансы и статистика, 2003.
3. Крёмке Д. Теория и практика построения баз данных. СПб: Питер, 2003

Тема 7. Кадровый менеджмент

7.1. Введение

Главным достоянием компании, работающей в сфере информационных технологий, и тем более в сфере разработки программного обеспечения, являются ее сотрудники. Поэтому ключом к успеху такой компании является грамотное управление персоналом.

Изучив учебный материал данной темы, Вы:

- узнаете или пополните свои об основных принципах и методах управления персоналом;
- узнаете или пополните свои знания об особенностях подбора, найма и увольнения работников в компании, работающей в сфере информационных технологий;
- познакомитесь с особенностями развития команды проекта.

В рамках темы рассматриваются следующие учебные вопросы:

- Кадровая политика фирмы.
- Основные принципы и методы управления персоналом.
- Персонал проекта по разработке программного обеспечения.

7.2. Кадровая политика фирмы

Главное достояние компании, работающей в области информационных технологий – это ее сотрудники.

Успешная деятельность компании невозможна без четко сформулированной и последовательно проводимой **кадровой политики (политики управления персоналом)**. В это понятия объединяются:

- методы отбора, стимулирования и развития персонала;
- формирование корпоративной культуры и командного духа.

Основная цель кадровой политики – получить максимальную отдачу от инвестиций в персонал через построение системы управления персоналом, мотивирующей сотрудников к достижению целей компании.

Приоритетами в области кадровой политики современной компании, работающей в области ИТ, являются:

- создание атмосферы, основанной на открытости, взаимном уважении и сотрудничестве;
- развитие системы мотивации персонала, позволяющей каждому сотруднику рассчитывать на карьерный рост и

вознаграждение, соответствующие его профессиональному уровню и личному вкладу в дело компании;

- совершенствование социальной защиты сотрудников;
- предоставление сотрудникам возможностей обучения и повышения квалификации.

Кадровая политика – это совокупность методов отбора, стимулирования и развития персонала, мотивирующая сотрудников на достижение целей компании.

7.3. Основные принципы и методы управления персоналом

Кадровая политика реализуется путем применения определенных методов управления персоналом. Методы следует применять, придерживаясь строгих принципов, в противном случае неизбежны проявления необъективности, субъективизма, нелояльности и другие отрицательные последствия вплоть до саботажа и вредительства.

С другой стороны, кадровая политика – это залог здоровья и основа роста компании. Все составляющие кадровой политики влияют друг на друга и на компанию в целом.

На рис.1 представлена схема, иллюстрирующая взаимосвязь кадровой политики и развития компании.



Рис. 1 Составляющие кадровой политики

Основными составляющими управления персоналом являются следующие:

- ✓ анализ работы персонала,
- ✓ планирование потребности в персонале,
- ✓ подбор, наём и увольнение работников,
- ✓ мотивация и вознаграждение (включая вопросы заработной платы и карьеры).

Рассмотрим подробнее все эти вопросы за исключением вопросов мотивации и вознаграждения (об этом речь шла в теме «Психология в менеджменте»). Отметим при этом, что управление персоналом в компании по разработке программного обеспечения в основном похоже на управление персоналом в организациях проектного типа, однако, имеется и несколько специальных аспектов, в особенности в вопросах подбора кадров.

7.3.1. Анализ работы персонала

Для осуществления эффективной кадровой политики необходимо ***регулярно анализировать работу имеющегося персонала***. Для анализа необходимо предварительно собрать достоверную информацию. Процедура сбора такой информации называется ***«оценкой персонала»***.

Уточнение термина «оценка персонала»

Здесь необходимо подчеркнуть, что термин «оценка персонала» означает, по меньшей мере, три различные вещи. В английском языке для этих целей используются разные термины, а русском – один, поэтому, чтобы не было путаницы, необходимо использовать уточняющие прилагательные. Рассмотрим понятие «оценка персонала» подробнее.

- ◎ **Оценка компетенции (Assessment)** – этот термин используется для обозначения процедуры, связанной с оценкой собственно сотрудника, его характера, знаний, навыков, умений, способностей и т.д.
- ◎ **Оценка достижений (Appraisal)** – используется для обозначения процедуры, предполагающей оценку результатов деятельности сотрудника, и представляет собой подведение итогов работы сотрудника за определенный период времени, не затрагивая его личных характеристик.
- ◎ **Оценка должности (Evaluation)** – этот термин используется для обозначения процедуры, предназначенной для определения относительной ценности конкретной должности для компании, безотносительно к тому, кто именно занимает должность и

насколько этот сотрудник справляется с должностными обязанностями.

Проводя оценку персонала, необходимо четко определить, что именно оценивается и применять соответствующие методы. Для оценки персонала применяются самые разные приемы, все множество которых невозможно описать. Ниже перечислены некоторые наиболее распространенные:

1. Аттестация.
2. Анкетирование.
3. Тестирование.
4. Собеседование.

Типичные ошибки и правила организации оценки персонала

Проводя оценку персонала, следует стараться избегать следующих типичных ошибок.

Ошибка цели.

Выбор подхода и инструментария оценки персонала зависит, прежде всего, от целей, которые должны быть с помощью этой оценки достигнуты. Ошибка цели возникает, если цель не ставится или, хуже того, утаивается или искажается.

Например, если речь идет об оценке компетенций, т.е. таких характеристик, как знания, навыки, способности, интересы и тому подобное, то целью использования полученных результатов может являться решение двух достаточно разноплановых задач:

- ! во-первых, понимание того, какими компетенциями обладает сотрудник, может быть использовано для назначения его на соответствующую должность или приема на работу;
- ! во-вторых, оценив уровень компетенций, необходимый для выполнения текущей деятельности сотрудника, мы можем определить, в каких знаниях и навыках сотрудник испытывает определенный дефицит.

Вполне очевидно, что *для оценки потенциала* (предполагаем зачисление в кадровый резерв или работу на другой должности) необходимо использовать *специальные инструменты*, такие как:

- тесты способностей,
- личностные вопросники, разработанные для профессиональной сферы,
- интервью по компетенциям,
- упражнения на основании рабочих ситуаций.

Однако если этот же специальный инструментарий используется для определения потребностей в обучении, то может возникнуть ряд очень серьезных проблем, таких как сопротивление сотрудников, страх и чувство неуверенности, снижение мотивации. Чему удивляться, если часто вместо заявленной цели определения потребностей в обучении, компания по итогам оценки персонала проводит массовое сокращение штатов.

Ошибка метода.

Такая ошибка возникает, если для оценки персонала применяется неподходящий метод или же если метод подходит, но применяется неправильно.

Например, одной из основных ошибок метода при оценке компетенции, является ***отсутствие модели компетенции*** соответствующей должности или группы должностей. Фактически, в этом случае невозможно провести оценку компетенции ввиду отсутствия базы для сравнения! При этом инструменты, с помощью которых проводится оценка, работают нормально, и дадут какой-то результат. Используя специально предназначенные для подобной оценки методы, мы получим описание поведенческих характеристик человека.

Так, например, по всем шкалам, которые представлены в профессиональном вопроснике, мы можем сказать о предпочтениях человека, о том, какие качества он в себе отмечает. А вот дать ответ на вопрос: хорошо это или плохо для данной должности – увы, не можем, поскольку нет шкалы, критериев, с которыми мы могли бы сопоставить результат. Для примера попробуйте ответить на вопрос, необходима ли такая компетенция как «высокая персональная самостоятельность и инициатива» для тестера?

Следующей не менее распространенной ошибкой этой группы можно назвать ***несоответствие инструмента целям оценки***.

Например, проведение «деловой игры», с участием большого количества сотрудников, может быть использовано для целей обучения. Но если при этом ставится еще цель оценить сотрудников в ходе игры, то затея обречена на провал. Преследуя такие диаметрально противоположные цели, как обучение и оценка, нельзя достичь ни одной из них. Любая учебная ситуация изначально предполагает принцип безопасности, то есть приобретая знания и навыки, мы, в том числе, учимся на ошибках. Глупо было бы наказывать за ошибки в такой ситуации. А как будут вести себя «обучающиеся», зная о том, что их еще и оценивают? А если они не будут знать, что их оценивают, то нормальное поведение, уместное в учебной ситуации, например, частые вопросы заинтересованного в своем развитии сотрудника, может быть легко истолковано десятком разных способов, начиная от высокой

оценки «инициативности» до низкой оценки по шкале «понимание и решение проблем».

Третья ошибка данной группы заключается в **некорректности самих инструментов, с помощью которых проводится оценка**. Например, соревнования типа олимпиад по программированию (решение искусственных задач на скорость) прекрасно подходят для выявления личных способностей к концентрации, знания специальных разделов дискретной математики и т.д. Но при этом такие соревновательные ситуации совершенно не годны для оценки компетенций, т.к. имеют весьма далекое отношение к бизнесу и кругу непосредственных задач сотрудников. Можно также говорить о некорректности использования многих популярных психометрических инструментов, таких как вопросники, взятые непосредственно из западных источников. Такой инструмент, как вопросник, должен учитывать специфику страны, языка и культуры. Просто перевода недостаточно. Требуется кропотливая работа по нормированию, пересчету значений шкал, изменению содержания, в случае необходимости. Проще говоря, необходимо приведение таких инструментов к специфике нашей среды, условиям нашего бизнеса.

Юридическая ошибка.

Юридическая ошибка возникает, когда при оценке персонала **нарушаются законы**.

Например, во многих странах запрещено использование некоторых психологических методик для отбора кандидатов. Безусловно, с помощью таких методик грамотный психолог может получить ответы на многие вопросы, связанные с текущим состоянием психики человека (заметим, не с уровнем развития компетенций). Однако когда речь идет о выборе кандидата на вакантную должность, вполне очевидно, что такой подход к оценке легко может быть подвергнут сомнению не только со стороны самого кандидата (отказано в приеме на работу на основании результатов клинического вопросника), но и со стороны судебных инстанций.

Пока что в условиях нашей реальности кандидаты, получившие отказ, редко обращаются в суд с жалобой на несправедливое к себе отношение, но это не значит, что они и впредь не будут этого делать.

Вторая проблема, связанная с легитимностью инструментов, связана с повальным **нарушением авторских прав** компаниями, которые используют соответствующие методики оценки персонала. Все знают, что красть нехорошо, все знают, что нехорошо пользоваться краденым, но не все знают, что кандидат может на законных основаниях поинтересоваться: «А на каком основании Вы используете данную методику? А есть ли у Вас соответствующие права и лицензии, а обучен ли и сертифицирован ли тот специалист, который проводит тестирование?» и так далее. И если удовлетворительного ответа на эти

вопросы кандидат не получит, то сможет на совершенно законных основаниях оспорить результаты оценки, а вот руководству компании придется держать ответ не только за оценку, но и за нарушение авторских прав в том числе.

Техническая ошибка.

К сожалению, ошибок этой группы избежать труднее всего. Ведь оценку проводят люди, а людям свойственно ошибаться.

Если рассматривать эти ошибки в логической последовательности, то первыми имеет смысл упомянуть ***ошибки, допускаемые в процессе администрирования оценочных процедур***, например, пренебрежение необходимостью создания одинаковых условий для всех оцениваемых, так как только в этом случае мы можем говорить о сравнимости полученных результатов.

Далее следуют ошибки, связанные с ***реализацией самой процедуры***. Это, например, отсутствие достаточного количества подготовленных наблюдателей.

Другая группа ошибок – ***ошибки наблюдения*** (быстрый переход к выводам, суждение по внешнему впечатлению, следование укоренившимся стереотипам).

Существуют, наконец, ***ошибки обработки данных и вычислительных процедур подведения итогов***. Снизить влияние этих ошибок на эффективность процедуры оценки персонала, можно только постоянно совершенствуя у наблюдателей необходимые навыки.

Преодолеть технические ошибки поможет специализированное обучение, а это, как хорошо известно, требует и больших временных издержек, и достаточно солидных финансовых затрат. Эти затраты будут оправданы только если компания собирается использовать оценку персонала в качестве регулярной процедуры.

Подводя итоги, укажем условия правильно организованной процедуры оценки персонала, с помощью которой компания сможет принять корректные управленческие решения:

- ясность целей, которых собирается достичь компания оценкой персонала;
- соответствие выбранной процедуры оценки целям;
- наличие инструментов и методов, предназначенных специально для проведения выбранной процедуры оценки;
- четкое юридическое сопровождение процедуры оценки персонала;
- наличие квалифицированных специалистов, обученных использованию соответствующих инструментов оценки.

7.3.2. Планирование потребности в персонале

Планирование потребности в персонале является одним из самых важных и в то же время трудных видов планирования. Планируя потребности в персонале, мы должны ответить на вопрос: «сколько и каких сотрудников нужно компании сейчас, через месяц, через год, чтобы достичь целей организации?». Ответить на этот вопрос нелегко даже для сравнительно консервативных государственных учреждений, тем более это трудно для компаний, разрабатывающих программное обеспечение, где ситуация меняется буквально ежедневно, причем самым непредсказуемым образом.

Рассмотрим два крайних случая.

✓ **Абсолютно стабильный бизнес.** Если цели организации определены и стабильны, производственный процесс налажен и не меняется, процессы установлены и измерены, то потребности в персонале могут быть определены очень точно.

Для этого достаточно:

- определить планируемый объем производства, который известен, поскольку известны цели организации,
- определить качественные потребности в персонале (соотношение по ролям), что также можно сделать, поскольку процесс стабилен и соотношение ролей не меняется,
- определить количественные потребности в сотрудниках, что сделать легко, поскольку технологии постоянны и производительность труда известна.

На первый взгляд кажется, что в области разработки программного обеспечения такая ситуация нереальна, однако, это не так. Например, так работают компании, которые занимаются аутсорсингом в режиме «body leasing». Фактически, такие компании предоставляют своих программистов в аренду на условиях почасовой оплаты по стабильному тарифу. Определение требований и управление проектом ведется на стороне заказчика, исполнитель предоставляет оборудованные рабочие места и инфраструктуру для работы. В последнее время такого рода услуги стали весьма популярными под достаточно громким названием ODC (Outsourcing Dedicated Center). Их суть заключается в том, что компания организует выделенный центр аутсорсинга для конкретного (обычно достаточно крупного) заказчика, подбирает для него персонал, иногда дополнительно обучает его, создает инфраструктуру, гарантирует юридическое и бухгалтерское обеспечения функционирования центра. Однако, в дальнейшем управление проектами в этом центре осуществляется непосредственно

заказчиком (компания может даже не владеть полной информацией об этих проектах).

✓ **Абсолютно стабильная команда.** Противоположной крайностью является программирующая организация, обычно небольшая, представляющая собой команду единомышленников, объединившихся для совместной деятельности. Такая команда обычно берется за любые посильные заказы и не знает, что будет делать через год-два. Как правило, состав команды является очень стабильным все (обычно, недолгое) время ее существования. При этом никакого планирования потребности в персонале не производится, «что имеем, то и используем». Производится определение целей исходя из имеющегося персонала, то есть не цели определяют потребности в персонале, а персонал, определяет возможные цели.

Большинство организаций находится в промежутке между указанными крайностями. Можно уверенно предсказать, что потребности в персонале будут меняться с течением времени, но предсказать величину и характер изменений достаточно трудно. В таком случае можно порекомендовать планировать не окончательную цель, а саму возможность изменения потребности в персонале. Другими словами, рекомендуется определить, сколько сотрудников и в какие сроки организация сможет переобучить, если возникнет такая потребность, сколько сотрудников и в какие сроки организация сможет нанять, сколько сотрудников и каким образом организация сможет уволить без потери способности к функционированию.

Планирование потребности в персонале принято разделять на *стратегическое и текущее*. В отношении персонала у каждого из этих видов планирования возникают свои специфические задачи, свои приоритетные направления.

- **Стратегическое планирование** представляет собой организационный механизм разработки и реализации стратегии, с помощью которой руководство организации концентрирует ресурсы в производствах, сулящих наибольшие перспективы в будущем. Стратегическое планирование направлено на удовлетворение потребностей в персонале, которые возникнут через несколько лет, и поэтому подразумевает отслеживание тенденций развития персонала, а также определение стратегической потребности в трудовых ресурсах, возникающей в процессе реализации тех или иных глобальных производственных планов. В рамках стратегического

планирования персонала должно быть уделено особое внимание вопросам определения долгосрочных потребностей в кадрах той или иной квалификации, выбору форм финансирования образования, разработки программ внутрифирменного обучения с тем, чтобы стратегические цели организации были достигнуты в поставленные сроки и с наибольшей компетентностью. Основной направленностью стратегического планирования персонала становится тем самым готовность к крупным, долгосрочным и дорогостоящим мероприятиям в области подготовки и развития трудового потенциала предприятия.

- **Текущее планирование** концентрирует внимание на небольших колебательных изменениях в трудовых ресурсах. Однако оно не сводится только к конкретизации и детализации стратегического плана. Наряду с детализацией и конкретизацией стратегического плана, текущее планирование персонала имеет свою собственную сферу применения, связанную, прежде всего с тем, что персонал представляет собой как бы живой организм, какие-то клетки которого умирают, а другие вновь рождаются. В рамках текущего планирования рассматриваются вопросы увольнений, выходов на пенсию, в декретные и творческие отпуска, текучести кадров и т.п. Основной особенностью текущего планирования является его оперативность, т.е. обеспечение готовности к быстрой реакции на небольшие изменения. Фактически, текущее планирование — это планирование замещения выбытия трудовых ресурсов. Текущее планирование связано с оперированием цифрами, полученными на основе анализа краткосрочных потребностей организации и характеристиками, отражающими колебания численности рабочей силы. Эти цифры, конечно же, не исчерпывают текущей план, они представляют собой лишь количественную его сторону. Затем необходимо разработать мероприятия с тем чтобы с наименьшими затратами и наибольшей эффективностью набрать недостающее количество работников. В плане должны быть рассмотрены и финансовые аспекты вопроса, так как все мероприятия по отбору кандидатов (в особенности реклама и специальное тестирование) стоят немалых денег.

Основным элементом планирования является **выявление потребностей в трудовых ресурсах**. Наряду с достаточно понятными вопросами замещения выбытия персонала (определение среднего уровня текучести кадров, количество выходов на пенсию и в долгосрочные отпуска), здесь существуют специфические процедуры

определения потребностей, отражающих развитие бизнеса. Именно этот аспект является наиболее сложным и интересным. Существует множество методов определения будущих потребностей в персонале. Основными из них являются:

- **Экстраполяция.** Этот метод планирования трудовых ресурсов основан на продлении в будущее существующих тенденций, связывающих показатели производства и количество персонала (прежде всего соотношение между объемом работ и численностью занятых). Для успешного применения экстраполяции необходимо знать тенденции и соотношения (получаемые из текущей статистики), на которые можно будет опираться в соответствующих вычислениях. Это возможно только в том случае, когда основная деятельность организации и производительность труда работников остаются неизменными. Кроме того, цифры, полученные путем экстраполяции, должны быть скорректированы с учетом планирования мероприятий по повышению производительности труда, если таковые имеются. В ряде случаев корректировка должна учитывать и емкость рынка. Если рынок уже заполнен, прием на работу дополнительных работников может привести к снижению производительности уже работающих.
- **Балансовый метод.** Данный метод очень близок к экстраполяции, но отличие заключается в том, что планирование персонала в данном случае осуществляется для нескольких связанных между собой видов деятельности в организации. Баланс между видами деятельности может быть просчитан как по продукции, так и по персоналу. Особенно эффективен этот метод для расчета количества вспомогательного или обслуживающего персонала. Основной недостаток экстраполяции здесь сохраняется: те тенденции и соотношения, которые распространяются на будущий период деятельности организации, должны уже существовать; т.е. данный метод хорошо работает применительно к неизменным условиям работы и видам деятельности.
- **Экспертный опрос.** Привлечение экспертов используется для определения потребностей в трудовых ресурсах, прежде всего при переходе на новую продукцию. В качестве экспертов могут выступать как менеджеры по персоналу, так и технические специалисты. Менеджеров спрашивают, потребуются ли им дополнительные работники для выполнения новых заданий, и если да, то в каком количестве. Такой опрос имеет целью не столько абстрактное построение структуры будущей занятости, сколько

конкретное распределение заданий между коллективами с соответствующей их комплектацией.

- **Моделирование.** Данный метод также применяется для определения потребностей в рабочей силе в изменяющихся условиях. Модель может быть составлена как на основе требований технологии (масштабов работ, соответствия между количеством работников и видом оказываемой услуги, количеством и спецификой потоков информации и др.), так и на основе социотехнического подхода, учитывающего запросы производственной и социальной подсистем организации (обогащение труда, развитие персонала, ротация кадров и др.). При реализации новых проектов могут быть использованы модели линейного программирования, минимизирующие потери, как от избыточного, так и от недостаточного количества работников.

7.3.3. Подбор и наём работников

Выбрать надежного и компетентного работника – дело нелегкое, требующее времени, усилий, разностороннего подхода и, что гораздо важнее, четкого представления того, каким должен быть претендент на определенную должность.

Прием на работу – непростой процесс, требующий объективности, ответственности, профессионализма, умения разбираться в людях и идти на компромисс. Лучше не торопиться и не делать поспешных выводов, чем потом ломать голову над тем, как избавиться от нерадивого работника или каким образом уладить конфликт на рабочем месте.

Процесс приема на работу является двусторонним. С одной стороны, работодатель оценивает, насколько кандидат соответствует предъявляемым требованиям. С другой стороны, кандидат, предлагающий свои знания и способности, ищет работу, которая может удовлетворить его потребности.

Данная проблема может быть решена, если требования обеих сторон частично или полностью совпадают. Поэтому процесс приема на работу основывается на определенном компромиссе и возможности договориться.

Источники набора кадров

Имеются два источника набора кадров: внутренний и внешний.

- ☉ **Внутренний** источник подразумевает заполнение вакансий путем привлечения работников, уже работающих в организации. Если компании нужны работники не на постоянную работу, а на

короткий срок или подразумевается выполнение какого-то конкретного объема работы, то можно использовать совмещение должностей. При этом необходимо ввести продуманную систему поощрений помимо надбавок за сверхплановую работу. Также важны рациональное планирование и обеспечение возможности качественного выполнения сотрудником необходимого объема работы. Если недостаток работников ощущается на должностях высокого уровня, следует проанализировать, имеются ли внутри организации возможные кандидаты на продвижение по служебной лестнице. Часто в такой ситуации возникают трудности с адаптацией у тех сотрудников, которым удалось «вырваться вперед». Это надо учитывать, поскольку может сказываться на деятельности будущего руководителя. Проблема заключается в том, что не каждому человеку, ранее бывшему исполнителем, легко в психологическом плане перейти на руководящую должность. Кроме того, изменяется статус по отношению к сослуживцам, а, следовательно, должен измениться и способ взаимодействия. У бывших коллег появляется больше ограничений во взаимоотношениях с вновь назначенным начальником, что не всегда до конца осознается ими, и тем самым провоцируются противоборство и недовольство со стороны руководителя, и, как следствие, возмущение и агрессия со стороны бывших коллег. Необходимо отметить, что внутренний отбор имеет немало преимуществ перед внешним. Во-первых, это обходится дешевле и требует меньших усилий. Во-вторых, пример внутреннего продвижения по служебной лестнице является стимулом для всего коллектива, и тем самым можно рассчитывать на большую отдачу от работников. В-третьих, что наиболее важно, такие работники уже «вписались» в организацию, освоили основные принципы ее работы, приспособились к особенностям взаимодействия, требованиям корпоративной культуры и т.п. Ориентация в работе и адаптация в этом случае будут требовать гораздо меньших затрат и времени. Однако при использовании внутреннего источника кадров есть и издержки – отсутствие у работника «свежего взгляда». Значение этого факта зависит от целей, которые ставит перед собой организация и от того, насколько ее стратегия соответствует этим целям.

- ◎ **Внешний отбор** подразумевает привлечение к работе людей, не связанных прежде с предприятием. Рассмотрим преимущества и недостатки данного способа. Первый плюс – это, конечно же, большой выбор кандидатов. К тому же привлечение новых людей дает больше шансов для свежих идей, решений. Однако неизвестно, сумеет ли вновь принятый работник установить контакт с незнакомым коллективом, и как долго у него будет

длиться процесс адаптации. Каким образом можно привлечь кандидатов со стороны? Существует множество способов. Мы перечислим их в порядке убывания эффективности:

- Вновь обратиться к тем, кто уже был кандидатом и «зачислен в резерв».
- Открыто или негласно оповестить работников организации о свободных местах.
- Дать объявления в специализированные СМИ.
- Установить контакты с образовательными учреждениями.
- Обратиться в агентство по подбору персонала.

Описание будущей работы

Отмечаются две тенденции в способе подачи информации о свободных вакансиях: идеализированный и реалистичный.

- ⊙ **Идеализированное** описание условий работы выглядит более привлекательным, чем реальность. В результате могут откликнуться довольно много претендентов на должность. Однако в будущем, когда ожидания работника не оправдаются, это неизбежно повлечет неудовлетворенность работой, недовольство, следствием чего может быть низкая работоспособность и отдача в работе, переход на другую должность, и, наконец, увольнение (в основном, по собственному желанию). При этом весь процесс поиска нужно будет начинать сначала.
- ⊙ **Реалистичное** описание работы не для всех будет привлекательным, и не всех будут устраивать условия. Однако этот вариант предполагает отклик только тех людей, чьим запросам соответствует данная работа. Поэтому в этом случае работа будет в основном оправдывать ожидания претендента, что повлечет за собой большую степень удовлетворенности и стабильности.

Практика показывает, что реалистичное описание условий работы экономически выгоднее.

Критерии отбора кадров

Как уже было сказано выше, для успешного поиска кандидата необходимо заранее сформулировать требования к будущему сотруднику.

Требования указывают на то, какими качествами должен обладать человек для того, чтобы эффективно выполнять свою работу. На

основании этих критериев будет приниматься решение о соответствии кандидата и о преимуществах того или иного кандидата перед другими. Такими критериями могут выступить предыдущий опыт работы претендента, его теоретические и практические знания, интеллектуальные способности, личностные особенности, мотивация, деловые качества, физические данные, состояние здоровья и др. Критериями могут выступить и ожидания кандидата. Например, его отношение к условиям труда – к возможным частым командировкам или ненормированному графику работы, внеплановым нагрузкам и т.п.

Отбор может быть затруднительным, если список предъявляемых требований слишком велик или наоборот слишком мал. Требования к кандидату должны быть реалистичными и гибкими, так как найти человека, абсолютно соответствующего предъявляемым характеристикам, достаточно трудно. Поэтому необходимо выработать подход к разработке самих критериев отбора.

Основные требования к ним должны быть следующие.

- **Соответствие.** Критерии отбора должны соответствовать тому роду занятия, для которого проводится отбор, т.е. если вам нужен, например, дизайнер графического интерфейса, то совсем не обязательным является требование знания китайского языка, если, конечно, интерфейс не предполагается делать на китайском языке.
- **Полнота.** Критерии должны быть полными, охватывающими все необходимые характеристики претендента на данную работу. Необходимо ориентироваться не только на уровень профессиональной подготовки кандидата, но и на другие качества, необходимые для успешной работы. Например, нанимая разработчика для участия в крупном и продолжительном проекте по разработке программного продукта, необходимо обратить внимание на его коммуникативные способности, ориентацию на успех. В противном случае возможен вариант, при котором уже в процессе работы сотрудник «не впишется» в коллектив или впадет в депрессию от первых неудач.
- **Надежность.** Критерии, на основании которых проводится отбор, должны обладать высоким уровнем достоверности, то есть правдивости. Для этого необходимо иметь возможность быстро и эффективно проверить те данные, которые получены в результате собеседования, анкетирования, резюме и иных мероприятий. Например, критерий «знание основных конструкций языка Java» надежно проверяется письменным тестированием за час, а

проверка критерия «моральная устойчивость в быту» затруднительна или невозможна.

Теперь немного о самих критериях отбора.

- **Образование.** Большинство работодателей предпочитает высшую степень образования низшей. Однако не забывайте, что такие данные необходимо соотносить с личными успехами и достижениями человека. Нужно уточнить вид образования, специализацию, квалификацию (возможно, она повышалась на различных курсах) и наличие дополнительного образования.
- **Опыт.** Кроме трудового стажа следует учесть характер предыдущей работы. Критерий оценки трудового стажа может быть различным: какое общее время деятельности в одной организации, сколько времени пребывал кандидат в данной организации на конкретной должности, как долго работал в конкретной должности в определенном структурном подразделении и т.п.
- **Физические данные.** Требование к физическим данным имеет очевидную подоплеку, тем более для ряда некоторых профессий. Отбор по внешним данным не представит для вас особого труда, поэтому целесообразно изучить медицинские заключения (если таковые имеются) на предмет противопоказаний к выполнению тех или иных работ.
- **Акцентуация личности.** Акцентуация личности создает так называемую «яркую индивидуальность», которой свойственны характерные черты. Например, гипертимы безумно активны, оптимистичны, иногда таких просто невозможно остановить! Педанты до бесконечности наводят порядок. Неплохо включить в список психологических методик, используемых при отборе персонала, методику определений акцентуаций характера К. Леонгарда. Акцентуация характера или темперамента во многом предопределяет поведение личности, поэтому полученные результаты нужно учитывать не только при отборе кандидатов на должность, но и помнить о них в процессе работы и общении вновь принятого сотрудника с коллективом.

Собеседование – общепринятая процедура при приеме на работу, в случае, если речь идет о разработке программного обеспечения, эта процедура применяется почти в 100% случаев. Прежде чем назначать и проводить собеседование, необходимо собрать достаточное количество информации о желающих поступить на работу – их анкеты, резюме, устные отзывы и т.п.

Собеседование включает в себя три этапа:

1. установление контакта с кандидатом;
2. интервью (получение необходимой информации и ее проверка), возможно профессиональное или психологическое тестирование;
3. завершение собеседования (подведение итогов и, может быть, договоренность о следующей встрече).

Основные задачи собеседования:

- определить уровень способностей кандидата, соответствие способностей требованиям должности;
- определить личные особенности кандидата.

Личные качества соискателя будут играть важную роль:

- в процессе его адаптации к новым условиям, корпоративной культуре организации, которая имеет свои правила, нормы, ценностные системы;
- в процессе вхождения в коллектив;
- в совместимости с руководителями, коллегами и т.п.

Психологическое тестирование

В зависимости от того, на поиск какого работника вы ориентируетесь, те или иные личностные особенности будут более или менее важны. Если вам нужен, например, усидчивый работник, который бы кропотливо и точно выполнял монотонную работу (например, тестирование), следует, в первую очередь, обратить внимание на такие характеристики, как:

- высокая ответственность;
- пунктуальность;
- склонность к постоянному виду деятельности;
- отсутствие импульсивности.

Если же вы ищете человека, в обязанности которого будут входить переговоры с заказчиком, то от кандидата потребуются такие качества, как:

- умение расположить к себе собеседника, наладить контакт, поддержать беседу;
- тактичность;
- дипломатичность;
- развитое аналитическое (и творческое) мышление;
- уравновешенность и т.п.

Личностные качества определяются, как правило, с помощью психологических тестов, описания которых можно найти в специальной литературы. Для применения таких тестов нужны методики и квалифицированные специалисты, однако, общую направленность и возможности психологического тестирования полезно представлять любому менеджеру.

- ***Методика диагностики стрессоустойчивости и социальной адаптации (Холмс и Раге)***. Она основывается на том, что психическим и физическим заболеваниям обычно предшествуют серьезные изменения в жизни. В зависимости от прошлых условий и текущего состояния, человек обладает определенной степенью стрессоустойчивости. Для диагностики социально-психологической адаптации можно использовать методику К. Роджерса и Р. Даймонда.
- ***Методика предрасположенности личности к конфликтному поведению (К. Томас)***. Эта методика позволяет выявить тип поведения личности в конфликтной ситуации (соревнование, сотрудничество, компромисс, избегание, приспособление).
- ***Проективные методики***, такие, как ТАТ (Тематический Ассоциативный Тест), тест Роршаха, тест «Несуществующее животное», тест «Дерево, дом, человек» позволяют работать с неосознанными тенденциями личности. Обработка результатов таких тестов требует детального внимательного анализа, комплексного подхода. Кроме того, наряду с данными методиками следует использовать и объективные результаты.
- ***Методика диагностики типа поведенческой активности (Л. И. Вассерман и Н.В. Гуменюк)***. Эта методика дает возможность установить уровень активности личности, целеустремленности, энергичности, склонности к доминированию.

- **Методика диагностики направленности личности (Б. Босс).** С ее помощью можно определить ориентацию личности в работе: «на дело», «на себя», «на общение».
- **Методики диагностики личности на мотивацию к успеху и на мотивацию к избеганию неудач (Т. Элерс) и методика диагностики степени готовности к риску (Шуберта).** Установлено, что люди с высоким уровнем защиты, т.е. страхом перед несчастными случаями, чаще попадают в неприятности, чем те, которые имеют высокую мотивацию на успех.

Стоит отметить необходимость комплексного анализа результатов всех психологических методик. Ни в коем случае не стоит делать выводов на основании одного или двух тестов, иначе вы сделаете непоправимую ошибку – упустите потенциального работника или примете не того человека, которого искали.

Интервью

Интервью – это неформальная беседа с кандидатом. Как правило, интервью должен проводить менеджер, несущий ответственность за результаты приема кандидата. Проанализируем следующие моменты интервьюирования. На что нужно обратить внимание при ответе кандидата на тот или иной вопрос и на каких вопросах при оценке деловых и личных особенностей особенно важно поставить акцент?

Ниже приведены некоторые рекомендации для менеджеров по формулировке вопросов и оценке ответов.

1. **«Расскажите немного о себе».** Во-первых, оцените насколько четко, ясно излагает собеседник свои мысли, как он следует теме вопроса, уверенно ли при этом держится, какие использует интонации. Во-вторых, выясните заинтересованность в выгодном представлении своей биографии – это будет видно из контекста повествования. Конечно, целью претендента является получение рабочего места и, соответственно, создание у вас собственного положительного образа. Поэтому посмотрите, насколько человек уверен в своих способностях, как он себя позиционирует – «рисует» ли он перед вами, «надувает щеки», подчеркивает ли свое превосходство перед другими или же говорит об опыте, квалификации, порядочности.
2. **«Какой Вам представляется жизнь? Есть ли в ней трудности?».** Если кандидат говорит о множестве трудностей, встречающихся на жизненном пути, сетует, что все изменения – неприятные и не к

лучшему, говорит о том, что одним везет «по жизни», а других судьба не балует и т.д., то перед вами, скорее пессимист, неуверенный в себе пассивный человек, в большей степени надеющийся на судьбу, стечение обстоятельств, чем на себя. Конечно, в качестве работника вам будет интересен, прежде всего, человек, позитивно и оптимистично настроенный, активный, рассчитывающий на себя, уверенный в своих силах, в том, что трудности на его пути всегда можно преодолеть и для этого необходимо приложить некоторые усилия.

3. **«Почему Вы считаете себя достойным занять данную должность?»**. Этот вопрос дает возможность кандидату без стеснения назвать свои основные профессиональные и личные достоинства. Здесь важно оценить умение человека убеждать, представлять себя.
4. **«Ваши сильные стороны?»**. Кандидат должен отметить в первую очередь те качества, которые важны для предполагаемой должности. Если вы услышите перечисление банального списка, попросите уточнить собеседника, чем именно эти качества отличают его от других.
5. **«Как Вы представляете свое положение в компании через пять лет? И почему?»**. Инициативный человек будет ориентирован на будущий успех, профессиональный рост. Отсутствие у человека далеко идущих планов, перспектив практически всегда является признаком его пассивности и безынициативности.

Другие моменты, на которые следует обратить внимание – это отношение к работе, заинтересованность в работе, уровень ответственности, целеустремленность, способность к самоорганизации, самооценка, независимость, тенденция к лидерству, характер и направленность отношений с людьми, корпоративность, мотивация к работе. Информацию об этих характеристиках смогут дать следующие вопросы:

- **Заинтересованность в работе, степень ответственности:**
 - ? Вы готовы к сверхурочной работе?
 - ? Как Вы проявляли свою инициативу на предыдущем месте работы?
 - ? Как Вы поступите, если работа покажется Вам не по силам?
 - ? Как Вы поступите, когда Ваша работа будет требовать от Вас дополнительных усилий и сверхурочного времени?
- **Целеустремленность:**

- ? Ставили ли Вы перед собой почти нереальные цели? Достигали ли Вы их и каким образом?
- ? Если Вы сталкиваетесь с трудностями, что Вы делаете?
- ? Вы принципиальный человек?
- ? Могут ли про Вас сказать другие, что Вы – человек слова?

- ***Степень самоорганизации:***
 - ? Планируете ли Вы свой день заранее?
 - ? Строите ли Вы планы на ближайшие полгода, год?
 - ? Планируете ли Вы свои расходы?
 - ? Много ли Вы тратите времени на то, чтобы привести свои вещи в порядок?

- ***Самооценка:***
 - ? Вы считаете себя способным человеком?
 - ? Вы можете принимать ответственные решения в сложных ситуациях?
 - ? Многого ли Вы добились в жизни?
 - ? Если у Вас будет выбор, приметесь ли Вы за дело, которое заведомо будет нелегким для Вас?
 - ? Если бы у Вас была возможность прожить жизнь заново, Вы бы изменили в ней что-то? Если да, то что?

- ***Независимость:***
 - ? Вы склонны отстаивать свое мнение?
 - ? Нравятся ли Вам люди, имеющие свое мнение, которое может быть идет вразрез с мнением остальных?
 - ? Когда Вам нужно принять важное решение, Вы предпочитаете сами решить или советуетесь с кем-то?
 - ? Если Вы считаете, что Ваш начальник не прав, Вы скажете ему о его заблуждении или предпочтете промолчать?

- ***Тенденция к лидерству:***
 - ? Вы предпочли бы руководить или подчиняться?
 - ? Как Вы считаете, смогли бы Вы управлять коллективом?
 - ? Вы хороший организатор?
 - ? Вам нравится быть в центре внимания?
 - ? Хотели ли Вы когда-нибудь открыть свое дело? Какое? Что Вам мешает?

- ***Направленность отношений:***
 - ? Вы легко сходитесь, находите общий язык с людьми?
 - ? Вы можете прочувствовать ситуацию другого человека так, будто Вы переживаете это?
 - ? Вы часто критикуете других людей?

? Вы считаете, что нужно помогать людям, или многие просто пользуются этим и могут справиться с трудностями сами?

- **Корпоративность:**

? Для Вас важна принадлежность к группе?

? Характерна ли была конфликтная обстановка на Вашей бывшей работе? Часто ли Вы конфликтовали с сотрудниками? С руководством?

? Ценны ли для Вас негласные правила, обычаи в коллективе? Придерживаетесь ли Вы их?

? Для Вас удобнее работать в коллективе или в одиночку?

- **Мотивация к работе:**

? Почему Вы желаете получить это место работы?

? На какую заработную плату Вы рассчитываете?

? Что для Вас в работе главное?

? Что значит для Вас хорошая работа?

? Что значит для Вас престижная работа?

Существует также ряд «затруднительных» вопросов, которые можно задать кандидату и на основании полученных ответов сделать соответствующие выводы:

? За что Вас критиковали на последнем месте работы?

? Что бы Вы хотели изменить в себе, как в человеке и как в профессионале?

? Какую именно работу Вы выполняете с наибольшим удовольствием? Какую с наименьшим?

? Опишите себя несколькими словами. Какими словами Вас описали бы Ваши коллеги? Близкие знакомые?

? Вы иногда искажаете правду? В каких случаях Вы это делаете?

? Опишите идеального начальника и его противоположный портрет.

? При каких обстоятельствах Вам трудно добиться успеха? Приведите пример.

? Что бы Вы изменили в работе для того, чтобы она стала более интересной?

? С чем Вы не хотели бы столкнуться на новом месте работы?

? У всех есть свои недостатки. Назовите, какие черты Вы хотели бы исправить в себе.

Профессиональное тестирование

Профессиональное тестирование обычно включается в интервью, поскольку проводить его должны профессионалы, имеющие опыт

работы в этой области. В то же время профессиональное тестирование (выявление профессиональной компетенции) представляет собой отдельную проблему кадрового менеджмента.

Существуют области деятельности, в которых методы профессионального тестирования применяются давно и надежно отработаны. Яркий пример – проверка водительских навыков при выдаче прав на управление транспортным средством. В тестировании такого рода создается ситуация, в которой испытуемый помещается в реальные условия работы. В то же время ситуация является учебной и безопасность участников гарантируется. Заметим, что подобные тесты продолжаются достаточно короткое время и стоят не очень дорого.

При тестировании профессиональных навыков программистов затруднительно создать ситуацию, которая является реальной, безопасной и недорогой одновременно.

Часто используемый для этой цели **тест «напишите программу для данной задачи»** (обычно весьма искусственного характера) не информативен. Умение написать за час программу в 100 строк для учебной задачи никоим образом не гарантирует умения написать за 100 часов программу в 10 000 строк для решения реальной задачи.

Тест «сделайте вот этот кусок нашей общей работы» не разрешит проводить руководство компании. Во-первых, «этот кусок» является know-how и секретной собственностью компании, во-вторых, проект уже задерживается, и кто будет отвечать, если тест не будет пройден, в-третьих, это уже не тест, а работа в режиме испытательного срока, за которую нужно платить зарплату.

Наконец, часто используемые **тесты, состоящие в решении математических задач или построения алгоритмов**, скорее проверяют наличие общей математической или алгоритмической культуры, чем наличие специальных профессиональных навыков.

Проблема тестирования программистов (и не только программистов, а представителей любых профессий, в которых существенным является количественный аспект работы) состоит в том, что невозможно проверить умение справляться с большими задачами, предлагая решать маленькие задачи.

В этом случае можно предложить один вид тестов, который уместно назвать **«найди ошибку»**. Тест проводится следующим образом. Нужно взять какой-нибудь фрагмент реального рабочего материала (текста программы, архитектурной модели, формулировки требований), причем достаточно небольшого фрагмента, 2-3 страницы, и по возможности автономного. Исходный материал следует выбирать самого лучшего качества (т.е. нужно брать текст отлаженной программы). В этот материал искусственно вносится некоторое количество «незаметных» ошибок (опечаток, пропусков, неадекватных

изменений). Практически достаточно 3-5 ошибок на программу в 100 строк. После этого испытуемому предлагается за час выполнить любое действие, предполагающее внимательное изучение представленного материала. Например, если речь идет о тексте программы, можно предложить вставить отсутствующие комментарии. Результат теста – количество ошибок, которые испытуемый заметит в тестовом материале.

Предлагаемый тип тестов основан на хорошо известном и статистически проверенном утверждении, что качество программиста проявляется не в том, насколько быстро и много он может программировать, а в том, насколько мало он при этом делает ошибок.

7.3.4. Увольнение работников

Приведем несколько важных тезисов, касающихся увольнения персонала.

1. Менеджер должен быть готов к увольнению любого сотрудника в любой момент.
2. Если сотрудника нужно уволить, то чем раньше это сделать, тем меньше будут издержки. При увольнении неградивого сотрудника нагрузка на остальных не возрастает, а уменьшается.
3. Увольнение – стрессовая ситуация для сотрудника, для менеджера и для компании, даже если речь идет просто об уходе сотрудника по собственному желанию, не сопровождающемся никаким конфликтом. При уходе любого сотрудника (если, конечно, речь не идет об откровенных бездельниках) возникает проблема замены.
4. Увольнение должно проводиться в соответствии с действующим трудовым законодательством. Так как увольнение (если оно только не происходит по собственному желанию работника) является конфликтной ситуацией, следует учитывать возможность обращения уволенного сотрудника в суд с жалобой на незаконное увольнение.
5. Уволить даже самого неградивого сотрудника психологически сложно. В особенности велики психологические препятствия при значительном сокращении персонала.

Стоит обратить внимание на специальный, активно развивающийся в последнее время вид консалтинговых услуг для бизнеса по работе с сокращенным персоналом – аутплейсмент.

Аутплейсмент (англ. Outplacement) — работа с сокращенным персоналом.

Аутплейсмент, как вид консалтинговых услуг, оказывают кадровые и рекрутинговые агентства. Он включает сопровождение, ориентацию на рынке труда, психологическую поддержку и, в конечном итоге, трудоустройство сокращенного в результате реструктуризации или каких-либо других организационных изменений работника. Чаще всего консалтинговые агентства, занимающиеся аутплейсментом, предлагают уволенным специалистам специальные обучающие программы, направленные на развитие навыков правильного поиска работы. Реже агентства сами ищут новую работу для сокращенных сотрудников своих компаний-клиентов. Аутплейсмент персонала, правда, как правило, могут себе позволить только очень крупные и успешные компании.

7.4. Персонал проекта по разработке программного обеспечения

7.4.1. Управление персоналом проекта

Главным ресурсом, необходимым для производства программного обеспечения, являются люди. В первую очередь, конечно, важны технические навыки участников проекта. Однако эти навыки необходимо применять в нужное время и в нужном месте, что предполагает комбинацию работы в команде и лидерства. В соответствии с «Законом Брукса», сформулированным Бруксом (Brooks) в его знаменитой книге «Мифический человеко-месяц», привлечение большего числа людей в погибающий программный проект не помогает, а может даже и навредить. Однако при грамотном управлении персоналом такие «дополнительные человеко-месяцы», появляющиеся в ходе проекта, могут оказаться очень полезными.

	Фредерик Брукс - профессор вычислительной техники в школе бизнеса Кенан университета штата Северная Каролина в Чэпел Хилл. Он известен, прежде всего, как "отец IBM System/360". Помимо этого, Брукс занимался разработкой в IBM архитектуры компьютеров Stretch и Harvest. В 1985 году Фредерик Брукс, Боб Эванс и Эрик Блох были награждены Национальной медалью в области технологии за проектирование разработки операционной
--	---



системы Operating System/360. Доктор Брукс был членом национального и военного комитетов по науке, основал в Чэпел Хилл факультет вычислительной техники и возглавлял его с 1964 по 1984 годы. В настоящее время он занимается преподаванием и исследованиями в области архитектуры компьютеров, молекулярной графики и виртуальных сред.

Обычно менеджер старается довести работу до конца так, чтобы при этом у подчиненных не возникало чувство неудовлетворенности. Менеджер сам в этом заинтересован, так как недовольные сотрудники работают непродуктивно. Одной из самых трудных задач менеджера является выполнение работы при условии получения коллективом удовлетворения от проделанной работы. Например, руководство компании может поставить условие, что работа должна быть выполнена на старом языке программирования для совместимости версий, хотя разработчикам было бы интереснее опробовать новый язык программирования. Менеджер должен сделать так, чтобы техническое творчество исполнителей имело нужное направление. Диктатура в управлении может вызвать негодование коллектива и привести к потере мотивации, а излишнее попустительство может привести к потере времени и напрасно проделанной работе. Решением проблемы является лидерство, выяснение истинных желаний и потребностей людей, активное их распределение и объединение в попытке достижения успеха. Лидеры проекта должны варьировать степень своей ответственности в управлении, основываясь на величине проекта. В больших проектах их обязанности состоят в основном в управлении. В малых же проектах лидеры могут обеспечивать как общее управление, так и личное участие непосредственно в разработке.

Разработчики хотят иметь интересную работу, хотят иметь возможность проявить себя, хотят, чтобы их заметили и наградили, и хотят теплых дружеских отношений в коллективе. Предпосылкой этих желаний является здоровое самоуважение, главным источником которого является качественная работа. Следовательно, необходимо, чтобы разработчики точно знали, как оценить трудозатраты, необходимые для производства хорошего продукта, как доказать себе и другим, что работа сделана корректно, и как измерить качество сделанной работы.

7.4.2. Развитие команды проекта

Оптимальный размер команды

Как правило, количество сотрудников, с которыми приходится регулярно общаться каждому из команды исполнителей проекта, составляет от трех до восьми человек. В принципе программист-разработчик может работать почти без регулярного индивидуального общения с кем бы то ни было. Однако такая изоляция обычно приводит к непониманию разработчиком предъявляемых к нему требований и, как следствие, к относительно низкому уровню эффективности. С другой стороны, если участник проекта работает в слишком большой команде, то из-за постоянного общения с каждым из коллег у него не останется времени на выполнение своих основных обязанностей по проекту. Это снова приводит к относительно низкому уровню эффективности. В частности, *регулярное общение* означает разговор с кем-либо в течение примерно двух часов в неделю. Таким образом, если сотрудник регулярно общается с десятью своими коллегами, то половина его рабочего времени (20 из 41 часа в неделю) проходит в разговорах. Менеджеры проектов, планирующие проекты с большими командами, должны это учитывать.

Подчиненность участников проекта

В проектно-ориентированных организациях персонал проекта организационно отчитывается перед руководителем проекта. Менеджер является начальником персонала. Разработчик всегда прикреплен к какому-либо конкретному проекту и организационно не связан с другими разработчиками в других проектах внутри компании. Преимуществом является упрощение вертикали власти, однако основной недостаток заключается в профессиональной изоляции разработчиков. Например, очень маленькие компании почти всегда организованы по принципу проектной ориентации, однако даже огромные компании иногда организуются так же, особенно когда хотят произвести впечатление на заказчика.

Вообще говоря, системы отчетности на практике имеют более сложную и комплексную структуру, чем в проектно-ориентированных организациях. Это происходит из-за использования *матричной структуры организации*. Такая структура предусматривает, что служащие относятся к функциональным подразделениям (например, отдел разработки, отдел продаж и т. п.) и выделяются этими подразделениями для участия в том или ином проекте. Таким образом, начальник программиста, отвечающий за своего подчиненного, является членом соответствующего функционального подразделения (например, отдел разработки). Однако в каждом проекте, в котором разработчик

принимает участие, он подчиняется менеджеру проекта. Обычно разработчиков на постоянной основе привлекают не более чем к двум проектам. Основным достоинством матричных организаций является профессионализм и возможность улучшения навыков. К недостаткам относится нечеткость в линиях субординации. Довольно часто компании стараются найти золотую середину между матричной и проектно-ориентированной структурами.

Развитие команды и развитие персонала

Каждый проект, кроме достижения основной своей цели, должен использоваться менеджером для развития команды. А именно, в процессе выполнения проекта:

- происходит приобретение участниками новых технических знаний, новых технологий, т.е. техническое обучение и совершенствование профессионального мастерства,
- вырабатываются и укрепляются навыки командной работы,
- приобретается опыт выполнения проектов в соответствии с принятой в компании схемой организации процесса разработки программного обеспечения.

Косвенным результатом любого проекта, поэтому является развитие персонала, что приводит к более успешному выполнению последующих проектов.

7.5. Заключение

Успешная деятельность компании невозможна без четко сформулированной и последовательно проводимой кадровой политики, под которой понимается совокупность методов отбора, стимулирования и развития персонала, а также формирования корпоративной культуры и командного духа. Управление персоналом в компании по разработке программного обеспечения в основном похоже на управление персоналом в организациях проектного типа, однако имеет и несколько специальных аспектов, в особенности в вопросах подбора кадров.

Смысловые связи между рассмотренными понятиями менеджмента удобно представить в виде карты памяти

7.6. «Карта памяти» по теме



7.7. Список использованной и дополнительной литературы

1. Неларин Корнелиус. HR менеджмент. Поиск, подбор, тренинг, адаптация, мотивация, дисциплина, этика, Издательство: Баланс Бизнес Букс, 2005.
2. Н. Г. Агеева, О. Н. Дмитриев, Э. С. Минаев. Менеджмент для инженера. Часть 1. Основы менеджмента. Серия: Бизнес для инженера, Издательства: Высшая школа, Доброе слово, 2002 г.
3. Гусарова Н.Ф. Общая психология / Уч. пособие. СПб: СПбГИТМО (ТУ), 2001.
4. Гусарова Н.Ф. Координация в технологических процессах со слабо формализуемыми критериями. Монография. СПб: СПбГИТМО(ТУ), 2001.
5. Баранов С. Н. Управление программным проектом. Лекции по спецкурсу "Технология программирования". - СПб: Санкт-Петербургский государственный электротехнический университет, рукопись, 1998.
6. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. - СПб.: Символ-Плюс, 1999.

Тема 8. Управление временем проекта

8.1. Введение

Управление временем проекта – это, прежде всего, определение временных рамок каждой из задач проекта, или, иными словами, создание, оптимизация и отслеживание календарных планов.

Изучив учебный материал данной темы, Вы:

- узнаете или пополните свои знания о видах календарных планов и об их возможных представлениях;
- научитесь читать, составлять и анализировать календарные планы в форме диаграмм Ганта и сетевых диаграмм;
- познакомитесь с методом анализа критического пути при планировании проекта.

В рамках темы рассматриваются следующие учебные вопросы:

- Календарный план проекта.
- Сетевое представление проекта.

8.2. Календарный план проекта

Заказчика, а значит, и менеджера проекта, в первую очередь интересуют сроки выполнения проекта. Для определения длительности проекта в целом и каждой из его фаз служит **календарный план (или план-график) проекта**, лежащий по сути дела в основе любого планирования. В самом простейшем варианте календарный план – это просто список всех работ проекта с указанием дат их начала и окончания. Менеджер проекта в любой момент, глядя на календарный план, в состоянии понять, насколько проект отстает от плана или опережает план.

Иногда укрупненный календарный план согласуется с заказчиком и включается в качестве приложения в договор (так делают, например, если оплата по договору предусмотрена в несколько этапов).

Из сказанного очевидно следует, что создание календарного плана проходит в несколько шагов:

- ✓ сначала формируется структура декомпозиции работ (СДР) проекта, т.е. представление проекта в виде иерархического дерева этапов и задач (работ),
- ✓ затем для каждой задачи проекта выставляются сроки в соответствии с возможностями команды разработчиков.

Структура декомпозиции работ (СДР) (work breakdown structure (WBS)) – это иерархическая структура, используемая для организации задач в отчетах по календарному плану и при отслеживании затрат.

Этап (фаза, phase) – это группа связанных задач, завершение которых означает выполнение важной части проекта.

Задача (работа, task) – действия, имеющие начало и конец. Планы проектов состоят из задач.

Далее мы более подробно опишем алгоритм создания календарного плана.

8.2.1. Создание списка задач и структуры декомпозиции работ

Недооценка важности структуры декомпозиции работ (СДР) – одна из наиболее серьезных ошибок, допускаемых при планировании проекта. СДР представляет собой иерархический список этапов и задач проекта. Это основа календарного плана проекта. Следует уделить должное внимание составлению СДР – в дальнейшем это сэкономит время, которое пришлось бы потратить на доработку и исправление ошибок.

Построение СДР *начинается с* составления списка основных компонентов проекта (суммарных задач) и их последующего разбиения на менее крупные компоненты (задачи). Следует продолжать структурировать каждый компонент, пока не будет достигнут уровень детализации, адекватный требованиям плана. Какой уровень детализации считается достаточным? Это зависит от следующих факторов:

! Задачи какого уровня предстоит отслеживать и включать в отчеты о состоянии?

Члены группы проекта должны будут предоставлять руководителю проекта сведения о состоянии задач, а тот должен будет составлять отчеты о состоянии этих задач. Если весь объем работ разбит на однодневные задачи, а проект рассчитан на несколько месяцев, то руководителю и рабочей группе придется потратить слишком много времени на ввод и отслеживание состояния.

! Какой уровень детализации гарантирует заблаговременное предупреждение о возможных рисках?

Задачи должны быть структурированы так, чтобы это позволяло выявлять проблемы с достаточным упреждением, позволяющим предпринять необходимые меры.

! Можно ли определить эффективные зависимости между задачами?

Если выполнение какой-либо работы не может начаться до завершения другой работы, необходимо оформить каждую из этих единиц работы в виде отдельной задачи, что позволило бы установить нужную зависимость.

Приведем пример законченной СДР с показанными номерами ее уровней.

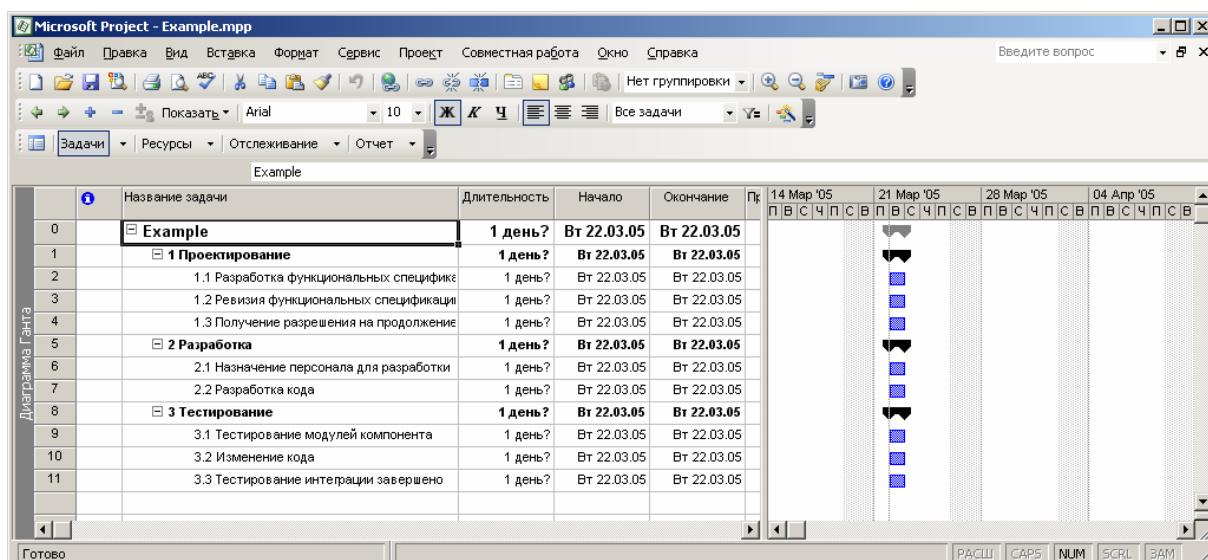


Рис. 1. СДР (WBS) проекта.

8.2.2. Оценка длительности задач и трудозатрат

Для определения времени, которое предполагается затратить на выполнение задач, вводятся значения трудозатрат или длительности.

Трудозатраты — это объем работ или число человеко-часов, необходимое для завершения задачи.

Длительность — это фактическое время, которое пройдет до завершения задачи.

Так, если задача требует 16 часов работы, выполняемой одним человеком, ее длительность составит два дня (при 8-часовом рабочем дне). Если работу будут выполнять два человека, длительность уменьшится до одного дня. Однако объем работ останется прежним. Если используется оценка на основе трудозатрат, длительность каждой задачи определяется назначениями ресурсов. Такой подход называется *планированием с фиксированным объемом работ*.

Можно использовать любой из двух методов планирования – на основе трудозатрат или на основе длительности, однако, необходимо сделать выбор заранее и строго придерживаться его при составлении плана.

Задавая трудозатраты или длительность в плане, необходимо следить за тем, чтобы они вводились на уровне задач самого нижнего уровня, а для суммарных задач (этапов) вычислялись автоматически (если для планирования используется специальное программное обеспечение, например, Microsoft Project или Primavera Project Management, то именно так и произойдет).

Ниже показана СДР с указанными длительностями каждой работы.

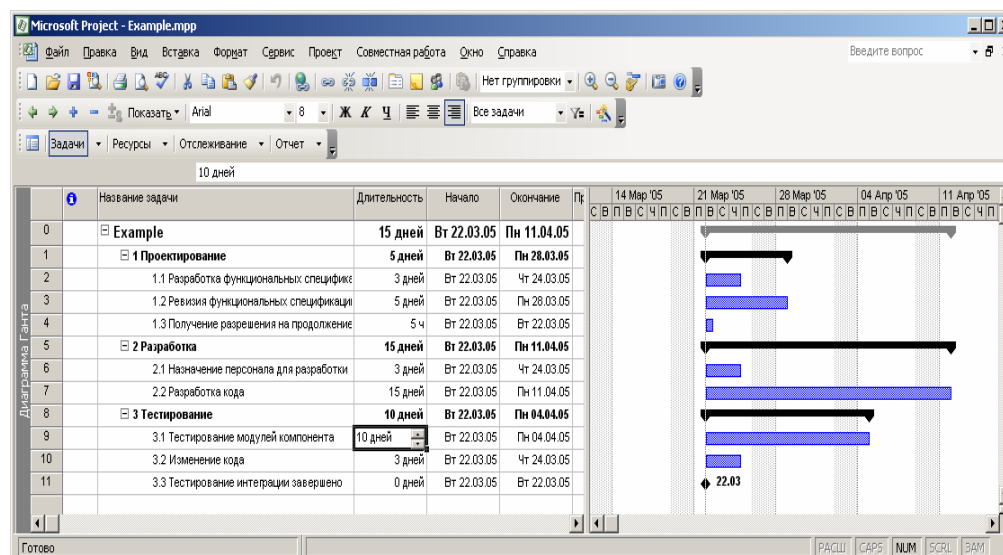


Рис. 2. СДР (WBS) проекта с указанием длительности каждой работы.

8.2.3. Создание зависимостей между задачами

Одним из наиболее важных этапов планирования является создание зависимостей или связей между задачами. Именно наличие этих данных отличает план как эффективное средство управления от плана, имеющего всего лишь презентационное предназначение.

Зависимость имеет место, когда начало или окончание одной задачи зависит от начала или окончания другой задачи. Большинство задач зависят от других задач. После установления зависимостей можно легко выяснить, от каких факторов зависит дата окончания проекта. Внеся изменения в одну задачу, можно будет сразу увидеть, как это отразится на остальной части плана.

Связывание (*Linking*) – установление зависимости между задачами в проекте. Для связывания задач пользователь определяет зависимость между их датами окончания или начала.

Задача составителя плана – добиться того, чтобы все задачи были включены в цепочку зависимостей. Для этого существует простое правило: *желательно, чтобы у каждой задачи был предшественник, если только она не начинается с даты начала проекта, а также задача-последователь, если это не последняя задача и не веха проекта* (о вехах см. ниже). Исключение составляют периодически выполняемые задачи (например, периодически выполняемые регламентные работы, архивация базы данных, периодическая аттестация и т.п.)

Предшественник (*Predecessor*) – задача, которую необходимо начать или завершить до начала или окончания другой задачи.

Последователь (*Successor*) – задача, которая не может быть начата или закончена до начала или окончания другой задачи.

При связывании задач можно задавать *зависимости различных типов*:

- ОН (окончание-начало),
- НН (начало-начало),
- ОО (окончание-окончание)
- НО (начало-окончание).

Наиболее употребительной является зависимость «Окончание-начало (ОН)»; она означает, что начало задачи-последователя «привязано» к окончанию задачи-предшественника. В простейшем случае это значит, что задача-предшественник должна закончиться, прежде чем сможет начаться задача-последователь. В более сложном случае, если задачи должны перекрываться, или если они зависят друг от друга, но между ними должен быть перерыв, то можно указать

запаздывание одной задачи относительно другой. Например, иногда задачи, связанные зависимостью «Окончание-начало (ОН)», выполняются параллельно, причем задача-последователь начинается не ранее, чем когда задача-предшественник будет выполнена на 50%. Так, например, можно организовать задачу тестирования модулей разрабатываемой программной системы (задача-последователь) и задачу написания программного кода модулей (задача-предшественник): ясно, что эти две задачи взаимозависимы, причем естественной зависимостью является именно «Окончание-начало (ОН)», но нет никакого смысла ждать до полной реализации всего кода, чтобы начинать тестирование.

Для некоторых задач можно задать и жесткие календарные ограничения (например, «начать не позднее, чем... » или «закончить не позднее, чем...»). Однако, как правило, делать это не рекомендуется, так как это существенно снижает гибкость планирования, создавая излишние препятствия для дальнейшей модификации планов.

8.2.4. Получение удобного представления календарного плана

Представление календарного плана в виде простого списка задач с указаниями их дат начала и окончания удобно разве лишь для очень коротких планов (именно в таком виде обычно и пишутся календарные планы, которые включаются по согласованию с заказчиком в текст договора, но в этом случае речь идет обычно не о детальных планах, а лишь о разбивке проекта на самые крупные этапы, «видимые» заказчиком).

Для достаточно подробных планов, которые менеджер использует в своей ежедневной работе, более удобны другие представления, как правило имеющие графический характер. Главным из таких представлений, получивших наибольшее распространение, является *ленточная диаграмма Гантта*.



Henry Laurence Gantt (1861-1919) – американский инженер, теоретик научной организации труда. Разработал ленточные диаграммы, впоследствии ставшие знаменитыми.

Диаграмма Гантта представляет собой прямоугольник: слева направо равномерно отсчитываются периоды времени (недели, месяцы), сверху вниз перечисляются работы, причём каждая работа представляется отрезком начало и конец которого размещаются в нужном периоде.

Ниже (см. рис. 3) показан календарный план проекта как в виде списка задач с указаниями их дат начала и окончания (а также длительностей), так и в виде диаграммы Гантта, причем представление календарного плана отформатировано с использованием MS Project так, чтобы названия задач критического пути и их отрезки на диаграмме Гантта выделялись красным цветом.

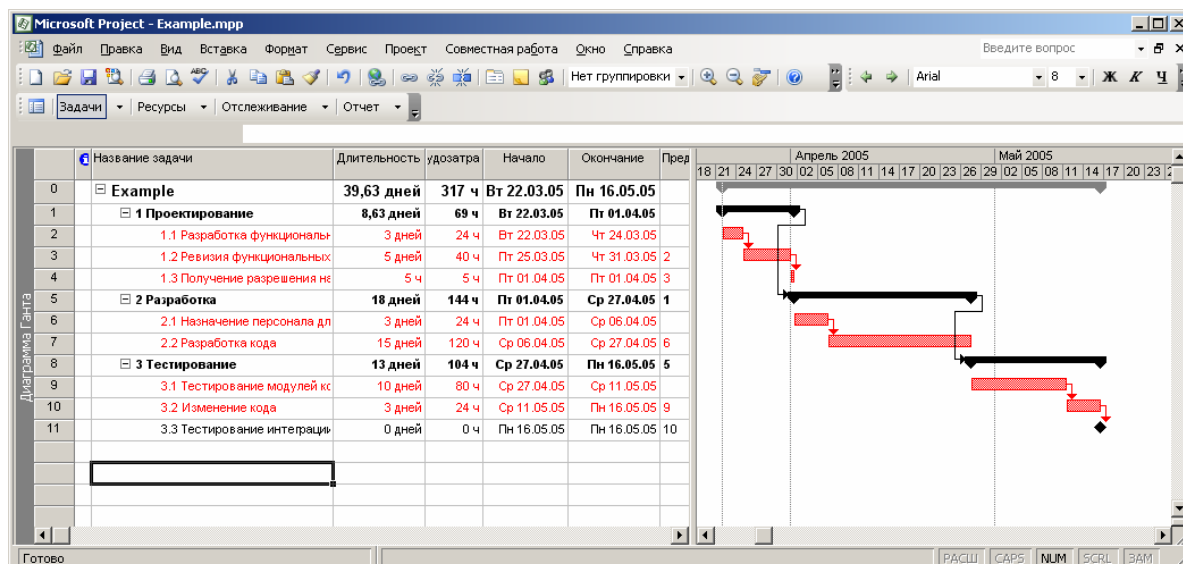


Рис. 3. Критический путь на диаграмме Гантта

Очень удобно при составлении календарных планов, имея в виду их дальнейшее представление в виде диаграмм Гантта, вводить в список задач (работ) проекта специальные *задачи-вехи*.

Bexa (Milestone) – опорная точка, отмечающая важное событие в проекте и используемая для контроля за ходом выполнения проекта.

Вехи являются фиктивными задачами, но они наглядно представляют основные опорные точки проекта (как правило, окончание важных этапов). Часто с вехой связано, например, совещание группы проекта или команды исполнителей. Имеет смысл делать веху последней задачей каждого этапа проекта, связывая ее зависимостью окончание-начало с предшественниками. При этом стоит придерживаться единообразного стиля именования вех. Так, вехи можно именовать, используя глаголы в совершенной форме, например, «тексты программ переданы в базу данных исходников», «все тесты пропущены», «группа оценки качества дала положительное заключение» и т.д., но не «выполняется прогон тестов».

Менее популярным для проектов производственного характера (а значит, и для проектов по разработке программного обеспечения) является *представление календарного плана проекта в виде календаря* (обычно с указанием задач по дням недели или по неделям). Такое представление все же иногда используется, в частности, для микропроектов непроизводственного характера, например, «встреча представителя потенциального инвестора» (с планированием по часам). С ним работают не только специализированные программы, предназначенные в помощь менеджеру проекта, но и такие достаточно общие инструменты, как, например, Microsoft Outlook.

8.3. Сетевое представление проекта

Для описания проектов с большим количеством сложных связей между задачами диаграмма Ганта оказывается недостаточно информативной. В этом случае для представления календарного плана используется *сетевое планирование*, в котором зависимости работ описываются *средствами теории графов*. План при этом отображается в виде сетевого графика, который представляет собой ориентированный граф с двумя выделенными вершинами – начало и конец работы, причем вершинами графа являются события, соответствующие пунктам плана, а рёбрами – работы. Рёбра нагружаются оценками длительности работ, например, в днях или неделях.

Сетевой график – очень удобный инструмент:

- во-первых, на нём чётко видна зависимость работ друг от друга,
- во-вторых, на основе сетевого графика можно вычислить длительность всей работы,
- в-третьих, пользуясь результатами этих расчётов, можно оптимизировать длительность и затраты на работу.

Длительность вычисляется следующим образом:

Первый вариант. Суммируем длительности работ по всем возможным путям в графе. Тот путь от начала к концу, который является самым длинным, объявляется критическим, потому что задержка любой работы, лежащей на этом пути, приводит к задержке всей работы в целом. Понятно, что критических путей (с одинаковой длительностью) может быть несколько.

Критический путь (Critical path) – набор запланированных задач, которые необходимо выполнить для окончания проекта в срок,

установленный календарным планом. Каждая задача на критическом пути является критической задачей.

Второй вариант. Для каждой работы вычислим, насколько можно отодвинуть ее окончание, чтобы общая длительность проекта не изменилась. Эта величина называется **полным резервом времени (*total slack*)** для данной работы, причем работа является критической, если полный резерв равен нулю (т.е. любая задержка этой работы ведет к задержке всего проекта). Критическим является путь, состоящий из критических работ.

Строго говоря, полный резерв времени данной задачи может быть и отрицательной величиной. Это происходит в том случае, когда у проекта установлена жесткая дата окончания, и для того чтобы в нее «вписаться», необходимо ускорить выполнение данной задачи. Стоит также отметить, что для менеджера проекта важным является и другой параметр задачи – ее **свободный резерв времени (*free slack*)**. Это та величина, на которую можно задержать окончание данной задачи, чтобы не произошло задержки задачи-последователя.

Рассмотрим пример. Здесь рассматривается классическая задача планирования: нормально работающая, полностью загруженная компания получила заказ, от которого по разным причинам невозможно отказаться. Перечислим названия событий, т.е. узлов в графе:

- 1 – начало работы;
- 2 – коллектив сформирован, рабочие места подготовлены;
- 3 – проектирование завершено;
- 4 – программирование завершено;
- 5 – комплексная отладка завершена;
- 6 – оборудование закуплено;
- 7 – группа технических писателей получила описание проекта и необходимые пояснения от проектировщиков;
- 8 – то же для ПО, разработка проектной документации завершена;
- 9 – группа технических писателей получила всю необходимую информацию об интерфейсах с пользователем, разработка программной документации завершена;
- 10 – группа оценки качества (Quality Assurance – QA) разработала тесты;
- 11 – группа QA оценила проект положительно;
- 12 – группа QA завершила автономное тестирование;
- 13 – группа QA завершила комплексное тестирование, получила всю документацию и действующий вариант системы;
- 14 – проверка качества завершена;
- 15 – конец работы (конечно, это не конец, будет ещё сопровождение, но на чём-то пример надо закончить).

Под каждым ребром графа записана планируемая длительность соответствующей работы (в неделях). Разумеется, каждая передача (3-7, 3-10 и т.д.) не может длиться более одного-двух дней, но ради простоты примера мы ограничились целыми числами.



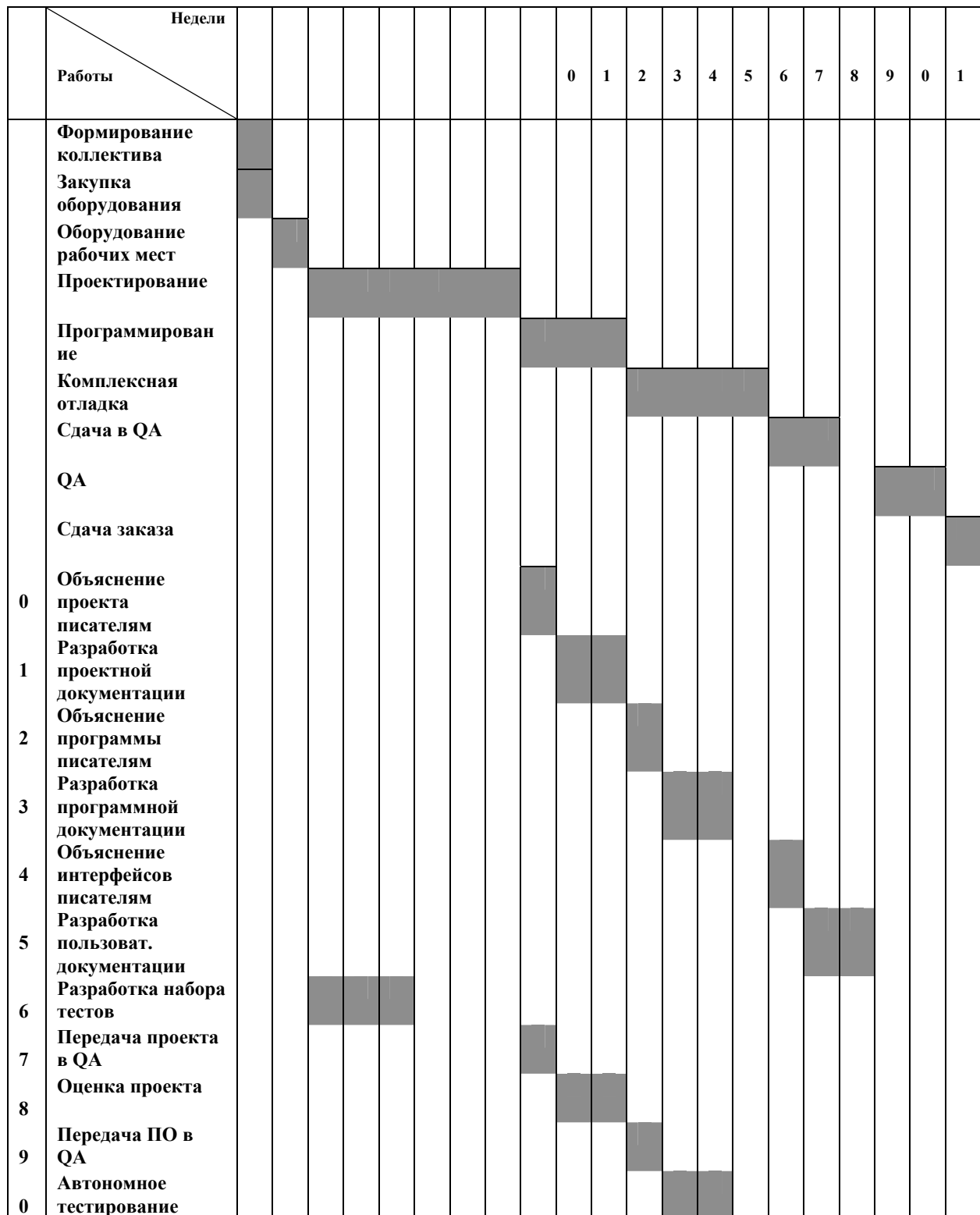
Критическими путями являются пути 1-6-2-3-4-5-9-13-14-15 и 1-6-2-3-4-5-12-13-14-15, т.е. вся работа не может быть выполнена быстрее, чем за 21 (двадцать одну) неделю. Понятно, что с точки зрения оптимальной загрузки коллектива, было бы лучше, чтобы все пути в графе от начала к концу имели бы примерно одинаковую длительность с тем, чтобы как-то уменьшить длину критического пути. Например, есть соблазн заставить группу QA проводить даже самое начальное тестирование, уменьшив нагрузку на программистов, работа которых находится на критическом пути. Но тогда очень трудно определить границы ответственности, программисты начинают выдавать откровенную халтуру и, в результате, сроки даже удлиняются. В реальных проектах, где работ очень много, всё-таки удаётся путём перераспределения работ улучшать сетевой график, по крайней мере, к этому стремятся все руководители.

Ещё несколько замечаний по данному примеру. Очевидно, неудачно спланированы работы между событиями 1, 2, 6. Коллектив сформирован за одну неделю, но рабочие места ещё не готовы. Группа технических писателей начинает работать на шесть недель позже

проектантов, а группа QA имеет трёхнедельный перерыв перед завершением проектирования и т.д.

Эти проблемы действительно трудны, каждая компания решает их по-своему, например, очевидным решением является использование одной и той же группы QA или технических писателей для нескольких групп разработчиков.

Для сравнения с сетевым графиком нарисую диаграмму Гантта для того же примера.



7. Пресняков В.Ф. Информационные технологии в управлении проектами. -М.: 1998 г.
8. Разумов И.М. Сетевые графики в планировании - М.: 1981 г.
9. Филлипс Д. Методы анализа сетей. - М.: 1984 г.

8.6. «Карта памяти» по теме



Тема 9. Управление ресурсами проекта

9.1. Введение

Планирование и управление – это не только составление, оптимизация и отслеживание календарных планов, но и управление ресурсами, необходимыми для реализации проекта. При этом управление временем проекта неразрывно связано с управлением ресурсами, выделенными каждой задаче.

Изучив учебный материал данной темы, Вы:

- узнаете или пополните свои знания о видах проектных ресурсов и о том, как следует назначать их задачам при планировании проекта;
- узнаете, как связаны между собой для разных типов задач трудозатраты и длительности задач с загрузкой ресурсов;
- получите общее представление о выравнивании загрузки ресурсов.

В рамках темы рассматриваются следующие учебные вопросы:

- ресурсы проекта;
- распределение ресурсов и календарный план.

9.2. Ресурсы проекта

Для осуществления проекта необходимы ресурсы – исполнители, оборудование и материалы, используемые в проекте.

Ресурсы (Resources) – исполнители, оборудование и материалы, используемые для выполнения задач в проекте.

9.2.1. Назначение ресурсов при планировании

После составления детальной структуры декомпозиции работ (СДР – WBS) проекта и определения логических связей между задачами естественным шагом планирования является ***назначение ресурсов задачам***. Процесс назначения ресурсов и составления календарного плана проекта при этом является итеративным по своей природе:

- ресурсы назначаются задачам;
- с учетом назначенных ресурсов определяется длительность каждой задачи;

- затем, если суммарная длительность проекта по полученному плану не удовлетворяет исходным требованиям заказчика, необходимо изменить первоначальный план (например, добавить ресурсы для критических задач с целью сокращения их длительности).

Итерации заканчиваются получением приемлемого по срокам, а также по загрузке ресурсов плана. Сходный итеративный процесс возникает и при любом изменении плана на любой стадии проекта (например, если по каким-либо причинам необходимо ускорить выполнение той или иной задачи).

Выбор назначаемых ресурсов зависит от конкретных потребностей планирования и отслеживания хода выполнения проекта.

Например, можно назначать именованные (фактические) ресурсы, такие как «Виктор Петров», или универсальные ресурсы (ресурсы-роли), такие как «Программисты».

Если процесс планирования проекта только начинается, имеет смысл использовать универсальные ресурсы. При дальнейшей детализации плана целесообразно заменить все универсальные ресурсы на конкретные имена исполнителей. Дополнительным преимуществом такого подхода является то, что многие специализированные программные системы, предназначенные для планирования и отслеживания хода выполнения проекта (например, MS Project), предусматривают возможность не только просмотра списков исполнителей, назначенных конкретным задачам, но и организуют возможности коммуникации между исполнителями и менеджером проекта.

Например, при использовании MS Project Server исполнители могут с оговоренной периодичностью посылать на сервер информацию, скажем, о текущем проценте выполнения порученных им задач, а менеджер проекта при помощи MS Project сможет увидеть и проанализировать собранную таким образом информацию о ходе выполнения проекта.

Универсальные ресурсы (*Generic resources*) – прототипы ресурсов, используемые для указания умений и навыков, необходимых для выполнения конкретной задачи.

Кроме сотрудников-исполнителей, занятых в проекте, называемых обычно **людскими ресурсами** (human resources, HR), задачи проекта используют также оборудование и материалы – **материальные ресурсы** (material resources). К последним относятся, например, компьютеры, сетевое оборудование, бумага для принтеров, программы и т.п.

Материальный ресурс (Material Resource) – комплектующие и другие расходные материалы, которые используются для выполнения задач в проекте.

Проекты в области информационных технологий и тем более проекты по разработке программного обеспечения в гораздо большей мере зависят от людских ресурсов, чем от материальных. Их успех определяется, как правило, командой исполнителей, а не оборудованием, на котором она работает. И львиная доля затрат в таких проектах – это затраты на заработную плату сотрудников, а не на приобретение оборудования и расходных материалов. Учитывая эту особенность IT-проектов, при их планировании (в особенности предварительном, не детальном) часто для простоты материальные ресурсы просто игнорируют, считая их данностью. Так удобно считать, если проект не предусматривает закупки специализированного оборудования, а выполняется, например, на стандартном оборудовании компании (рабочих станциях, серверах, сетевом оборудовании и т.п.), на котором выполняются и все другие проекты. Тогда деятельность по технологическому обслуживанию, ремонтам, закупке нового оборудования и расходных материалов удобно вывести «за рамки» конкретного проекта, считая ее просто «обеспечивающей» деятельностью компании. При этом менеджер проекта может сосредоточиться на планировании использования и управлении только лишь людскими ресурсами – командой исполнителей. Исключениями в проектах в области информационных технологий являются:

- ✓ ситуации, в которых проект потребляет необычно (по сравнению со сходными проектами, ведущимися в компании) много какого-либо материального ресурса (например, проект по переводу с английского языка и адаптации документации к существующему программному комплексу может потребовать необычно большое количество бумаги, что стоит отметить в плане проекта);
- ✓ ситуациями, когда проект требует закупки специального сложного и/или дорогостоящего оборудования.

Отметим, наконец, что ресурсы следует назначать подзадачам, а не суммарным задачам. Это обеспечивает создание более эффективных и управляемых планов.

В качестве иллюстрации к сказанному выше, приведем пример, где покажем ввод информации о ресурсах и назначение ресурсов задачам с использованием MS Project.

	Название ресурса	Тип	Единицы измерения материалов	Краткое название	Группа	Макс. единиц	Стандартная ставка	Ставка сверхурочных	Затраты на исполз.	Начисление	Базовый календарь	Код
1	Программисты	Трудовой		П		100%	300,00р./ч	450,00р./ч	30,00р.	Пропорциональное	Стандартный	
2	Руководитель проекта	Трудовой		Р		100%	600,00р./ч	900,00р./ч	120,00р.	Пропорциональное	Стандартный	

Рис. 1. Таблица ресурсов в MS Project

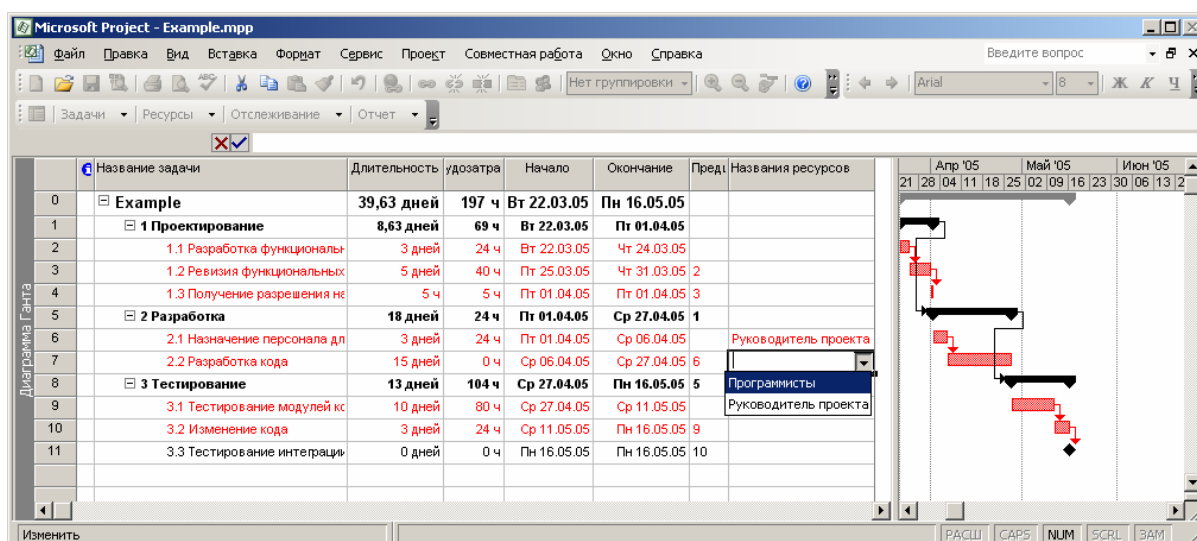


Рис. 2. Назначение ресурсов задачам

Менеджерам обычно нравится указывать назначенные задачам ресурсы прямо на диаграмме Ганта над каждой задачей. В случае, если назначенных ресурсов немного (например, 2-3 человека на задачу), такое представление является очень наглядным.

9.2.2. Загрузка ресурсов

Ресурсы никогда не бывают безграничны. Назначать ресурсы задачам надо, естественно, с учетом их фактического наличия. Это, однако, не означает, что, если в распоряжении менеджера проекта имеется лишь два тестера, то на задачу тестирования можно назначить лишь двоих тестеров. Можно назначить и больше, например, запланировав для них сверхурочную работу (скажем, работу по 10, а не по 8 часов в день), естественно, предусмотрев для этого соответствующую оплату. С формальной точки зрения загрузка этих

ресурсов будет не 100% (что соответствует 8 рабочим часам в день), а $10/8 \cdot 100\% = 125\%$ (в качестве замечания отметим, что в реальности счет надо вести не от стандартной цифры 8 рабочих часов в день, а от конкретного календаря проекта, который включает праздники, в т.ч. и корпоративные, и выходные, а также нерабочие часы). Злоупотреблять этим при планировании опасно, особенно в проектах по разработке программного обеспечения, так как «переработка», а особенно ее реальная эффективность у работников интеллектуального труда трудно контролируется, поэтому формально фигурирующая в плане цифра загрузки 125% не означает возрастание эффективности по сравнению с «номинальной» (при стандартной, т.е. стопроцентной загрузке) на 25%.

9.3. Распределение ресурсов и календарный план

Теперь настало время поговорить непосредственно о распределении ресурсов.

9.3.1. Трудозатраты, объем и длительность работы

Как связаны между собой длительность задачи (работы), загрузка ресурсов и общие трудозатраты на выполнение этой задачи (общее количество человеко-часов, затраченных на выполнение задачи)?

С формальной точки зрения связь определяется простой арифметической формулой: ***длительность, умноженная на загрузку ресурса, определяет необходимый объем работы (трудозатраты)***. Это означает, что операция, для выполнения которой необходимо 8 часов (длительность), с назначенным исполнителем на полное время (100% загрузки ресурса), потребует 8 рабочих часов (объем работы). Для выполнения операции, длительность которой равна 40 часам при загрузке ресурса на 10% времени, требуется всего 4 рабочих часа.

$\text{Объем работы} = \text{Длительность} \times \text{Загрузка}$
--

Например, если изменить оценку объема работы с 4 до 16 часов для операции длительностью 8 часов, то либо изменится длительность операции до 16 часов, либо загрузка ресурса возрастет до 200% (два исполнителя).

Все задачи можно разделить на три типа:

1. задачи с фиксированным объемом работ (фиксированными трудозатратами),
2. задачи с фиксированной продолжительностью,
3. задачи с фиксированной загрузкой ресурсов.

Для **задачи с фиксированной длительностью** любое изменение либо объема работы, либо загрузки ресурсов будет затрагивать только второй параметр, не изменяя длительности. Для операции длительностью 40 часов увеличение объема работы на 8 часов увеличит загрузку ресурса на 20%. Именно такие задачи в основном и встречаются в проектах в области информационных технологий, ибо принцип таких проектов: «не имеет значения, сколько людей будет работать над этой операцией, ее длительность не изменится» (это очень похоже на принцип «даже вдвое большее число женщин не смогут родить ребенка за время, меньшее 9 месяцев»). Независимо от того, сколько ресурсов или необходимой работы будет требоваться, задача всегда будет иметь одну и ту же длительность.

Для **задачи с фиксированной загрузкой** ресурсов любое изменение либо длительности, либо объема работы будет затрагивать другой параметр. Для операции с длительностью 40 часов и загрузкой ресурса в 10%, при увеличении объема работы до 8 часов, длительность увеличится до 80 часов. Выбор фиксированной загрузки ресурса наиболее предпочтителен для задач, у которых всегда одинаковый объем работы. В литературе такие задачи определяются как операции «одинакового усилия», такие как контроль, управление, отчетность. Чем больше времени они исполняются, тем больше соответствующий объем работы.

Наконец, **задачи с фиксированным объемом работы** (трудозатратами) часто встречаются в классических промышленных процессах (производство одежды, обуви, продуктов и т.п.) при наличии эффективного процесса производства. Например, в случае, если имеется достаточно машин, удвоение количества операторов этих машин увеличит производительность почти в два раза. Удвоение ресурсов приводит к тому, что необходимо вдвое меньше времени для того, чтобы произвести заданный объем. Но использовать в планировании задачи с фиксированным объемом работы следует с осторожностью. При этом не учитывается эффект обучения, взаимодействия внутри больших групп людей, стоимости распределения работ или стоимости смещения работ во времени.

При планировании менеджеру следует постоянно следить за типом каждой конкретной задачи проекта, и менять длительность в зависимости от загрузки ресурсов в соответствии с типом конкретной задачи.

9.3.2. Выравнивание загрузки ресурсов

Загрузку ресурсов-исполнителей удобно представлять в виде **графика загрузки ресурсов (resource graph)**, в котором по оси абсцисс отложено рабочее время (обычно в днях или более крупных единицах), а по оси ординат – загрузка каждого конкретного исполнителя.

В идеале следует:

- ☉ стремиться к тому, чтобы ресурсы были загружены равномерно в течение всего проекта или, по крайней мере, тех задач, на которые они назначены,
- ☉ не допускать перегрузки ресурсов (работы исполнителя с загрузкой более 100%), или существенной незагруженности ресурсов.

Конечно, эти правила не являются догмой. Так, в определенных ситуациях (речь об этом уже шла) перегрузку определенных ресурсов на некоторых важных этапах проекта можно допустить, запланировав ее как сверхурочную работу (естественно, с соответствующей повышенной оплатой, дополненной, возможно, и другими стимулами нематериального характера). Однако это, очевидно, не может быть нормой. Обычно сверхурочная работа вообще изначально не планируется, а является следствием «авралов», т.е. в конечном счете недостаточного опыта планирования (например, вследствие незнания менеджером-планировщиком реальных возможностей команды исполнителей) или плохого управления проектом (в частности, плохого управления рисками проекта). Вообще говоря, поэтому, план, допускающий перегрузку ресурсов, следует считать нереалистичным. Следует в этом случае изменить план таким образом, чтобы устранить перегрузку ресурсов. Действия, направленные на это, называются **выравниванием загрузки ресурсов (resource leveling)**.

Недостаточная загруженность («простой») ресурса, конечно, менее опасна, чем перегрузка, но она свидетельствует о недостаточно эффективном использовании ресурсов. Следовательно, в этом случае план также стоит модифицировать с целью оптимизации загрузки исполнителей. Самыми простыми методами выравнивания загрузки являются ввод дополнительного ресурса, либо изменения длительности задач, либо, если перегрузка возникает вследствие того, что один и тот же исполнитель участвует в нескольких работах, то сдвиг соответствующих работ или введение разрывов в расписание работ. Для борьбы с недостаточной загруженностью можно, наоборот, уменьшать длительность задач или, например, организовать календарный план так, чтобы один и тот же исполнитель мог участвовать одновременно в нескольких работах.

Как выравнивание загрузки ресурсов, так и, все действия, направленные на модификацию плана с целью оптимизации загрузки, невозможно формализовать. В общем случае они сводятся к изменению

календарного плана и переназначению ресурсов. Общего алгоритма здесь нет. Существующее программное обеспечение для менеджмента проектов может только помочь менеджеру в простейших расчетах. Например, MS Project в состоянии «автоматически выровнять» загрузку всех ресурсов за счет удлинения соответствующих задач (при фиксированном объеме работы), а также рассчитать, на сколько при этом должна увеличиться *длительность* каждой *задачи* (leveling delay). Для проектов в области информационных технологий, правда, от этого обычно мало пользы, ведь для этих проектов, как правило, уравнение

$$\text{Длительность} \times \text{Загрузка} = \text{const (фиксированный объем работы)}$$

имеет слабое отношение к реальности: уменьшение загрузки исполнителей в 2 раза не означает автоматического увеличения длительности в 2 раза (см. предыдущий раздел).

Поэтому для таких проектов лучше не пользоваться возможностями автоматического выравнивания ресурсов, предоставляемыми специальным программным обеспечением, а производить выравнивание «вручную», ориентируясь в основном на свой опыт аналогичных проектов и знание возможностей команды исполнителей.

9.4. Заключение

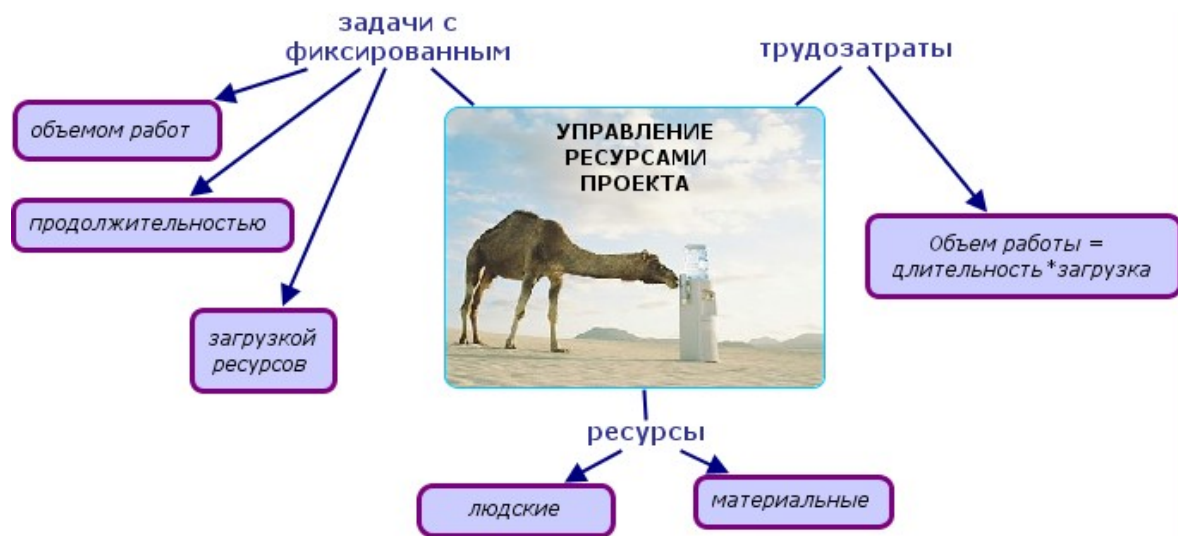
Управление ресурсами проекта начинается с назначения ресурсов задачам проекта. При этом назначения и загрузка ресурсов неразрывным образом связаны с календарным планом проекта, так что любое изменение назначений (или загрузки) ресурсов приводит к изменению календарного плана, и наоборот. Следует стремиться к тому, чтобы ресурсы были загружены по возможности равномерно в течение проекта, а также не допускать перегрузки ресурсов или существенной незагруженности ресурсов.

9.5. Список использованной и рекомендованной литературы

1. Баранов С. Н. Управление программным проектом. Лекции по спецкурсу "Технология программирования". - СПб: Санкт-Петербургский государственный электротехнический университет, рукопись, 1998.
2. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. - СПб.: Символ-Плюс, 1999.
3. А.К. Гульяев. Microsoft Office. Project Server 2003. Project Professional 2003. Управление проектами. Практическое пособие. Корона-принт, 2004.

4. Эдвард Йордон. Путь камикадзе. Как разработчику программного обеспечения выжить в безнадежном проекте. - М.: ЛОРИ, 2001.
5. Куперштейн В.И. Современные информационные технологии в делопроизводстве и управлении.-СПб.: 1999.
6. Полковников Е. В., Полковникова А. В., Управление проектами с использованием Time Line, М., МИФИ, 1994.
7. Пресняков В.Ф. Информационные технологии в управлении проектами. М. 1998 г.

9.6. «Карта памяти» по теме



Тема 10. Организация проектного финансирования

10.1. Введение

Часто (но не всегда) в задачу менеджера проекта входит частичное или полное управление финансами проекта. Финансовый менеджмент – это весьма специальный вид деятельности, требующий, как правило, специального образования. Не претендуя на изложение всех вопросов управления финансами проекта, в этой теме мы рассмотрим только те из них, с которыми придется столкнуться любому менеджеру программного проекта.

Изучив учебный материал данной темы, Вы:

- узнаете или пополните свои знания о функциях менеджера проекта в управлении финансами и об основных принципах управления финансами проекта;
- узнаете способы оценки финансовой эффективности проектов;
- получите общее представление об оценках стоимостей и планировании затрат.

В рамках темы рассматриваются следующие учебные вопросы:

- финансовый менеджмент;
- оценки стоимостей и планирование затрат.

10.2. Финансовый менеджмент

Среди различных видов менеджмента особое место занимает финансовый менеджмент. Это весьма специальный вид деятельности, требующий, как правило, и специального образования. Недаром в любой серьезной компании существует особая должность финансового менеджера (финансовый директор). Данный курс не посвящен финансовому менеджменту (на эту тему существует масса специальной литературы), так что мы будем затрагивать вопросы управления финансами лишь настолько, насколько с ними приходится сталкиваться менеджеру проектов в области информационных технологий или, более конкретно, в области разработки программного обеспечения.

10.2.1. Менеджмент проекта и финансовый менеджмент

В какой мере менеджеру программного проекта приходится в своей деятельности сталкиваться с вопросами управления финансами

проекта, контролем расходов и доходов, оценкой финансовой эффективности проектов?

К сожалению, общего ответа на этот вопрос на сегодняшний день, по-видимому, не существует. Все зависит от практики конкретных компаний. Условно можно выделить три используемые в настоящее время модели взаимоотношения менеджмента проекта с финансами.

Модель 1. Полноценное управление финансами проекта.

Менеджер проекта полностью контролирует всю финансовую сторону проекта, от согласования с заказчиком договорной цены и графика платежей (в случае, если речь идет о разработке программного обеспечения на заказ) или поиска финансирования (в случае, например, инициативной разработки), до планирования и текущего контроля расходов, обеспечения стабильности платежей и, в конечном счете, обеспечения финансовой эффективности проекта для компании.

Это довольно редко применяемая на сегодняшний день модель. Ее область применения ограничена в основном достаточно мелкими компаниями или «бригадами» разработчиков, не ведущими одновременно большого числа проектов. В больших компаниях так тоже бывает, например, когда вся компания создана фактически под единственный (пусть и очень большой) проект. Дело в том, что практически у любой компании (за исключением разве что самых мелких) существуют значительные расходы, не связанные непосредственно с проектной деятельностью: даже если компания проектов вообще не ведет, она все равно расходует средства. Например, зарплату сотрудникам (если только они не работают по договорам подряда, что опять-таки возможно, как правило, для очень мелких компаний, или собранных «под конкретный проект») приходится выплачивать постоянно, независимо от реальной загрузки сотрудников в проектах. Постоянными являются и расходы на техническое обслуживание оборудования, уборку помещения, административные расходы. «Разнести» эти расходы по отдельным проектам практически невозможно (а если и можно, то весьма неестественно), так что управлять ими гораздо естественнее финансовому менеджеру компании.

Есть и еще одно немаловажное ограничение на использование этой модели: руководство компании (являющееся представителем собственников) далеко не всегда с энтузиазмом относится к раскрытию всей финансовой «подноготной» компании менеджерам среднего звена.

Модель 2. Менеджер проекта не имеет никакого отношения к управлению финансами проекта.

В этом случае менеджер не имеет отношения и к финансовым переговорам с заказчиком (этим занимаются, например, специалисты по

продажам компании, или отдел по работе с клиентами, либо высшее руководство компании) и часто даже ничего не знает о финансировании проекта.

Несмотря на крайнюю неэффективность модели, она, увы, весьма распространена. Часто использование такой модели – это вынужденная мера. Например, если заказчики находятся в США, там же находятся и представители компании по работе с клиентами, а основное производственное подразделение находится в России, то менеджер проекта, непосредственно руководящий командой исполнителей, не имеет никакой реальной возможности влиять на финансирование проекта заказчиком (не в ущерб своей работе по руководству исполнителями).

Почему любая модель, в которой менеджер проекта непосредственно не участвует в управлении финансами проекта, в конечном итоге неэффективна для компании? Потому что компании существуют с целью получения прибыли, поэтому любой проект направлен на достижение именно этой цели, а менеджер проекта, не имеющий отношения (или имеющий весьма слабое отношение) к финансам, просто не в состоянии оптимизировать проект с целью достижения максимальной финансовой эффективности.

Модель 3 (промежуточная).

Менеджер проекта управляет доходами проекта, согласует с заказчиком договорную цену и график платежей (и/или участвует в поиске дополнительного стороннего финансирования), отвечает за своевременное поступление средств, иногда (частично) управляет расходами проекта (как правило, лишь теми, которые легко отнести именно к конкретному проекту – зарплата привлеченного персонала, закупки специального оборудования и т.п.).

Это частично решает проблему неэффективности: менеджер, хоть и не полностью, «видит» финансы проекта, и в состоянии пытаться оптимизировать финансовый результат проекта. Те общие расходы компании, которые трудно или нежелательно для руководства компании раскрывать менеджеру проекта, могут быть, тем не менее, включены в расходную смету проекта по статьям «накладные расходы» или «отчисления на развитие». В государственных учреждениях могут использоваться различные нормативы (например, связанные с ограничениями законодательного характера).

В идеале с точки зрения финансовой эффективности для компании следует ориентироваться на первую модель (менеджер проекта является финансовым менеджером) или, по крайней мере, стремиться к ней, если полное достижение этой модели не реализуемо.

10.2.2. Функции менеджера проекта в управлении финансами

Далее мы будем рассматривать только ситуации, в которых менеджеру проекта приходится, хотя бы частично, управлять финансами проекта. Как уже сказано, в идеале следует всегда стремиться к такому положению дел.

В этом случае менеджер:

- ✓ на прединвестиционной стадии – производит грубую оценку стоимости проекта,
- ✓ в дальнейшем при конкретизации требований уточняет оценку стоимости, формируя параллельно смету и бюджет проекта,
- ✓ участвует в переговорах с заказчиком о согласовании договорной цены и графика платежей (поэтапная оплата),
- ✓ (возможно) участвует (если это необходимо) в поиске стороннего финансирования,
- ✓ (возможно) отвечает перед командой исполнителей и перед руководством своей фирмы за своевременное бесперебойное финансирование (с этой целью, при необходимости, ведет дополнительные переговоры с заказчиком или ищет дополнительное стороннее финансирование).

10.2.3. Общие принципы управления финансами проекта

Для понимания общих принципов управления финансами проекта вообще-то полезно понимать общие принципы управления финансами компании. Дело в том, что выполнение проекта есть с точки зрения финансиста лишь одно из средств в достижении финансовых целей компании. Поэтому и эффективность проекта следует оценивать по тем же правилам, по каким оценивается любая финансовая деятельность компании. Финансовых менеджеров этому учат специально в рамках большого (и весьма серьезно математизированного) курса «Корпоративные финансы». Не претендуя на изложение такого объемного материала, приведем лишь основные положения, которые в принципе должны быть понятны и людям без специального финансового образования.

1. Долгосрочной целью деятельности компании является получение прибыли.

Важно понимать, что далеко не все операции компании, связанные с движением денежных средств, приводят к изменению прибыли. Например, покупка оборудования, становящегося на баланс (т.е.

становящегося активом) компании, связана с движением денежных средств, но не приводит к изменению прибыли. И наоборот, не все операции, связанные с изменением прибыли, связаны с движением денежных средств.

Например, продажа продукции в кредит не связана с притоком денежных средств, но изменяет прибыль компании.

В долгосрочной перспективе чистый приток денежных средств (поступающие платежи минус исходящие платежи) равен изменению чистой прибыли, но в краткосрочной перспективе это не так. Поэтому в краткосрочной перспективе компания может испытывать трудности с ликвидностью, даже если ее деятельность является прибыльной. А именно проблемы с ликвидностью и являются причиной разорения компаний. Таким образом, краткосрочной целью деятельности компании является максимизация чистого притока денежных средств.

2. Любой программный проект с точки зрения финансовых целей принято считать краткосрочным. Поэтому **цель управления финансами проекта – максимизация чистого притока денежных средств, а не максимизация прибыли.**

10.2.4. Оценка финансовой эффективности проекта

Для оценки финансовой эффективности проекта используются, оценки, основанные на анализе потоков платежей. Наиболее популярным (и наиболее простым) критерием оценки финансовой эффективности является **NPV (net present value) – чистая дисконтированная стоимость проекта.**

NPV рассчитывается по формуле

$$NPV = CF_0 + \frac{CF_1}{(1+r_1)} + \frac{CF_2}{(1+r_2)^2} + \dots + \frac{CF_n}{(1+r_n)^n}$$

где

CF_i – чистый поток платежей в период i ,

n – число периодов,

r_i – дисконтная ставка в период i (в процентах за период).

Финансовый менеджер должен стремиться к максимизации NPV проекта.

Стоит отдельно прокомментировать понятие **дисконтной ставки**. Это «стоимость» финансирования проекта, иначе говоря, это та эффективная процентная ставка, под которую можно «занять» денег для

финансирования проекта. Естественно, она зависит от источника финансирования. Такими источниками обычно являются следующие:

- ♦ Собственный капитал компании и в том числе (что с точки зрения финансиста то же самое) доходы от реализации продукции, полученной в результате проекта (деньги заказчика или покупателя). В этом случае в качестве дисконтной ставки принято использовать **взвешенную среднюю стоимость собственного капитала** (WACC – weighted average cost of capital).
- ♦ Краткосрочные кредитные средства, привлекаемые под конкретный проект. В этом случае в качестве дисконтной ставки принято использовать **процентную ставку по кредиту**.

При расчете дисконтной ставки следует учитывать не только чистую стоимость капитала (т.е. процентную ставку на капитал), но и делать поправку на действующую налоговую ставку.

Чаще всего в расчетах используют взвешенную среднюю стоимость собственного капитала. Приведем пример расчета этой величины. Рассмотрим компанию со структурой капитала компании, приведенной в таблице 1. Стоимости каждой из составных частей капитала также приведены в этой таблице.

Таблица 1.

Пример: данные для расчета WACC.

Составная часть капитала компании	Стоимость (%/год)	% полного капитала
Чистая прибыль	10%	10%
Обыкновенные акции	12%	10%
Привилегированные акции	9%	15%
Долговые обязательства	10%	65%

Расчет взвешенной средней стоимости собственного капитала для этого примера показан в таблице 2.

Таблица 2.

Пример: расчет WACC.

Составная часть капитала	Стоимость	X	% полного	Итого
--------------------------	-----------	---	-----------	-------

компания	(%/год)		капитала	
Чистая прибыль	10%	X	10%	1,00%
Обыкновенные акции	12%	X	10%	1,20%
Привилегированные акции	9%	X	15%	1,35%
Долговые обязательства	10%	X	65%	6,50%
Итого				10,05%

Таким образом, в примере WACC равна 10,05 %/год.

Приведем теперь пример расчета NPV для проекта, график движения денежных средств по календарным периодам которого показан в таблице 3 (+ означает поступление средств, - означает платеж). Предположим, что стоимость капитала равна 10% за календарный период (все периоды предполагаются одинаковой длительности).

Таблица 3.

Пример: данные для расчета NPV.

Номер периода	Платежи, \$
0	-7,500
1	+2,000
2	+2,000
3	+2,000
4	+2,000
5	+2,000

Расчет NPV для этого примера показан в таблице 4.

Таблица 4.

Пример: расчет NPV.

Номер периода	Платежи, \$	Дисконтированные платежи, \$	Накопленный дисконтированный платеж, \$
0	-18000	-18000	-18000
1	8000	7272,73	-10727,27

2	6000	4958,68	-5768,59
3	4000	3005,26	-2763,33
4	4000	2732,05	-31,28
5	3000	1862,76	1831,48
ИТОГО NPV			1831,48

Расчет NPV дает ответ на еще один важный для финансового менеджера вопрос: когда проект начинает окупаться, иными словами, позволяет определить *точку безубыточности* (breakeven point) проекта. Из таблицы 4 видно, что с учетом стоимости капитала проект выходить на окупаемость не раньше окончания четвертого календарного периода, т.е. через пять календарных периодов после начала.

10.3. Оценки стоимостей и планирование затрат

10.3.1. Грубая предварительная оценка стоимости проекта

Грубая предварительная оценка стоимости проекта должна быть произведена менеджером на самом начальном этапе, когда по сути дела еще точно не известно, какое в точности программное обеспечение должно разрабатываться (есть только разве что самая общая идея, полученная из предварительных контактов с заказчиком). В процессе переговоров с заказчиком и детализации требований эта оценка уточняется и в уже уточненном виде является основой определения договорной цены. Адекватная оценка стоимости проекта важна как для заказчика, так и для исполнителя проекта.

Для получения такой оценки наиболее естественно использовать опыт компании в выполнении аналогичных проектов. Если соответствующей статистики по стоимостям аналогичных проектов, выполненных компанией, нет, то помочь в оценке трудоемкости и стоимости разработки программного обеспечения могут специально разработанные *эвристические модели*.

Вероятно, одной из наиболее известных моделей данного рода является *конструктивная модель стоимости* (Constructive Cost Model – **COCOMO**), разработанная в конце 70х годов Барри Боэмом (Barry Boehm). Построенная на основе анализа ряда проектов, выполненных в основном в интересах Министерства Обороны США, она устанавливает в виде простой эвристической формулы соответствие между размером системы в тысячах условных строк кода (KSLOC – Kilo Source Lines of Code) и трудоемкостью (в человеко-часах), а значит, при наличии

оценки усредненной по компании стоимости человеко-часа, и стоимостью разработки.

Базовый тип модели COCOMO рассчитан только на относительно маленькие проекты, разрабатываемые командами, хорошо знакомыми с прикладной областью. Более развитые модели, основанные на COCOMO, вводят в расчет до 15 поправочных факторов, принадлежащих к одной из четырех категорий:

- ♦ **атрибуты продукта**, такие, как его сложность и требования к его надежности;
- ♦ **атрибуты системы**, такие, как ограничения на оперативную память и время выполнения;
- ♦ **атрибуты команды исполнителей**, такие, как опыт в прикладной области;
- ♦ **атрибуты проекта**, такие, как используемые средства разработки.

Наиболее продвинутые модели такого рода дополнительно вводят разбиение по стадиям жизненного цикла проекта.

Рассмотрим в качестве примера расчет трудоемкости, стоимости и длительности разработки с использованием модели COCOMO (это самая простая и, наверное, самая употребительная модель, основанная на тысячах условных строк кода, причем именно своей простотой она выгодно отличается от других моделей). Заложенные в модель формулы элементарны:

$$\begin{aligned} \text{Объем работы (трудоемкость в человеко-месяцах)} \\ &= 3.0 * \text{EAF} * (\text{KSLOC})^{1.12} \\ \text{Длительность} &= 2.5 * (\text{Работа})^{0.35} \end{aligned}$$

Предполагается, что рассчитанные величины не включают затраты на планирование и определение требований. Здесь EAF – поправочный коэффициент, равный произведению 15 поправочных факторов, оценивающих упомянутые выше атрибуты. Для конкретности рассмотрим пример оценки трудоемкости и длительности разработки программного комплекса объемом примерно 170.000 строк кода (при этом, естественно, не считаются комментарии, директивы компилятору и отладочные инструкции, поэтому этот параметр также принято называть KDSI — Kilo Delivered Source Instructions). Пример мы взяли с сайта <http://www.saitistica.ru> – это пример реально существующей платформы управления интернет-системами (корпоративным web-контентом).

Расчет EAF для конкретной компании, выполняющий проект, показан в таблице 5 (предполагаем только, что в компании уже имеют

большой опыт в программировании подобных приложений, и в использовании соответствующих элементов разработки, а остальные параметры для простоты считаем номинальными).

Таблица 5.

Пример: расчет EAF

Фактор	Значение	Влияние
Знание языка программирования	Номинальное	1.0
Ограничение времени выполнения	Номинальное	1.0
Размер базы данных	Номинальный	1.0
Межремонтный срок службы компьютера	Номинальный	1.0
Знание виртуальной машины	Номинальное	1.0
Изменчивость виртуальной машины	Номинальная	1.0
Использование программных инструментов	Высокое	0.9
Использование современных методов	Номинальное	1.0
Ограничение объема памяти	Низкое	1.0
Знание приложений	Высокое	0.95
Ограничение по срокам разработки	Номинальное	1.0
Требуемая надежность	Номинальная	1.0
Сложность продукта	Номинальная	1.0
Способности персонала/команды	Номинальные	1.0
Способности аналитика	Номинальные	1.0
ИТОГО EAF		$1.0^{13} * 0.9 * 0.95 = 0.855$

Расчет трудоемкости и длительности разработки дает

$$\begin{aligned}\text{Трудоемкость} &= 3.0 * 0,855 * (170)^{1.12} \\ &= 2.8 * 0.855 * 474 \\ &= 808 \text{ человеко-месяца,}\end{aligned}$$

$$\text{Длительность} = 2.5 * (808)^{0.35} = 2.5 * 9.5 = 26 \text{ месяцев,}$$

без учета планирования и определения требований

Заметим, что фигурирующие в формулах фиксированные мультипликативные константы и показатели степени могут меняться в зависимости от сложности системы. Стандартная модель COSOMO предусматривает три уровня сложности систем – простые, промежуточные и сложные (каждому уровню соответствует свой набор констант). В данном примере мы считали сложность системы промежуточной.

В современных развитых вариантах модели COSOMO заложены и распределение трудоемкости и длительности по стадиям жизненного цикла в зависимости от выбранной модели жизненного цикла. Так, для классической водопадной модели имеем

Вид деятельности	Трудоемкость (%)	Длительность (%)
Планирование и определение требований	(+8)	(+36)
Проектирование продукта	18	36
Детальное проектирование	25	18
Кодирование и тестирование отдельных модулей	26	18
Интеграция и тестирование	31	28
ИТОГО:	108%	136%

Это дает итоговые цифры для приведенного примера:

Трудоемкость 872 человеко-месяца
Длительность 35 месяцев.

В модели COCOMO можно получить и распределение работ по видам деятельности WBS. Для «стандартной» (в смысле COCOMO) структуры WBS в данном примере имеем^

Этап проекта	Бюджет (%)	Трудоемкость (человеко-месяцы)
Анализ требований	4	35
Проектирование продукта	12	105
Программирование	44	384
Планирование тестирования	6	52
Верификация и аттестация	14	122
Канцелярия проекта	7	61
Управление конфигурацией и обеспечение качества	7	61
Создание руководств	6	52
ИТОГО	100%	872

Наконец, можно получить и прогноз на требуемое количество работников на протяжении всего цикла создания продукта.

Для использования величины KSLOC в расчетах полезно привести сравнение количества строк кода для различных реально существующих систем.

Программа	Количество строк	Примечания
Communiware	80 тыс.	
Navision Axapta	400 тыс.	
Галактика	1.5 млн.	
SAP R/3	10 млн.	
SUN StarOffice 6	9 млн.	
Система управления огнем для истребителя F-16	236 тыс.	Стоила \$85 млн. Усилия по сопровождению программ комплекса,

		их улучшению и устранению ошибок потребовали дополнительно \$250 млн.
Компьютерная сеть Олимпийских игр в Солт Лейк Сити	10 млн.	Стоимость – \$300 млн.
Электронная система торгов NASDAQ	11 млн.	Только на поддержку в 2000 году потрачено \$55 млн.
Solaris 7	12 млн.	
Windows NT 4.0	16 млн.	
Компьютерная системе Агентства национальной безопасности США	25 млн.	
Программное обеспечение проекта НАСА "Space Shuttle"	40 млн.	
Windows 2000	40 млн.	
Debian Linux	55 млн.	Самый полный вариант, более 10 CD

Существенным недостатком описанных моделей является то, что они основаны на тысячах условных строк кода, как метрике размера программного комплекса. Эта метрика была популярна в 60-70х годах прошлого века, но в настоящее время не считается адекватной мерой объема проекта.

Видимо, одной из первых попыток отойти от данной метрики, была разработка Аланом Альбрехтом (Alan Albrecht) в середине 70-х годов **метода функциональных точек** с целью разработки механизма предсказания усилий, сопряженных с разработкой программных систем. Метод был впервые опубликован в 1979 году. В 1984 году Альбрехт усовершенствовал свой метод и с 1986 года, в котором была сформирована Международная Ассоциация Пользователей Функциональных Точек (International Function Point User Group – IFPUG), было опубликовано несколько ревизий метода.

Чарльз Саймон (Charles Symon) разработал другой, аналогичный, но несколько более логичный и использующий более современную терминологию, **метод функциональных точек Mark II**. В отличие от FPA IFPUG, МК II FPA использует единое понятие транзакции, имеющей вход, обработку и выход. МК II FPA принят в качестве национального стандарта Великобритании. Другими аналогичными методами, являются Feature Points, разработанный Кэйперсом Джонсом (Capers Jones) и 3D Points, разработанный в компании Boeing.

Со временем модель COSOMO оказалась устаревшей в значительной своей части. Поэтому на ее основе была разработана модель **COSOMO II**, опубликованная в 1999 году. Она усовершенствует оригинальную модель в следующих основных направлениях:

- ✓ использование входных данных, доступных на ранних этапах жизненного цикла системы для оценки ее сложности (в частности, использование функциональных точек);
- ✓ подходы, основанные на повторном использовании, включая интеграцию коммерческих продуктов, реинжиниринг, генерацию приложений;
- ✓ объектно-ориентированные подходы, поддерживаемые распределенным ПО промежуточного слоя;
- ✓ влияние зрелости процессов разработки.
- ✓ новые – циклические и обобщенные – модели процессов разработки;

В течение восьмидесятых годов в СССР также на основе COSOMO были разработаны собственные модели оценки трудоемкости разработки программных систем, утвержденные Госкомтруда в 1986 году.

10.3.2. Планирование затрат и составление смет

Планирование затрат (Budgeting) – определение потребности в финансировании по этапам жизненного цикла проекта.

При составлении детального плана проекта с назначением ресурсов задачам достаточно легко определить с учетом стоимостей каждого из ресурсов (обычных и сверхурочных ставок оплат труда исполнителей, стоимостей материальных ресурсов) как суммарные затраты на проект, так и распределение этих затрат во времени. С точки зрения финансового менеджмента важны оба типа информации. В составлении смет и графиков расходов может помочь любое

стандартное программное средство для управления проектом, например, MS Project.

Сводка суммарных затрат по проекту (обычно называемая **сметой проекта**) дает общую стоимость проекта. О стандартных разделах сметы (типах затрат) программного проекта речь уже шла в предыдущих главах. Удобно разбивать смету на основные этапы проекта (например, по основным задачам). Представленная в таком виде смета (обычно ее называют **бюджетом проекта**) показывает, какие задачи являются наиболее дорогостоящими. Для минимизации стоимости проекта, естественно, следует оптимизировать именно наиболее дорогостоящие задачи.

Бюджет проекта (Project Budget) – сметная стоимость, распределенная по периодам выполнения проекта.

Анализ распределения затрат во времени (графиков расходования средств) нужен для того, чтобы правильно планировать поступление денежных средств от заказчика (или со стороны). Здесь важен принцип: **все расходы должны быть покрыты соответствующим притоком денежных средств.**

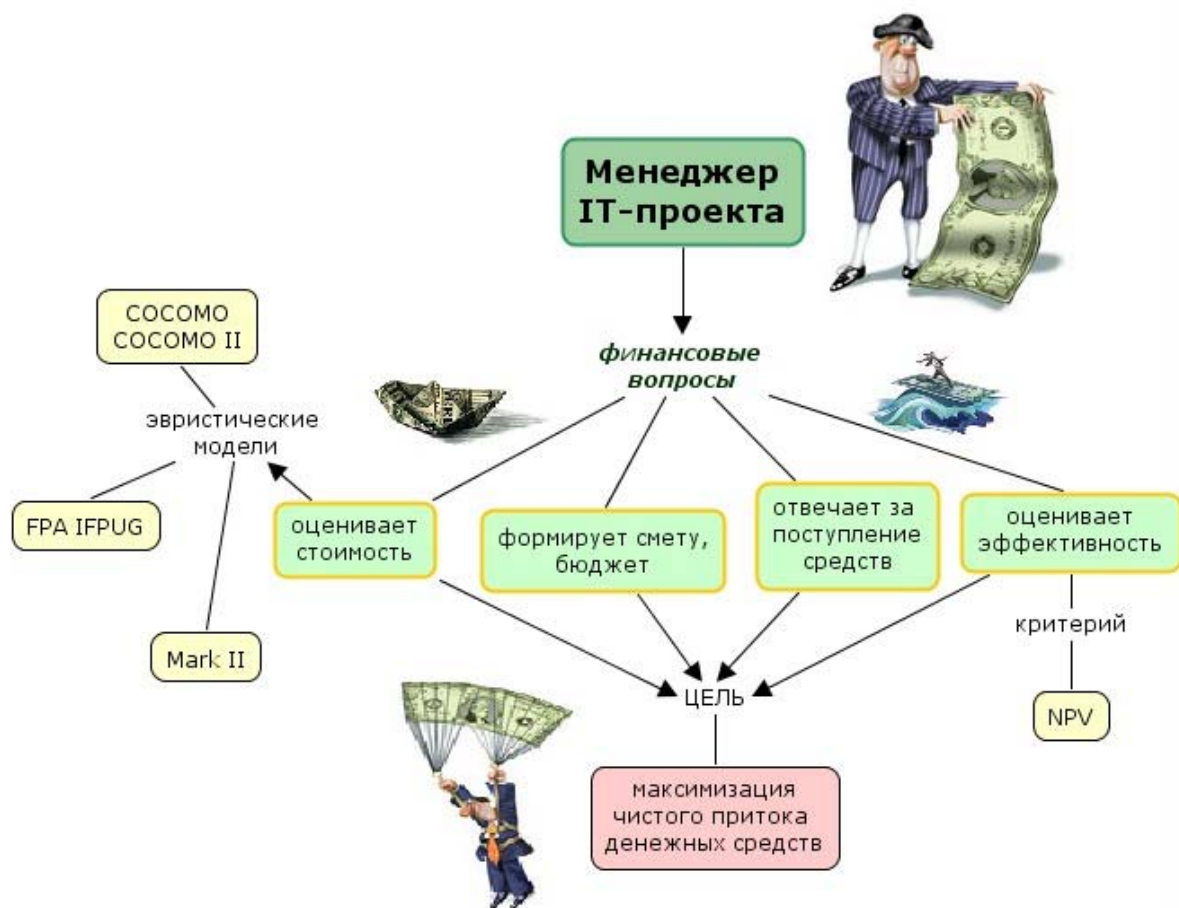
10.4. Заключение

Обычно менеджер проекта в той или иной мере участвует в управлении финансами проекта. Для этого ему стоит знать о том, как оценивать эффективность проекта.

Как правило, менеджер проекта:

- производит первоначальную грубую оценку стоимости проекта,
- в дальнейшем при конкретизации требований уточняет оценку стоимости,
- участвует в переговорах с заказчиком о согласовании договорной цены и графика платежей (поэтапная оплата),
- участвует (если это необходимо) в поиске стороннего финансирования,
- отвечает перед командой исполнителей и перед руководством своей фирмы за своевременное бесперебойное финансирование (с этой целью, при необходимости, ведет дополнительные переговоры с заказчиком или ищет дополнительное стороннее финансирование).

10.5. «Карта памяти» по теме



10.6. Список использованной и рекомендованной литературы

1. Баранов С. Н. Управление программным проектом. Лекции по спецкурсу "Технология программирования". - СПб: Санкт-Петербургский государственный электротехнический университет, рукопись, 1998.
2. Boehm B. W. Software Engineering Economics. Prentice Hall, 1981.

Тема 11. Управление рисками проекта

11.1. Введение

Рисками называют негативные события вероятностного характера, отрицательно влияющие на исход проекта. Для успешной реализации проектов одной из основ управления проектом должно быть управление рисками. Оно представлено как одно из девяти основных областей знаний в области управления проектами, описанных PMI (Американским институтом управления проектами).

Изучив учебный материал данной темы, Вы:

- узнаете или пополните свои знания о том, что такое риск, и о характеристиках риска;
- узнаете или пополните свои знания о об основных источниках риска в проектах по разработке программного обеспечения;
- получите общее представление о порядке управления рисками проекта по разработке программного обеспечения.

В рамках темы рассматриваются следующие учебные вопросы:

- риски при разработке программного обеспечения;
- порядок управления рисками программных проектов.

11.2. Риски при разработке программного обеспечения

В настоящее время эффективность и результативность процесса разработки программного обеспечения в целом оставляет желать лучшего. Несмотря на заметные успехи технологии программирования, остается весьма значительной доля проектов по разработке программного обеспечения, которые нельзя считать вполне успешными. Наряду с эффектными достижениями имеются и сравнительно многочисленные досадные неудачи. К сожалению, до сих пор слишком часто приходится делать вывод, что программирование – это рискованный бизнес, программы ненадежны, а программисты неуправляемы.

Так, в исследовании «Хаос», проведенном Standish Group, приводятся результаты проектов разработки приложений:

- среднее превышение времени разработки — 222%;
- среднее превышение затрат на разработку — 189%;
- отставание в удовлетворении требований пользователя на 27 месяцев.

В исследовании «Хаос» также сообщается, что лишь:

- 25% проектов внедрены успешно,
- 28% проектов остановлены,
- в 46% проектов возникли значительные проблемы с поставкой.

Очевидно, что разработка приложений не всегда завершается успешно и часто связана с задержками или превышением бюджета. Поэтому неудивительно, что менеджеры проектов стремятся улучшить управление рисками в разработке программного обеспечения.

Основы математической теории принятия управленческих решений в условиях риска и неопределенности изложены в теме 3. В данной теме рассматриваются более конкретные методы управления рисками применительно к менеджменту проектов разработки программных продуктов.

11.2.1. Риск и неопределенность

Риск – это негативное событие вероятностного характера, отрицательно влияющее на исход проекта.

Важно понимать, что риск – это *вероятное* событие, которое может произойти, а может и не произойти. Если негативное событие обязательно произойдет, то это уже не риск, а постоянно действующий фактор.

Например, «болезнь или увольнение менеджера проекта» – это риск, а «отсутствие опыта разработки у персонала» – это фактор.

11.2.2. Характеристики риска

Первой характеристикой риска является вероятность того, что рисковое событие произойдет. Вероятности принято задавать числом от 0 до 1 или в процентах.

Второй важной величиной, характеризующей риск, является **ущерб** в случае реализации риска. Эта величина определяет, насколько сильно пострадает проект, если рисковое событие произойдет. Например, риск «болезнь или увольнение менеджера проекта» несомненно, нанесет ущерб проекту, но если процесс разработки организован правильно, то реализация этого риска не является катастрофой для проекта. В то же время риск «потеря всех данных проекта» может стать полной катастрофой, если это печальное событие произойдет накануне выпуска первой версии продукта.

Существуют различные способы задания величины ущерба, наиболее простой и понятный – указать долю общего бюджета проекта, которая будет безвозвратно потеряна, если рисковое событие произойдет. В таком случае ущерб также можно задавать числом от 0 до 1 или в процентах.

Две введенные величины, вероятность и ущерб, позволяют сравнивать риски между собой. Как правило, в качестве меры, или величины риска выбирается математическое ожидание ущерба.

Величина риска – это математическое ожидание ущерба.

Например, пусть вероятность рискового события «болезнь или увольнение менеджера проекта» составляет 10%, и ущерб также составляет 10%. Тогда величина риска составляет 1%. Пусть вероятность рискового события «потеря всех данных проекта» – 1%, а ущерб – 100%. Тогда величина риска также составляет 1%. Эти риски равны по величине. Если вероятность рискового события «недостаточные навыки программирования» равна 10%, а ущерб – 50%, то величина такого риска 5%, он существенно больше предыдущих.

Риски подразделяются на:

- **устранимые**, т.е. такие, которых с помощью специальных мер можно избежать или которые можно предотвратить;
- **неустранимые**, т.е. такие, повлиять на которые менеджер проекта не может.

Например, риск «потеря всех данных проекта» можно практически преодолеть, проводя резервное копирование всех данных каждый день и сохраняя резервные копии в сейфах трех разных банков. Риск «прекращение финансирования проекта в результате банкротства инвестора» менеджер проекта, как правило, предотвратить не может.

Различие между преодолением и предотвращением риска заключается в следующем:

- **Преодоление риска** – это проведение мероприятий, в результате которых вероятность рискового события сводится к нулю.
- **Предотвращение риска** – это проведение мероприятий, в результате которых ущерб риска сводится к нулю.

В обоих случаях величина риска сводится к нулю, поэтому преодоление и предотвращение вместе называются **устранением риска**.

Например, риск «потеря всех данных проекта» можно преодолеть, проводя резервное копирование, а риск «болезнь или увольнение

менеджера проекта» можно предотвратить, подготовив заранее и введя в курс дела еще одного резервного менеджера.

Для устранимых рисков очень важной характеристикой является стоимость мероприятий по преодолению или предотвращению риска. Для единообразия стоимость устранения также можно задавать числом от 0 до 1 или в процентах. Например, стоимость преодоления риска «потеря всех данных проекта» можно оценить в 1%, а стоимость предотвращения риска «болезнь или увольнение менеджера проекта» более реалистично оценить в 5%.

11.2.3. Причины возникновения рисковых ситуаций

Перечислим некоторые из наиболее типичных и важных причин, приводящих к возникновению рисковых ситуаций в проектах по разработке программного обеспечения. Это:

- ✓ нереальная оценка требуемого времени реализации проекта и выделяемого бюджета;
- ✓ нереальная оценка возможностей команды исполнителей;
- ✓ недостаточное количество и квалификация команды исполнителей;
- ✓ недостаточное владение исполнителями инструментарием разработки;
- ✓ ошибки определения требований к разрабатываемому ПО (в т.ч. недостаточная детализация требований);
- ✓ нарушения основных правил процессов разработки (например, нарушения в управлении версиями, приводящие к потере версий);
- ✓ непрерывное изменение требований к разрабатываемому ПО по ходу проекта;
- ✓ существенное изменение рыночной ситуации, делающее бессмысленным следование первоначальным планам (например, появление на рынке доступного ПО, превосходящего по возможностям разрабатываемое);
- ✓ непрерывное изменение "правил игры" в команде исполнителей или группе проекта (правил коммуникации, распределения ответственности, распределения обязанностей);
- ✓ ошибки проектирования архитектуры ПО;
- ✓ ошибки разработки;
- ✓ ошибки интеграции;
- ✓ недостатки внешнего обслуживания;
- ✓ технические и программные сбои;

- ✓ злоумышленные действия сторонних лиц (например, "атаки" хакеров).

11.3. Порядок управления рисками программных проектов

Существует множество моделей управления рисками проектов, в т.ч. проектов по разработке программного обеспечения. Все эти модели в значительной мере схожи между собой.

В редакции РМВОК – 2000, управление рисками это «систематические процессы, связанные с идентификацией, анализом рисков и принятием решений, которые обеспечивают минимизацию негативных последствий наступления рисков событий и максимизацию вероятности и последствий наступления позитивных событий».



Рис. 1. Управление рисками по PMBOK

Процесс управления рисками включает в себя

- **Планирование управления рисками** – планирование деятельности по управлению рисками проекта, включая набор методов, средств и организации управления рисками.
- **Идентификация факторов риска** – определение рисков, способных повлиять на проект, и документирование их характеристик.
- **Оценка рисков** – качественный и количественный анализ рисков с целью определения их влияния на проект.
- **Планирование реагирования на риски** – разработка мер, обеспечивающих минимизацию вероятности и ослабление отрицательных последствий рисков событий при общем повышении вероятности успешного завершения проекта.

- **Мониторинг и контроль риска** – мониторинг наступления рисков событий, определение новых рисков, выполнение плана управления рисками проекта и оценка эффективности действий по минимизации рисков.

11.3.1. Основные модели управления рисками программных проектов

Общие методы анализа рисков в сложных системах регламентированы стандартом **ГОСТ Р 51901** — "Управление надежностью. Анализ риска технологических систем". Основной задачей стандарта является обоснование решений, касающихся анализа риска реализации проектов и технологий сложных систем. Хотя стандарт и не направлен исключительно на проекты разработки программного обеспечения, изложенные в нем рекомендации могут быть применены и в таких проектах.

Управление рисками в жизненном цикле программных проектов специально регламентировано международными стандартами **ISO 12207** «Процессы жизненного цикла программных средств» и **ISO 15504** «Оценка и аттестация зрелости процессов создания и сопровождения программных средств и информационных систем», которые целесообразно использовать при разработке комплексов программ. В стандарте ISO 15504 содержится специальный раздел "МАН.4. Процесс управления рисками", назначением которого является регламентирование и планирование процессов выявления и устранения рисков на протяжении всего жизненного цикла программного изделия.

Software Engineering Institute (SEI) разработал **модель управления рисками при разработке программного обеспечения**, частично вошедшую в PMBoK Guide 2000, включающую в себя как требования упомянутых стандартов, так и известные "хорошие практики" (best practices) управления рисками. Подготовку управления рисками проекта эта модель рекомендует проводить в следующей последовательности:

- ⊙ постановка целей управления рисками в соответствии с целями проекта;
- ⊙ планирование и координация работ по оценке рисков проекта
- ⊙ (выделение группы экспертов для оценки рисков, подготовка и планирование интервью с исполнителями с целью выявления рисков в результатах их деятельности);
- ⊙ идентификация и оценка рисков (обнаружение возможных рисков ситуаций и оценивание влияния рисков на проект);
- ⊙ подготовка к устранению рисков (разработка мероприятий по сокращению или устранению рисков и подготовка отчета с

рекомендациями для руководства проекта по анализу и управлению рисками);

- ◎ разработка плана управления рисками, включающего
 - списки рисков по этапам или работам проекта,
 - методы, процессы и инструменты, применяемые для сокращения или устранения рисков,
 - организацию и распределение ответственности за управление рисками, а также за обеспечение допустимого уровня рисков проекта.

Выполняется план управления рисками, как правило, итерационно по основным крупным этапам жизненного цикла проекта. Обычный порядок действий согласно модели SEI при этом следующий.

1. Идентификация рисков:
 - определение возможных источников рисков,
 - идентификация потенциальных рисковых событий,
 - определение ситуаций и условий их возникновения,
 - определение возможного ущерба.
2. Анализ рисков (определение качественных, а лучше – количественных характеристик риска, ранжирование рисков по приоритетам).
3. Планирование ответов на риски – действий по уменьшению или устранению риска.
4. Отслеживание рисков – контроль текущих рисков, корректировка планов по снижению риска, и просмотр результатов выполненных мероприятий плана управления рисками.
5. Подготовка и реализация планов дальнейшего снижения и ликвидации рисков проекта.

11.3.2. Выявление, идентификация и анализ рисков

Прежде, чем с риском можно что-то сделать, он должен быть идентифицирован.

Идентификация риска заключается в фиксации всех факторов беспокойства и озабоченности, связанных с проектом, а затем в постоянном обдумывании всей командой других возможных опасений.

Для обнаружения рисков необходим скептический склад ума. Идентификация риска сродни проведению инспектирования – глобальному поиску дефектов в плане разработки. Для идентификации рисков полезно иметь контрольный список (checklist, шаблон) рисков сходных проектов.

Специалистами по управлению рисками в результате исследований были предложены следующие глобальные категории рисков, характерные для самых простых проектов разработки программных продуктов:

- ⊙ Недостаточная вовлеченность в проект высшего руководства.
- ⊙ Невозможность привлечения реальных и адекватных пользователей.
- ⊙ Непонимание требований разработчиками.
- ⊙ Изменение области применения или целей проекта.
- ⊙ Нехватка знаний или навыков у персонала.

Следует обратить внимание, что в этом списке только 20% рисков относятся к техническим, а 80% относятся к организационным. Такое соотношение отражает реальное положение дел. Риски, возникающие из-за плохой организации проекта, намного больше рисков, возникающих из-за выбора неадекватного инструмента, недостаточного опыта программирования, использования новой информационной технологии и т.п. Таким образом, основная нагрузка по управлению рисками ложится на плечи менеджера проекта.

Каждый идентифицированный риск должен с радостью восприниматься командой проекта, так как в этом случае с ним можно начать что-то делать.

Настоящей проблемой являются риски, которые не удалось идентифицировать.

Такие риски похожи на мины, ждущие свои жертвы. Поскольку процент проектов, которые уже никогда не завершатся, поистине огромен, то только постоянное внимание к рискам делает более вероятным тот факт, что проект попадет в 20% более удачливых проектов или будет прекращен до того, как будут потрачены громадные деньги и загублены карьеры участников.

Менеджер проекта имеет дело с рисками, которые можно классифицировать следующим образом:

- **Известное в известном** – риски известны разработчикам проекта, определены категории риска, а также реалистичные оценки вероятности и ущерба конкретных рисков для данного проекта. Например, если отсутствует достаточно квалифицированный исполнитель для крупного раздела проекта, проявляющийся в этом случае риск относится к известному типу, а относительно его наличия в данном проекте также можно сделать определенные выводы.

- **Известное в неизвестном** – риски известны команде разработчиков проектов, знакома категория риска, но неизвестны реальная вероятность проявления или возможная величина ущерба для данного проекта. Например, отсутствие достаточного взаимодействия с конечными пользователями обычно приводит к риску, связанному с некорректной формулировкой и идентификацией требований. Однако может быть неизвестно, существует ли вообще этот риск и может ли он нанести ущерб для данного проекта.
- **Неизвестное в неизвестном** – риски могут быть известны разработчикам проекта, знакома категория риска, но неизвестны ни вероятность, ни ущерб, а значит, неизвестна его реальная величина для данного проекта разработки программного продукта. Например, подобные риски проявляются в том случае, когда проект использует специфическое технологическое решение, которое указано в требованиях контракта для данного проекта, но незнакомо менеджеру проекта и команде проекта. При отсутствии опыта работы с инструментом, менеджер проекта не может знать всех потенциальных рисков, которые может повлечь за собой его применение.

Идентифицированный риск, величина которого может быть достаточно надежно оценена, естественно назвать **выявленным**.

Выявление риска – это оценка его величины.

Выявление рисков – довольно трудная задача, потому что она неразрывно связана с задачей **анализа** – оценки численного значения величин в условиях неопределенности.

Методы, используемые для оценки величины риска, обычно являются количественными. Однако полный количественный анализ не всегда возможен из-за недостатка информации о системе или деятельности, подвергающейся анализу. При таких обстоятельствах может оказаться эффективным сравнительное количественное или качественное ранжирование риска специалистами, хорошо информированными в данной области и системах. В тех случаях, когда проводится качественное ранжирование, необходимо иметь четкое разъяснение всех используемых терминов и должно быть зафиксировано обоснование всех классификаций вероятностей и ущербов. В том случае, когда проводится полная количественная оценка величины риска, необходимо учитывать, что расчетные значения риска представляют собой приближенные оценки и следует позаботиться о том, чтобы их

точность соответствовала точности используемых исходных данных и аналитических методов.

Элементы процесса оценки величины риска являются общими для всех видов риска. Прежде всего, анализируются возможные причины негативного события с целью определения частоты возникновения таких событий, их продолжительности, а также характера. В процессе анализа может возникнуть необходимость определения оценки вероятности опасности, вызывающей негативные последствия, и проведения анализов последовательности обуславливающих событий.

Анализ вероятности используется для оценки вероятности каждого негативного события, проявившегося на стадии идентификации опасностей, угроз.

Для оценки частот происходящих событий обычно применяются следующие три метода:

- ✓ использование имеющихся статистических данных (предысторий);
- ✓ получение частот негативных событий на основе аналитических или имитационных методов;
- ✓ использование мнений экспертов.

Данные эксплуатации используются с целью определения частоты, с которой негативные события происходили в прошлом, и, исходя из этого, прогнозирование частоты, с которой они могут произойти в будущем.

Анализ ущерба используется для оценки вероятного воздействия, которое вызывается нежелательным событием. Анализ ущерба должен:

- основываться на выбранных негативных событиях;
- описывать и оценивать любые последствия, являющиеся результатом негативных событий;
- учитывать существующие меры, направленные на смягчение последствий, наряду со всеми соответствующими условиями, оказывающими влияние на последствия;
- устанавливать критерии, используемые для полной идентификации последствий;
- рассматривать и учитывать как немедленные последствия, так и те, которые могут проявиться по прошествии определенного периода времени, если это не противоречит сфере распространения исследований;
- рассматривать и учитывать вторичные последствия, распространяющиеся на смежные компоненты и системы.

Существует **множество неопределенностей**, связанных с оценкой риска. Понимание неопределенностей и вызывающих их причин необходимо для эффективной интерпретации значений величины риска. Анализ неопределенностей должен предусматривать определение изменений и неточностей в результатах моделирования, которые являются следствием отклонения параметров и предположений, применяемых при построении модели. Областью, тесно связанной с анализом неопределенностей, является **анализ чувствительности**. Анализ подразумевает определение изменений в реакции модели на отклонения отдельных параметров модели. Оценка неопределенности состоит из преобразования неопределенности критических параметров модели в неопределенность результатов в соответствии с моделью риска. Это относится как к неопределенностям данных, так и к неопределенностям модели. Должны быть определены те параметры, к которым чувствителен анализ.

В случае, когда обоснованную оценку вероятности рисков получить трудно (что является скорее правилом, чем исключением) весьма полезным является выполнение требования стандарта ISO 15504, согласно которому для каждого вида риска (или набора рисков) должны быть определены метрики, отражающие изменения в состоянии (вероятности, ущербе, временном диапазоне проявления) рисков в зависимости от деятельности по их устранению.

Спектр методик количественного анализа неопределенностей и рисков проекта весьма широк: от PERT анализа и анализа "что-если" (what-if analysis) до сложного имитационного моделирования методом Монте-Карло. Для этих методов существует и **специальное программное обеспечение**. Так, ***Open Plan Professional*** позволяет охарактеризовать неопределенности, связанные с длительностью проекта. Запрашивая вероятностную, а не строго определенную, длительность отдельных работ, Open Plan моделирует вклад факторов неопределенностей в вычисление даты окончания проекта. Анализ рисков в Open Plan реализуется следующими средствами:

- Процедуры ввода вероятностного распределения длительности для избранных или всех работ проекта.
- Выполнение анализа рисков по методу Монте-Карло для определения вклада вероятностей в даты проекта.
- Предоставление представлений и отчетов, которые используются для анализа влияния неопределенностей на реализацию проекта) см., например, рис. 2, на котором представлена гистограмма распределения даты возможного окончания проекта, построенная при помощи Open Plan Professional).

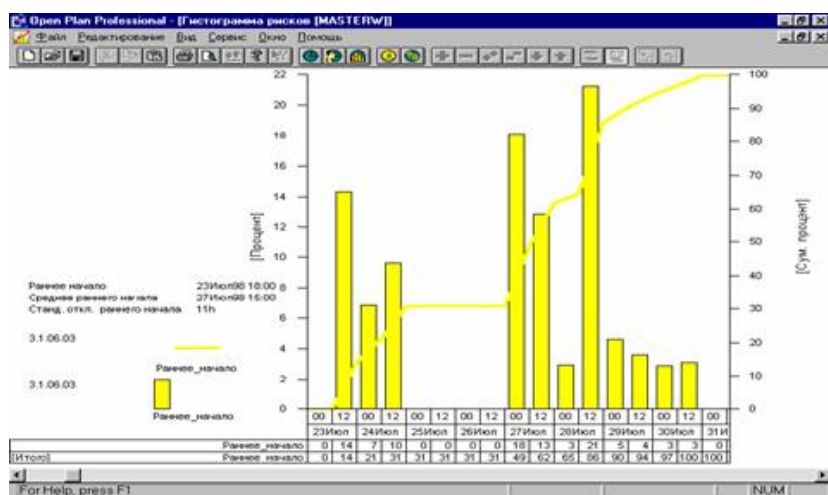


Рис. 2. Гистограмма распределения даты окончания проекта (Open Plan Professional)

Среди количественных методов оценки неопределенностей и их влияния на проекты весьма популярен метод PERT-анализа (Project Evaluation and Review Technique). Его суть заключается в том, что для каждой работы проекта указываются три оценки его длительности (трудоемкости, стоимости) — оптимистическая (О), наиболее вероятная (В) и пессимистическая (П). Полученные оценки осредняются с заданными весами, давая возможность анализировать построенную таким образом сетевую диаграмму, содержащую ожидаемые длительности (трудоемкости, стоимости). Стандартная формула осреднения $(O + 4B + П)/6$, однако при необходимости веса можно менять. Такой анализ позволяет автоматизировать, например, MS Project Professional.

Метод PERT был разработан сотрудниками Военно-морского флота США в 1957 году для обеспечения создания ракеты «Поларис». Применяя PERT-анализ, они попытались симитировать график выполнения работ по созданию ракеты путем построения логической сети взаимозависимых последовательных событий. На начальной стадии PERT-представление было сфокусировано на контроле временных характеристик графика и прогнозировании вероятности успешного завершения программы. Но прежде чем PERT-представление было окончательно принято руководителями программ в промышленности, Военно-воздушные силы США внесли дополнение в методику, добавив к нему функцию оценки ресурсов. Таким образом, в 1962 году появилась PERT/Cost-методика (PERT-анализ с целью стоимостного прогнозирования), в то время как первоначально PERT-анализ был известен под названием PERT/Time (PERT-анализ для определения времени реализации проекта).

Проверка результатов анализа должна осуществляться экспертами, не привлеченными к участию в выполнении проекта. Проверка должна включать в себя следующие этапы:

- проверка соответствия области применения анализа поставленным задачам;
- проверка всех важных допущений при анализе для обеспечения уверенности в том, что они являются правдоподобными в условиях имеющейся информации;
- подтверждение аналитиком правильности использованных методов, моделей и данных;
- проверка результатов анализа на повторяемость с привлечением персонала, не участвующего в выполнении анализа;
- проверка результатов анализа на устойчивость по отношению к различным форматам данных.

Одним из самых простых методов суммировать результаты анализа является ранжирование рисков по приоритетам, поскольку средств и времени для устранения всех рисков как правило, не хватает. Если есть оценки вероятности риска p , ущерба d и стоимости устранения c , то приоритет q риска можно определять по формуле

$$q = (1 - p)(1 - d)c.$$

11.3.3. Планирование ответов на риски. Методы снижения риска

Методы снижения риска – это методы, направленные на уменьшение величины риска, то есть либо методы снижения вероятности возникновения нежелательного события, либо методы уменьшения величины ожидаемого ущерба, либо то и другое вместе.

На практике редко возникает столь благоприятная ситуация, чтобы можно было полностью устранить риск, то есть снизить величину риска до нуля. Как правило, проводятся только мероприятия, снижающие риски до приемлемой величины, но не устраняющие риск полностью. При этом нужно учитывать стоимость мероприятий по уменьшению или устранению риска. Обычно руководствуются следующим простым правилом.

Пусть оценки вероятности и ущерба риска без проведения мероприятия по снижению риска равны p_1 и d_1 , с проведением мероприятия p_2 и d_2 , соответственно, а стоимость мероприятия по уменьшению риска c . Тогда такое мероприятие имеет смысл, если $p_1 d_1 \geq p_2 d_2 + c$.

Рассмотрим пример. Пусть вероятность рисковогo события «недостаточные навыки программирования» равна 10%, ущерб — 50%, стоимость мероприятия по сокращения риска «предварительное обучение» — 1%, а вероятность и ущерб в случае обучения составляют 5% и 20%, соответственно. Тогда $10\% \times 50\% = 5\%$, $5\% \times 20\% + 1\% = 1\% + 1\% = 2\%$ и обучение имеет смысл.

Мероприятия, направленные на снижение риска, могут быть самыми разными. Они могут устранять первичные причины, вызвавшие появление рисковогo ситуаций, или уменьшать вероятность и/или ущерб от рисковогo событий. Приведем несколько примеров.

- Риск появления новых требований в процессе работы над проектом можно уменьшить согласованием подробного перечня требований с заказчиком, включения этого списка в договор и точное следование этим требованиям.
- Риск существенного изменения рыночной ситуации, делающей выполнение исходного плана бессмысленным, снижается предварительным исследованием рынка, экспертной оценкой и/или консультацией у опытного стороннего консультанта.
- Риск недостаточных навыков владения исполнителями инструментами разработки можно уменьшить специальным тренингом.

Естественно, список можно продолжать.

Не следует пренебрегать одним важным способом минимизации рисков - заключением договора страхования, полностью или частично покрывающего возможный ущерб, со специализированной страховой компанией. Таким образом можно уменьшать риски, являющиеся в определенном смысле "внешними" по отношению к компании, выполняющей проект (например, риски срыва поставок или платежей, риски изменения рыночной ситуации и т.п.)

При выполнении проектов менеджеры постоянно накапливают опыт по управлению рисками, и при наличии такого опыта может создать для себя "шаблоны" планов ответа на риски — списки возможных рисков и действий по их уменьшению.

11.3.4. План управления рисками

Детальный план управления рисками проекта представляет собой таблицу с указанием:

1. выявленных рисков для каждой работы (этапа) проекта или для проекта в целом
2. категории риска (риски, связанные с управлением требованиями, риски, связанные с бюджетом, риски, связанные с инструментарием и т.п.),

3. вероятности и оцениваемого ущерба (либо заранее выбранных метрик, характеризующих изменения этих величин), и, наконец,
4. планируемых мероприятий по их уменьшению с указанием ответственных лиц.

Риски в плане, как правило, приводятся в порядке убывания приоритета.

Следует отметить, что управление рисками является непрерывным процессом в рамках жизненного цикла проекта. Отслеживание и контроль рисков, включенных в список, должны выполняться на регулярной основе.

Для каждой риска или категории рисков следует назначить ответственных за мероприятия по снижению рисков. Также следует установить формат отчета об управлении рисками для каждой категории рисков. Эти отчеты обычно рассматриваются регулярно (например, на регулярных, скажем, еженедельных, совещаниях группы проекта, либо на регулярных совещаниях менеджера и руководителей команд и т.п.) Многие менеджеры любят регулярно отслеживать сводный отчет о состоянии, скажем, десяти наиболее приоритетных рисков.

11.3.5. Мониторинг рисков

Очень важно отметить, что все величины, связанные с риском, не являются постоянными в ходе проекта. Вероятность рискового события и возможный ущерб могут увеличиваться и уменьшаться. В связи с этим необходимо постоянно

- оценивать результаты выполненных мероприятий плана управления рисками,
- отслеживать риски, существующие в проекте на текущий момент,
- корректировать планы по снижению риска в соответствии с текущей ситуацией.

Пусть, например, на начало проекта в плане управления риском был отмечен риск недостаточной квалификации команды разработчиков и предложены соответствующие мероприятия по его снижению. Однако по окончании фазы разработки (и даже раньше, когда разработка уже идет полным ходом), этот риск уже не является актуальным – он либо уже полностью устранен (т.е. подобрана достаточно квалифицированная команда), либо превратился в постоянно действующий фактор (т.е. команда действительно не имеет достаточной квалификации, но менеджер об этом уже точно знает).

11.4. Заключение

Исход любого проекта зависит от большого числа факторов неопределенности. Поэтому управление рисками должно быть одной из основ управления проектами. Отличительные черты управления рисками в проектах по разработке программного обеспечения состоят в основном в большом количестве рисков, вероятностная оценка которых весьма затруднительна. В остальном же управление рисками в таких проектах в основном следует общим принципам управления рисками.

11.5. «Карта памяти» по теме



11.6. Использованная литература

1. Р. Т. Фатрелл, Д. Ф. Шафер, Л. И. Шафер. *Управление программными проектами: достижение оптимального качества при минимальных затратах*. Пер. с англ. Вильямс, 2003.
2. Липаев В.В. Анализ и сокращение рисков проектов программных средств. <http://www.jetinfo.ru/2005/1/1/article1.1.2005.html>
3. B.W. Boehm. *Software risk management*. IEEE Computer Society Press. Washington, 1989.
4. R. Charett. *Software engineering risk analysis and management*. N.Y.: McGraw — Hill, 1989.
5. I. Sommerville. *Software engineering*. Lancaster University. Pearson Education Limited, 2001.

Тема 12. Особенности аутсорсинга

12.1. Введение

Аутсорсинг является одной из наиболее современных и успешных бизнес-моделей, позволяющих добиться реальных конкурентных преимуществ. Современный аутсорсинг в области информационных технологий – достаточно распространенное явление, имеющее ряд особенностей, которые будут рассмотрены в этой теме.

Изучив учебный материал данной темы, Вы:

- узнаете или пополните свои знания о том, что такое аутсорсинг и аутстаффинг;
- узнаете о том, для чего следует их использовать;
- получите общее представление об общих схемах аутсорсинга программных проектов и их особенностях.

В рамках темы рассматриваются следующие учебные вопросы:

- аутсорсинг;
- аутстаффинг;
- аутсорсинг информационных технологий.

12.2. Аутсорсинг

Широкое распространение аутсорсинга (outsourcing) в мире современного бизнеса – это относительно новое явление, хотя сам по себе аутсорсинг и не является изобретением последнего времени.

Дело в том, что в современном мире развитой рыночной экономики компании сталкиваются с беспрецедентным давлением со стороны рынка. Выживают и добиваются успеха только те, кто ведет свой бизнес наиболее эффективным способом, добиваясь снижения операционных расходов при сохранении высокого качества товаров и услуг. Для увеличения эффективности деятельности, компании вынуждены все более специализироваться, сосредотачивая максимальные ресурсы лишь на тех видах деятельности, в которых они наиболее успешны. Аутсорсинг является одной из наиболее современных и успешных бизнес-моделей, позволяющих добиться реальных конкурентных преимуществ как раз за счет перенесения всех не основных для компании видов деятельности на плечи сторонних организаций, что дает возможность полностью сосредоточиться лишь на основных видах деятельности, в которых компания имеет серьезные преимущества перед конкурентами.

12.2.1. Что такое аутсорсинг?

При переводе на русский язык «в лоб» outsourcing звучит как «заимствование ресурсов извне». Другими же словами, аутсорсинг – это выполнение сторонней организацией определенных задач или некоторых бизнес-процессов, обычно не являющихся профильным для бизнеса компании, но, тем не менее, необходимых для полноценного функционирования бизнеса. При использовании аутсорсинга компания может сосредоточить все свои ресурсы на том виде деятельности, который считает основным, и передать остальные (поддерживающие, сопутствующие) функции надежному и профессиональному партнеру.

Аутсорсинг – это выполнение сторонней организацией определенных задач или некоторых бизнес-процессов, не являющихся профильными для бизнеса компании, но необходимых для полноценного функционирования бизнеса.

Наиболее часто аутсорсингу подлежат:

- ✓ информационные технологии, включая разработку, интеграцию, внедрение или поддержку программного обеспечения,
- ✓ ведение бухгалтерии,
- ✓ расчет заработной платы,
- ✓ набор и обучение персонала,
- ✓ административная поддержка,
- ✓ маркетинговые и PR услуги,
- ✓ обслуживание корпоративной собственности,
- ✓ организация питания и уборки офиса,
- ✓ служба безопасности.

Частично аутсорсинг используется с весьма давних времен.

Например, уже давно было принято не держать в каждой компании специальную службу охраны, а пользоваться услугами специальных охранных фирм. Однако тенденцией последнего времени, особенно в странах (и секторах экономики) с развитой рыночной конкуренцией, является чрезвычайно широкое использование аутсорсинга. В настоящее время не редкость, когда компании для всех упомянутых выше вспомогательных видов деятельности не держат специальных сотрудников, а пользуются услугами специальных фирм.

Например,

- ⊙ для организации регулярной уборки помещения заключается договор с клининговой фирмой (от англ. to clean – чистить) вместо того, чтобы держать штат уборщиц;

- ⊙ для организации набора персонала заключается договор с рекрутинговым агентством вместо содержания собственной кадровой службы;
- ⊙ для организации регулярных курсов повышения квалификации сотрудников заключается договор с образовательной организацией (учебным центром), специализирующейся именно на учебной деятельности, вместо содержания собственных преподавателей и организаторов учебного процесса.

В последнее время популярность приобретает и аутсорсинг менеджмента проектов. При этом именно для функции профессионального управления проектом нанимается специальная аутсорсинговая компания, специализирующаяся на этом виде деятельности.

12.2.2. Плюсы и минусы аутсорсинга

Аутсорсинг, безусловно, обладает рядом преимуществ по сравнению с классической бизнес-моделью, когда компания, наряду с основным бизнесом, занимается еще и всеми обеспечивающими функциями.

На стратегическом уровне аутсорсинг позволяет:

- + сфокусировать внимание на основном бизнесе;
- + уменьшить необходимость инвестиций в неосновные фонды;
- + гибко реагировать на изменения на рынке (кризисы, дефолты) и внутри компании (реорганизации, реструктуризации, слияния, поглощения) и др.

Тактические преимущества аутсорсинга следующие:

- + отсутствие необходимости в расширении штата компании;
- + сокращение косвенных (накладных) расходов. Важно подчеркнуть, что речь о снижении только косвенных расходов, а не всех вообще расходов: в ряде случаев аутсорсинг может вести и к росту затрат. Однако даже и в последнем случае многие компании готовы пойти на возможный рост затрат в связи с переходом на аутсорсинг в основном ради того, чтобы получить преимущества от сосредоточения усилий на своей основной деятельности;
- + гарантированное качество услуг, предоставляемых профессиональной аутсорсинговой компанией.

Однако аутсорсинг имеет и ряд минусов (о чем в последнее время говорят реже):

- это может быть довольно дорогим удовольствием. Например, услуги профессиональной клининговой компании могут обойтись существенно дороже заработной платы уборщицы;
- аутсорсингом нужно управлять, что не тривиально. Управление аутсорсингом может иметь весьма специфические сложности, в особенности, если речь идет о разработке программного обеспечения;
- наконец, аутсорсинг, в особенности в области разработки программного обеспечения, несет в себе ряд дополнительных рисков.

В каких случаях использование аутсорсинга наиболее эффективно? Вот основные из таких ситуаций:

- ✓ Компания принимает стратегическое решение сосредоточить ресурсы на основном виде деятельности.
- ✓ Компания переживает период внутренних перемен (реструктуризация, реорганизация, процесс слияния, поглощения). Передача в этот момент части внутренних функций является наиболее безболезненным путем решения задач, стоящих перед организацией.
- ✓ Бизнес растет и нагрузки на отдельные функции увеличиваются пропорционально с ним. Используя аутсорсинг, отпадает необходимость нанимать дополнительный персонал и оборудовать рабочие места – аутсорсинговый партнер выполнит любой объем работы точно и в срок.
- ✓ Крупные компании часто сталкиваются с тем, что стоимость неосновных бизнес-процессов становится непропорционально высокой. Использование аутсорсинга в этом случае может дать ощутимую экономическую выгоду.
- ✓ Компания испытывает недовольство тем уровнем сервиса, который оказывают внутренние службы.

12.3. Аутстаффинг

Аутсорсинг – не единственная бизнес-стратегия, дающая компании возможность сосредоточиться на тех видах деятельности, в которых она видит свои основные конкурентные преимущества. В последнее время большую популярность приобретает и другая

стратегия, направленная на реализацию той же цели — *аутстаффинг (outstaffing)*.

Аутстаффинг — это вывод персонала за штат компании и трудоустройство его в специальное аутстаффинговое агентство.

Иногда компаниям экономически выгоднее вывести персонал, подлежащий сокращению, или временный персонал, набранный для выполнения конкретного проекта, за штат, а не проводить его через свой отдел кадров.

Чаще всего этой услугой пользуются крупные международные корпорации, в которых управлению человеческими ресурсами уделяется огромное внимание. Участвуя в программе аутстаффинга, эти компании сокращают расходы по содержанию персонала — им не надо думать о разработке компенсационных пакетов, заботиться о планах развития карьеры, тренингах, решать спорные трудовые вопросы — все это осуществляет аутстаффинговое агентство. Немаловажно также, что в случае необходимости провести сокращения компания не подрывает свой имидж и стоимость своих акций, потому что формально никаких сокращений не происходит.

Итак, *фактически сотрудники работают в компании, однако, формально они числятся работниками агентства, оказывающего эту услугу*. В агентстве лежат их трудовые книжки, ведутся личные дела, выплачивается зарплата и все связанные с ней налоги и выплаты, а если необходимо, осуществляются процедуры по завершению трудового договора. Компания ежемесячно выплачивает агентству фиксированную сумму, включающую в себя комиссию за аутстаффинг и стоимость расходов по содержанию персонала в штате агентства.

Аутстаффинг эффективен, если:

- + в компании существует жесткое ограничение по численности персонала или по фонду оплаты труда;
- + необходимо привлечение персонала для выполнения временных работ или проектов (выполнение дополнительного объема работ, сезонные работы), в т.ч. необходимо снизить затраты, вызванные простоями;
- + для проверки кандидата необходим более длительный, чем установлен действующим законодательством, испытательный срок.

Аутстаффинг позволяет:

- + сократить административные издержки,
- + решить проблемы, связанные с ограничением штата,

- + оптимизировать отношения с персоналом, задействованным в отдельных проектах, в том числе выплачивать бонусы и вознаграждения,
- + ввести дополнительный нематериальный стимул для работников – быть переведенными в штат,
- + увеличить мобильность в вопросах привлечения, сокращения персонала,
- + освободить собственные кадровые службы и бухгалтерию от части функций в области ведения бухгалтерского и налогового учета, а также кадрового делопроизводства,
- + сохранить или получить статус малого предприятия, возможность работать по упрощенной системе налогообложения,
- + снизить финансовую нагрузку и риски и увеличение прибыли в расчете на одного сотрудника,
- + исключить риски трудовых споров (эти риски несет аутстаффинговое агентство).

Очевидным результатом этого является повышение эффективности компаний, снижение расходов и усиление позиций на рынке.

В связи с развитием информационных технологий, в особенности Интернета, в последнее время аутстаффинг иногда принимает форму *«виртуального офиса»*. В данной разновидности ведения бизнеса персонал, предоставляемый компании аутстаффинговым агентством, фактически работает в компании, но физически не присутствует в офисе компании. Физически персонал работает в помещениях аутстаффингового агентства, общаясь с компанией-работодателем по Интернету. Такой режим работы дает еще больше выгод компании, пользующейся аутстаффингом, поскольку отпадает необходимость в содержании дополнительных производственных площадей и оборудования. Не все виды бизнеса позволяют использовать технологию «виртуального офиса», но проекты по разработке программного обеспечения позволяют, и в этой области технология «виртуального офиса» распространяется все шире.

12.4. Аутсорсинг информационных технологий

Аутсорсинг информационных технологий (IT outsourcing) – это вид услуг, подразумевающий частичную или полную передачу заказчиком своих функций по разработке, интеграции и/или поддержке программных систем или компьютерного оборудования систем в специализированную компанию. При этом исполнитель гарантирует выполнение обозначенных в договоре функций в соответствии с утвержденным уровнем сервиса.

Под понятие аутсорсинга информационных технологий подпадают работы (как одноразовые, так и долгосрочные) в области использования информационных технологий: разработка или поддержка программного обеспечения, обслуживание компьютерного или сетевого оборудования, создание и ведение информационных сайтов в Интернете, консультации в этой области и т.п. В дальнейшем нас будут интересовать аутсорсинг проектов, связанных с программным обеспечением.

12.4.1. Аутсорсинг программных проектов

Аутсорсинг проектов в области разработки, интеграции или поддержки программного обеспечения имеет ряд особенностей.

Прежде чем их перечислять, отметим, что в силу традиции последнего времени под **аутсорсингом программных проектов** понимают не обычные проекты по разработке программного обеспечения по заказу фирмы, для которой этот вид деятельности не является основным (что в принципе соответствует общему определению аутсорсинга), а проекты по разработке программного обеспечения, в которых заказчиком является как раз компания, профессионально занимающаяся программным обеспечением. При этом в подавляющем большинстве случаев такой аутсорсинг связан исключительно с минимизацией затрат.

Классический пример, происходящий в последние годы повсеместно в странах Западной Европы и США, это вывод заказов на разработку программного обеспечения в аутсорсинговые компании-разработчики, расположенные на территориях стран с относительно низким средним уровнем заработных плат (Индия, Россия, Украина, Вьетнам и т.д.) Основная идея здесь – исключительно сокращение расходов на разработку: стоимость труда программиста в Бангалоре (Индия) очевидно дешевле, чем программиста в Силиконовой долине в США. Этот феномен получил название **«оффшорного программирования»**, поэтому аутсорсинг в области разработки программного обеспечения почти всегда воспринимается как синоним оффшорного программирования (хотя это и неправильно).

Оффшорное программирование характерно в последнее время и для внутрироссийских проектов. Так, в связи с ростом заработных плат в Москве заказы на разработку программного обеспечения все чаще передаются московскими компаниями-разработчиками полностью или частично (для создания отдельных компонентов программных систем) их коллегам из других городов или стран, где заработные платы исполнителей меньше.

Следует отметить, что оффшорное программирование – весьма неустойчивое явление, так что компания, ориентирующаяся исключительно на этот вид деятельности, подвержена существенным

рискам, связанным с нестабильностью рыночной ситуации. Дело в том, что:

- ✓ заработные платы исполнителей имеют тенденцию к росту, что связано как с улучшением общеэкономической ситуации, так и, главное, с тем, что с ростом числа аутсорсинговых заказов весьма быстро исчерпывается резерв квалифицированных исполнителей – специалистов в области информационных технологий, а значит, для привлечения специалистов хорошего уровня в компанию приходится тратить все больше и больше. Так что издержки растут, причем, как показывает опыт, именно в оффшорном программировании и в России – весьма быстро;
- ✓ заказчики аутсорсинговых проектов по разработке программного обеспечения достаточно легко переносят свои заказы туда, где издержки ниже. При этом надо учесть, что значительное число «классических» проектов в области оффшорного программирования не требуют владения специальными сложными технологиями, обладанием специфическим know-how и т.п., так что соответствующие проекты можно достаточно легко передоверить большому числу компаний.

В результате длительное поддержание конкурентоспособности в области оффшорного программирования в условиях неизбежного роста издержек является весьма трудной для компании задачей.

12.4.2. Виды аутсорсинга программных проектов

Имеет смысл провести следующую достаточно простую классификацию аутсорсинговых программных проектов.

1. Заказы на полную разработку программ.

Этот вид аутсорсинга принципиально не отличается от обычной разработки программного обеспечения на заказ. По сути дела единственное, но немаловажное отличие, в том, что «настоящий» заказчик, который в конечном итоге и определяет требования к разрабатываемому программному обеспечению, скрыт от конечного исполнителя. Последний имеет договорные отношения с компанией, передавшей ему аутсорсинговый заказ. Таким образом, схема взаимоотношений оказывается «трехзвенной»:

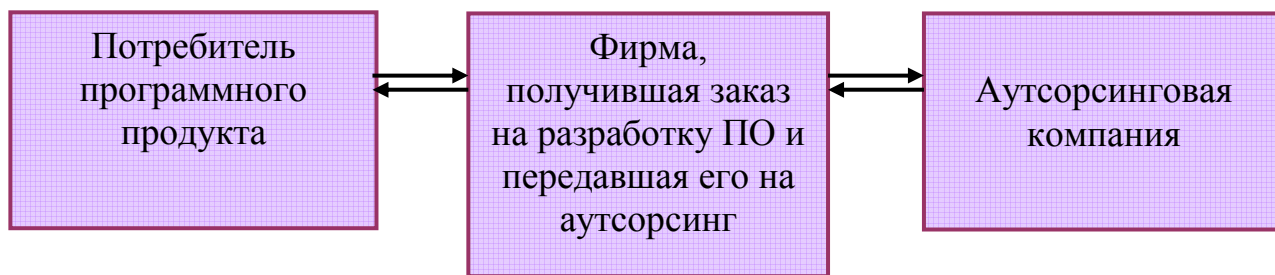


Рис. 1. Взаимоотношения между участниками аутсорсингового проекта

Фирма, изначально получившая заказ на разработку и передавшая его на аутсорсинг, фактически играет здесь роль посредника. Но именно на нее ложится бремя сбора и формализации требований к разрабатываемому продукту и передачи их в виде уже готового технического задания аутсорсинговой компании, а также грамотное управление аутсорсинговым проектом.

Достоинством такой схемы для аутсорсинговой компании является наличие готового технического задания. Однако риски непонимания истинных требований потребителя продукта здесь намного выше, чем при классической работе без посредника, а это значит, что увеличивается вероятность разработки не удовлетворяющего потребителя продукта, что ведет к конфликтам между участниками проекта, повышает риски неплатежей и превышения длительности проекта. Как следствие, для того чтобы избежать таких неприятностей, необходимо уделить особое внимание менеджменту проекта (увеличение квалификации исполнителей мало влияет на подобные риски), в особенности со стороны фирмы-«посредника». Это особенно важно, когда аутсорсинговая компания работает удаленно, а связь осуществляется в основном по телефону или через Интернет.

2. Заказы на разработку отдельных компонентов программного комплекса.

На сегодняшний день это наиболее распространенный вид аутсорсинговых заказов на разработку сложного программного обеспечения.

Схема взаимодействия «потребитель продукта – компания-разработчик, передающая аутсорсинговые заказы – аутсорсинговые компании» не отличается от изложенной выше, так что все вышеизложенное относится и к описываемой ситуации. Новое здесь то, что отдельные модули или компоненты программного комплекса создаются несколькими компаниями. Для аутсорсинговых компаний сложность проекта резко повышается за счет необходимости координации своих действий со смежниками.

Для фирмы, стоящей в середине цепочки (она отвечает перед непосредственным потребителем за конечный продукт) также резко возрастает сложность, а значит, и стоимость управления проектом.

Дополнительно к функциям взаимодействия с конечным потребителем-заказчиком, сбора требований, формулировки технических заданий и управления проектом (в данном случае является управление субподрядными аутсорсинговыми проектами), ей приходится также брать на себя функции интеграции и тестирования программного комплекса в целом.

Для всех участников проекта возрастают риски, связанные с взаимодействием со смежниками, а также риски разработки не удовлетворяющего потребителя продукта, что опять-таки повышает риски неплатежей и превышения плановой длительности проекта. Для того чтобы избежать провала проекта (который в этой ситуации гораздо более вероятен, чем в ситуации простой работы на заказ без посредника, или даже чем в ситуации аутсорсингового заказа на полную разработку), кроме высокопрофессионального и иногда даже виртуозного менеджмента проектов компании-интегратору необходимо иметь в своем составе профессиональную команду системных архитекторов, интеграторов и тестеров. Такие проекты обычно весьма сложны в управлении, прежде всего, в силу необходимости координации действий поставщиков компонентов системы, внешних и внутренних консультантов и заказчика-потребителя продукта. И все-таки на эту сложность идут сознательно ради снижения затрат на проект в целом (если содержать команду исполнителей дороже, чем передать заказ на аутсорсинг).

3. Заказы на поддержку существующего программного обеспечения.

Мечта многих компаний-разработчиков, так как дает возможность длительного безбедного существования. Причем в отличие от «оффшорного программирования» это весьма устойчивый бизнес. Это типичный аутсорсинг услуги (ничем принципиально не отличающийся, скажем, от аутсорсинга ведения бухгалтерии), и выгоден он всем сторонам.

К сожалению, аутсорсинг поддержки программного обеспечения пока еще не повсеместное явление, хотя и встречается довольно часто. Причина этого, возможно, в том, что многие компании-владельцы программного обеспечения боятся попасть в зависимость от сторонних организаций в части поддержки и предпочитают нанимать для этого собственных сотрудников.

12.5. Заключение

Аутсорсинг – это выполнение сторонней организацией определенных задач или некоторых бизнес-процессов, обычно не являющихся профильным для бизнеса компании, но, тем не менее, необходимых для полноценного функционирования бизнеса.

Аутстаффинг – это вывод персонала за штат компании и трудоустройство его в специальное аутстаффинговое агентство.

Обе эти бизнес-стратегии все чаще применяются в последнее время с целью позволить компаниям сосредоточиться на тех видах деятельности, в которых они имеют конкурентные преимущества. Аутсорсинг программных проектов, как правило, вызван необходимостью снижения затрат на разработку программного обеспечения. Реализация аутсорсинговых программных проектов связана с рядом специфических трудностей, а именно, с необходимостью координации действий заказчика-потребителя разрабатываемого продукта и аутсорсинговых компаний-субподрядчиков, разрабатывающих отдельные его компоненты, когда «субподрядчики» как правило не имеют непосредственного контакта с заказчиком. Следует при этом уделять особое внимание менеджменту проекта и интеграции и тестированию системы в целом.

12.6. Список использованной и рекомендованной литературы

1. Б. А. Аникин, И. Л. Рудая. Аутсорсинг и аутстаффинг: высокие технологии менеджмента. Серия: Высшее образование. Издательство: Инфра-М, 2005 г.
2. Роб Аалдерс. ИТ аутсорсинг. Практическое руководство. Серия: Библиотека IBS. Издательство: Альпина Бизнес Букс, 2004.
3. Элизабет Спарроу. Успешный ИТ-аутсорсинг. Издательство: КУДИЦ-Образ, 2004.
4. У. Ройс. Управление проектами по созданию программного обеспечения, Издательство: ЛОРИ, 2002
5. Филлипс Д. Менеджмент ИТ-проектов, Издательство: ЛОРИ, 2005.

12.7. «Карта памяти» по теме

