

УПРАВЛЕНИЕ ПРОЕКТАМИ РАЗРАБОТКИ ПО

Основные этапы разработки ПО

1. Анализ и описание требований к ПО
2. **Планирование выполнения проекта разработки ПО**
3. Разработка архитектуры ПО (общей структуры).
4. Укрупненное проектирование ПО.
5. Детальное проектирование ПО.
6. Кодирование и тестирование единиц кода.
7. Системное тестирование.
8. Внедрение разработанного ПО у заказчика.

- **Проектом** называется совокупность распределенных во времени мероприятий или работ, направленных на достижение поставленной цели.
- Отличия процесса разработки ПО от процессов реализации технических проектов:
- Программный продукт нематериален
- Не существует стандартных процессов разработки ПО
- Большие программные проекты - это часто "одноразовые" проекты

Свойства Проекта:

Проект всегда

- имеет четко определенные
 - Цель, начало и конец
- исполняется командой
- использует материальные ресурсы
- имеет бюджет
- имеет ограничения трех видов:
 - Ограничения по бюджету
 - Ограничения по времени
 - Ограничения по ресурсам

- Цель выражается в получении некоторого результата. Достижение этого результата означает успешное завершение и окончание проекта
- Начало проекта совпадает с началом первой работы, направленной на достижение поставленной цели. Начало может задаваться директивно, либо рассчитываться в результате составления плана работ по проекту.
- Конец проекта совпадает с концом последней работы, направленной на получение заданного результата.
- Проект исполняется командой, в состав которой входит руководитель проекта, менеджеры, исполнители. Помимо основной команды в нем могут участвовать сторонние исполнители, команды и организации, которые привлекаются на временной основе для выполнения отдельных работ.
- При реализации проекта используются материальные ресурсы. Их номенклатура и количество определяются характером проекта и входящих в него работ.
- Проект имеет бюджет. Стоимость проекта складывается из:
 - стоимости израсходованных материальных ресурсов,
 - затрат по оплате труда реализующей его команды и
 - прочих расходов, связанных с особенностями конкретных видов работ.

Управление проектом

- Управление проектом (Project Management - PM) – это наука и искусство руководства и координации людских и материальных ресурсов на протяжении жизненного цикла проекта путем применения современных методов и техники управления для достижения определенных в проекте результатов по составу и объему работ, стоимости, времени, качеству и удовлетворению участников проекта (Свод знаний по управлению проектами (PMBOK))

- Ключевой категорией, участвующей в процессе управления проектами, являются
- ограничения. Известный закон Лермана гласит: "Любую техническую проблему можно преодолеть, имея достаточно времени и денег", а следствие Лермана уточняет: "Вам никогда не будет хватать либо времени, либо денег".



Качество, время и деньги интерпретируются площадями внутренних треугольников. В этом треугольнике центр и верхняя вершины фиксированы, а нижние вершины могут перемещаться. Треугольник иллюстрирует, что любое сокращение финансов или времени ведет к сокращению качества, а увеличение качества может быть достигнуто за счет увеличения финансирования или сроков.

Работы по управлению

проектом

- Управление проектом включает следующие виды работ:
 1. планирование проекта;
 2. мониторинг хода выполнения проекта;
 3. управление рисками;
 4. анализ завершения проекта;

Планирование выполнения проекта разработки ПО

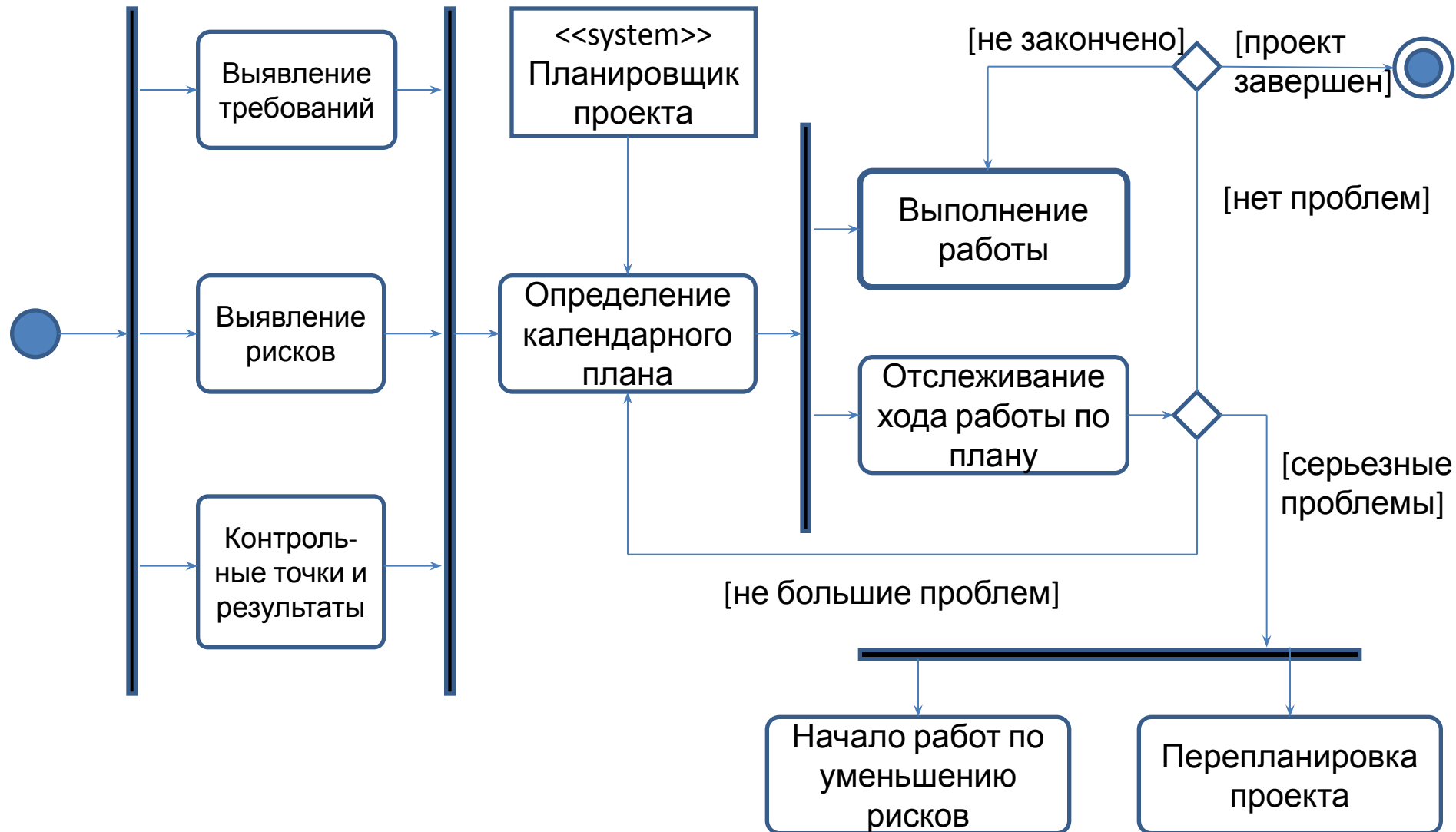
- Планирование работ это наиболее важная часть работ по управлению проектом.
- Основные цели:
 - определить обоснованные цели проекта:
 - затраты (деньги);
 - календарный план;
 - уровень качества.
- Проект будет считаться успешным, если он достигнет поставленные цели по деньгам, срокам и качеству.

- Без детального планирования невозможно отслеживать и оперативно управлять ходом выполнения проекта.
- Никакое количество технических усилий, приложенных позже, не сможет компенсировать недостаток тщательного планирования.
- Недостаток планирования большого проекта по разработке ПО практически является гарантией его неудачного выполнения.

Входные данные планирования

- Входными данными для планирования проекта разработки ПО являются:
 - спецификация требований к ПО (СТПО);
 - предполагаемая архитектура разрабатываемого ПО.
- Для планирования не требуются детальные требования к ПО, но должны быть уже известны все важные требования.
- Очень желательным является наличие ключевых архитектурных решений.

Процесс планирования проекта



Задачи решаемые в ходе планирования проекта

1. Оценка трудоемкости проекта.
2. Планирование сроков выполнения проекта и состава исполнителей.
3. Управление качеством.
4. Управление рисками.
5. Планирование мониторинга выполнения проекта.

Оценка трудоемкости

- Для планирования проекта разработки ПО важным обязательным условием является
 - общая оценка трудоемкости (объема работ) и
 - общая оценка сроков выполнения работ.
- Такие оценки требуются до того, как процесс разработки будет запущен, т.к. они определяют цели проекта по стоимости и по срокам.
- Без таких оценок нельзя ответить на такие вопросы, как:
 - «опаздывает ли выполнение проекта?»
 - «превышены ли затраты на выполнение проекта?».
- Более практическим использованием таких оценок является обсуждение с заказчиком (торг) цены подписания контракта на выполнение проекта.

- Т.к. основными затратами на выполнение проекта являются усилия разработчиков ПО (и сл-но их зарплата), то стоимость проекта можно определить из объема работ по выполнению проекта, путем использования подходящей величины стоимости человека-месяца (месячной зарплаты).
- Оценки объема работ (трудоемкости) и сроков выполнения также требуются
 - для определения количества разработчиков на разных этапах проекта,
 - для детального планирования и
 - для мониторинга проекта.

Подходы к оценке трудоемкости

- Нисходящий подход к оценке трудоемкости
 - оценка основывается на размере проекта;
 - трудоемкость рассчитывается, как функция размера проекта.
 - для учета специфики вводятся поправочные коэффициенты.
- Восходящий подход к оценке трудоемкости
 - проект делится на задачи,
 - получаются оценки для каждой задачи проекта.
 - из оценок разных задач определяется общая

Нисходящий подход. Формула оценка трудоемкости проекта разработки ПО

- определение трудоемкости на основе размера проекта:

$$T = a * SIZE^b,$$

- где
 - T – трудоемкость в чел-месяцах;
 - a и b - константы,
 - $SIZE$ - размер проекта (обычно в KLOC). KLOC = тысячи передаваемых строк кода ($SIZE$)
- Значения используемых констант a и b для конкретной организации определяется с помощью **регрессионного статистического анализа**, который применяется к данным проектов, которые выполнялись ранее.

Источники затрат проекта

- Для оценки влияния других факторов используется до 15 разных показателей проекта, называемых - источниками затрат.
- Примерами источников затрат являются:
 - требование надежности ПО;
 - сложность ПО;
 - способности аналитиков;
 - использование современных инструментов разработки;
 - жесткость сроков разработки.
- Каждый источник затрат имеет качественную оценочную шкалу (высокий, низкий, нормальный) и для каждой оценки задается коэффициент влияния.
- Этот коэффициент влияния умножается на оценку трудоемкости:

$$T' = T * k_1 * k_2 * k_3 * \dots * k_n$$

Таблица коэффициентов трудоемкости для разных источников затрат

Источник затрат	Оценки				
	Очень низкая	Низкая	Нормальная	Высокая	Очень высокая
Свойства продукта					
RELY, требуемая надежность	0.75	0.88	1.00	1.15	1.40
DATA, размер БД		0.94	1.00	1.08	1.16
CPLX, сложность ПС	0.70	0.85	1.00	1.15	1.30
Свойства компьютера					
TIME, ограничение на время выполнения			1.00	1.11	1.30
STOR, ограничение на основную память			1.00	1.06	1.21
VITR, изменчивость виртуальной машины		0.87	1.00	1.15	1.30
TURN, время обработки заданий на компьютере		0.87	1.00	1.07	1.15

Восходящий подход

- Вначале определяются основные единицы ПО (компоненты или модули), которые должны быть разработаны в проекте.
 - Каждая программная единица, на основании некоторого критерия, оценивается, как **простая**, **средняя** или **сложная** (категория классификации).
- Для каждой категории определяется средняя трудоемкость кодирования, которая основывается
 - на прошлых данных из аналогичных проектов;
 - на основе методических указаний;
 - на основе опыта специалистов.
- Определяется общая оценка трудоемкости кодирования, т.к. для каждой категории известно кол-во прогр. единиц и оценка трудоемкости кодирования.

Поэтапное описание метода

1. Выявление в системе модулей и классификация их на простые, средние и сложные.
2. Определение средней трудоемкости кодирования для простых/средних/сложных модулей.
3. Вычисление общей трудоемкости кодирования на основе трудоемкости разных типов модулей и их количества.
4. Оценивание трудоемкости других работ и общей трудоемкости на основе распределения трудоемкостей работ в аналогичных проектах.
5. Уточнение оценок на основе специфических

Управление командой проекта

Состав команды определяется опытом и уровнем коллектива, особенностями проекта, применяемыми технологиями и уровнем этих технологий.

«Классический» вариант состава команды включает следующие позиции

Менеджер проекта - главное действующее лицо, обладающее знаниями и навыками, необходимыми для успешного управления проектом. Его основные функции:

- о Подбор и управление кадрами
- о Подготовка и исполнение плана проекта
- о Руководство командой
- о Обеспечение связи между подразделениями
- о Обеспечение готовности продукта

Проектировщик - это функция проектирования архитектуры высокого уровня и контроля ее выполнения. В небольших командах функция распределяется между менеджером и разработчиками. В больших проектах это может быть целый отдел. Основными функциями проектирования являются:

- о Анализ требований
- о Разработка архитектуры и основных интерфейсов
- о Участие в планировании проекта
- о Контроль выполнения проекта
- о Участие в подборе кадров

Разработчик – роль, ответственная за непосредственное создание конечного продукта. Помимо собственно программирования (кодирования) в его функции входит:

- о Контроль архитектурных и технических спецификаций продукта
- о Подбор технологических инструментов и стандартов
- о Диагностика и разрешение всех технических проблем
- о Контроль за работой разработчиков документации, тестирования, технологов
- о Мониторинг состояния продукта (ведение списка обнаруженных ошибок)
- о Подбор инструментов разработки, метрик и стандартов. Контроль их использования.

· **Тестировщик – роль, ответственная за удовлетворение требований к продукту**

- (функциональных и нефункциональных). В функции тестировщика входит:
- o Составление плана тестирования. План тестирования составляет один из элементов проекта и составляется до начала реализации (разработки) проекта. Время, отводимое в плане на тестирование может быть сопоставимо с временем разработки.
 - o Контроль выполнения плана. Важнейшая функция контроля – поддержка целостности базы данных зарегистрированных ошибок. В этой базе регистрируется: кто, когда и где обнаружил, описание ошибки, описание состояния среды; статус ошибки: приоритет, кто разрешает; состояние ошибки: висит, в разработке, разрешена, проблемы
 - o Разработка тестов. Самая трудоемкая часть в работе тестировщика. Тестирование должно обеспечить полную проверку функциональности при всех режимах работы продукта.
 - o Автоматизация тестирования включает автоматизацию составления тестов, автоматизацию пропуска тестов и автоматизацию обработки результатов тестирования. В виду важности автоматизации тестирования, иногда вводят нового участника – инженера по автоматизации.
 - o Выбор инструментов, метрик, стандартов для организации процесса тестирования.
 - o Организация Бета тестирования - тестирования почти готового продукта внешними тестерами (пользователями).

Инженер по качеству.

- о Составление плана качества. План качества включает все мероприятия по повышению качества (на всех уровнях). Имеет долговременный характер. План тестирования – его оперативная составляющая.
- о Описание процессов. Описание процессов является их формализацией. При описании вводятся метрики процесса, влияющие на качество продукта.
- о Оценка процессов включает регистрацию хода выполнения процессов и оценку значений установленных метрик процессов. Выявление «слабых» мест и выработка рекомендаций по улучшению процессов.
- о Улучшение процессов - переопределение процесса, автоматизация части работ, обучение персонала.

Повышение качества процессов требует участия всех действующих лиц.

Технический писатель или разработчик пользовательской (и иной) документации как части программного продукта. Функциями технического писателя являются:

- о Разработка плана документирования, который включает состав, сроки подготовки и порядок тестирования документов.
- о Выбор и разработка стандартов и шаблонов подготовки документов
- о Выбор средств автоматизации документирования
- о Разработка документации
- о Организация тестирования документации
- о Участие в тестировании продукта

Технолог разработки ПО обеспечивает выполнение следующих задач:

- о Поддержка модели ЖЦ - создание служб и структур по поддержке работоспособности принятой модели ЖЦ ПО.
- о Создание и сопровождение среды сборки продукта. Функция особенно важна на завершающих этапах разработки или при использовании модели прототипирования. В такой ситуации сборка будет проводиться достаточно часто (в некоторых случаях - ежедневно). Среда сборки должна быть подготовлена заранее, сборка должна проводиться быстро и без сбоев. С учетом сборки версий это не простая задача.
- о Создание и сопровождение процедуры установки с тем, чтобы каждая сборка устанавливалась автоматически с учетом версии и конфигураций сред.
- о Управление исходными текстами - сопровождение и администрирование системы управления версиями исходных текстов.

Модели управления командой

Административная модель (теория X)

- Властная пирамида – решения принимаются сверху-вниз.
- Четкое распределение ролей и обязанностей.
- Четкое распределение ответственности.
- Следование инструкциям, процедурам, технологиям.
- Роль менеджера: планирование, контроль, принятие основных решений

Модель хаоса (теория Y)

- Отсутствие явно выраженных признаков власти.
- Роль менеджера – поставить задачу, обеспечить ресурсами, не мешать и следить, чтобы не мешали другие.
- Отсутствие инструкций и регламентированных процедур.
- Индивидуальная инициатива - решения по проблеме принимаются там, где проблема обнаружена.
- Процесс напоминает творческую игру участников на основе дружеской соревновательности.

Открытая архитектура (теория Z)

- Адаптация к условиям работы
- Коллективное обсуждение проблем, выработка консенсуса и принятие решения – не все могут согласиться, но принятое решение является коллективным и в силу этого – обязательным для всех.
- Распределенная ответственность – отвечают все, кто обсуждал, выработывал, принимал.
- Динамика состава рабочих групп в зависимости от текущих задач.
- Отсутствие специализации – участники меняются ролями и функциями и могут при необходимости заменить друг друга.
- Задача менеджера – активное (но рядовое, не руководящее) участие в процессе, контроль конструктивности обсуждений, обеспечение возможности активного участия всех.

- **Управление проектом** состоит из:

- структурного и
- календарного планирования и
- оперативного управления

- **Структурное планирование** заключается в

- разбиении проекта на этапы и работы,
- оценки их длительности,
- определении последовательности их выполнения

Результатом структурного планирования является сетевой график работ, который используется для оптимизации проекта по длительности

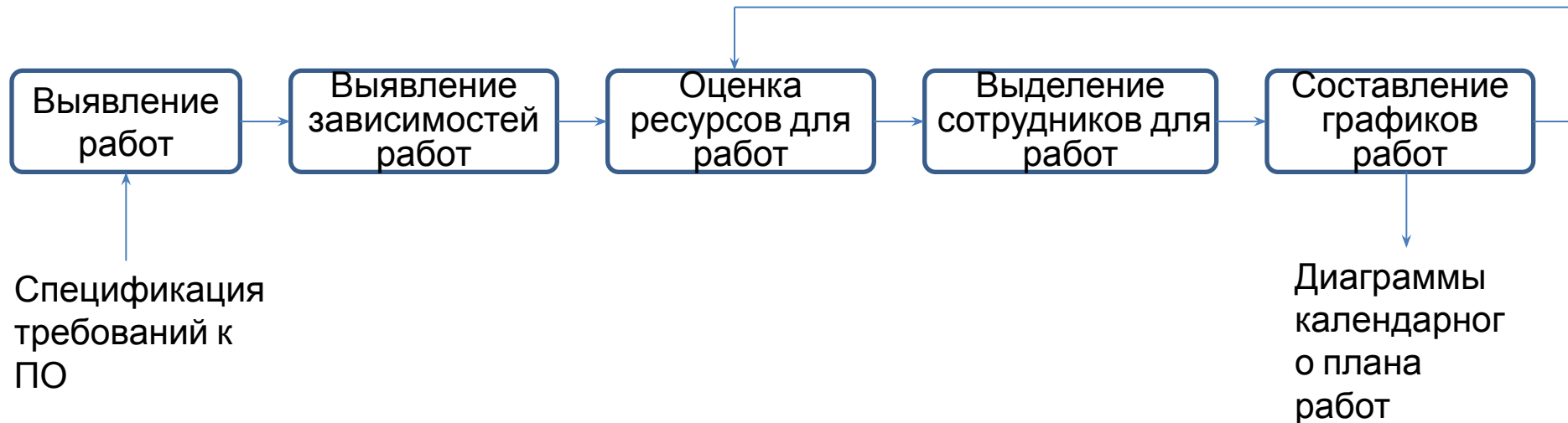
- **Календарное планирование** заключается в составлении временной диаграммы работ и распределении между работами трудовых ресурсов (исполнителей).

Результатом календарного планирования является диаграмма Ганта, графически отображающая периоды выполнения работ на оси времени. На этом этапе может выполняться оптимизация ресурсов и бюджета проекта.

- **Оперативное управление** состоит в регулярном сопоставлении фактического графика работ с плановым.

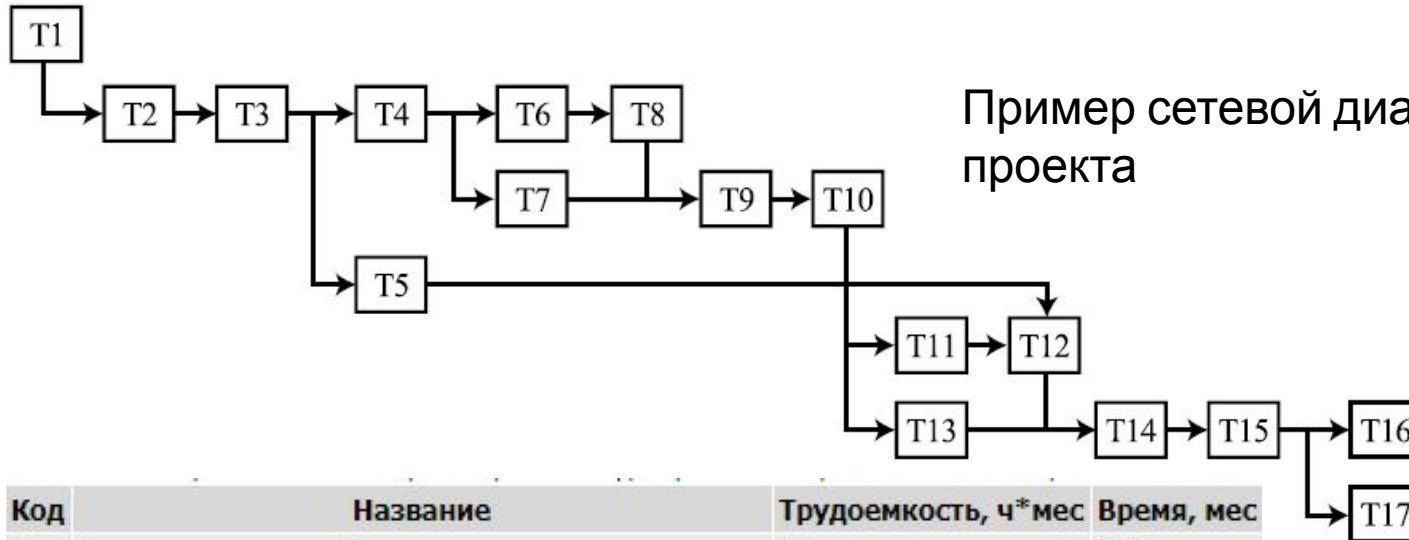
Результатом серьезных отклонений является принятие решений об изменении первоначального структурного или календарного плана.

Схема составления диаграммы календарного плана работ



Управление ресурсами

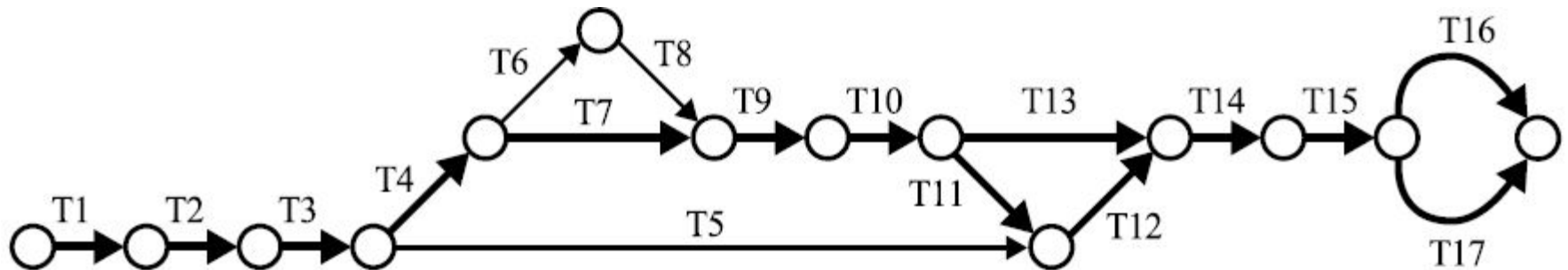
- При разработке графика проекта нужно выполнить следующие действия:
- Уточнить имеющуюся структуру работ проекта для того, чтобы использовать ее в рамках выбранного процесса разработки.
- Установить зависимости между отдельными работами, присутствующими в уточненной структуре работ проекта. Зависимости могут иметь разный характер: "финиш-старт" (работа может быть начата только после конца другой), "старт-старт" (работа может быть начата только с началом другой), "старт-финиш" и "финиш-финиш". Если вы встречаете более сложную зависимость типа "работу можно начать, только если сделана некоторая часть другой", это признак того, что работы декомпозированы недостаточно детально и нужно разбить вторую работу на части. Вставив *фиктивные работы*, не требующие *ресурсов* и времени, можно все зависимости привести к виду "финиш-старт".



Пример сетевой диаграммы проекта

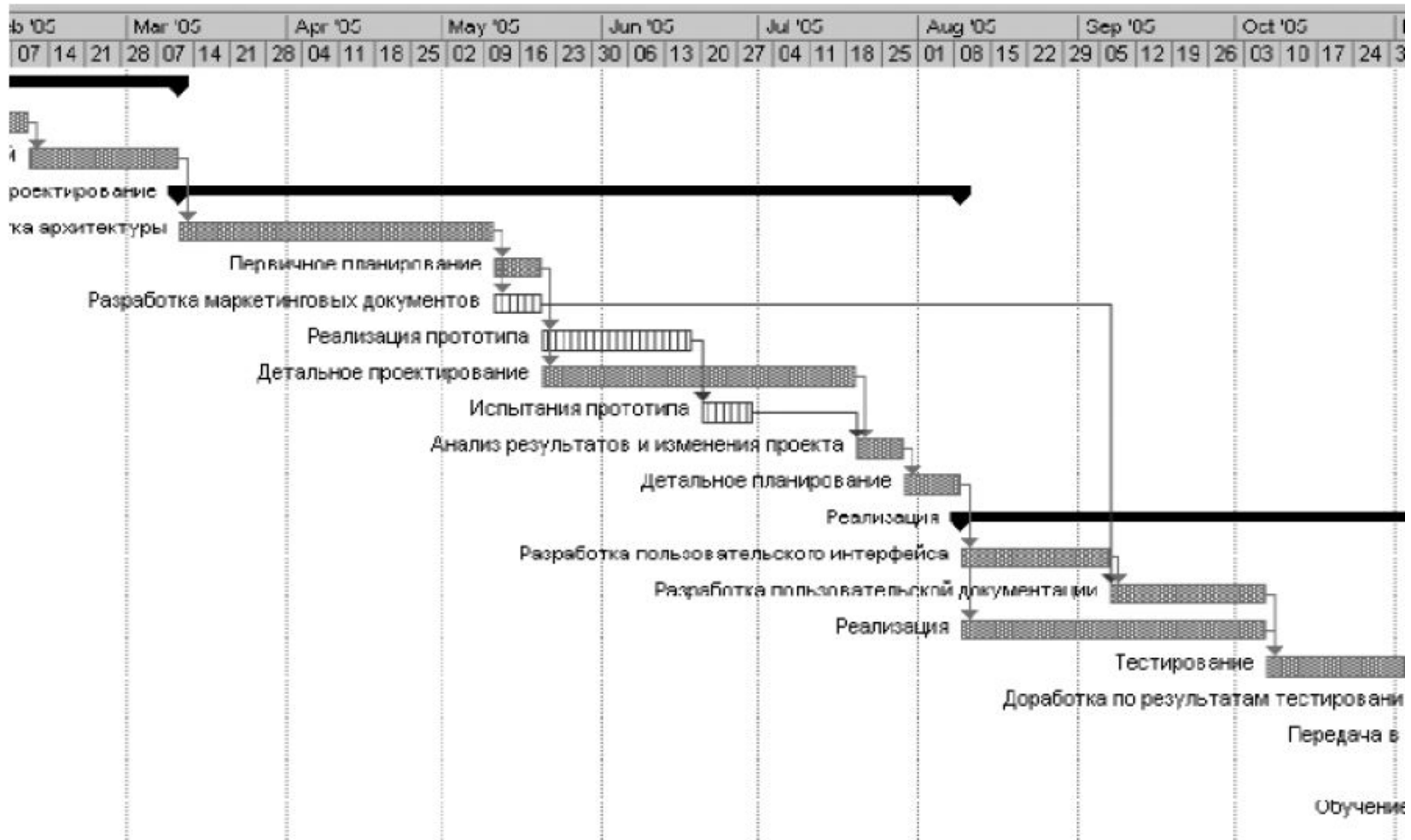
Код	Название	Трудоемкость, ч*мес	Время, мес
T1	Формулировка <i>целей и содержания проекта</i>	0,6	0,3
T2	Сбор и анализ требований	3,0	1,0
T3	Разработка архитектуры	6,0	2,0
T4	Первичное планирование	0,6	0,3
T5	Разработка маркетинговых документов	0,3	0,3
T6	Реализация прототипа	3,0	1,0
T7	Детальное проектирование	6,0	2,0
T8	Испытания прототипа	0,6	0,3
T9	Анализ результатов испытаний и изменения проекта	1,0	0,3
T10	Детальное планирование	1,0	0,3
T11	Разработка и отладка пользовательского интерфейса	3,0	1,0
T12	Разработка пользовательской документации	2,0	1,0
T13	Реализация	8,0	2,0
T14	Тестирование	4,0	1,0
T15	Доработка по результатам тестирования	4,0	1,0
T16	Развертывание	1,5	0,5
T17	Обучение <i>пользователей</i>	2,0	0,5

- ***PERT-диаграмма*** (*PERT* — program evaluation and review technique, техника оценки и обзора программ). Часто различий между этими двумя видами диаграмм не делают, называя обе *и сетевыми, и PERT-диаграммами*.



- *Сетевые и PERT-диаграммы* используются для планирования продолжительности проекта и выделения *критических путей* — последовательностей работ от начала до конца проекта, сумма длительностей которых максимальна среди таких последовательностей. В примере, *критических путей* несколько — работы, лежащие на них, изображены жирными стрелками. Выполнить проект быстрее, чем за время, требующееся для прохождения по *критическому пути*, нельзя. Поэтому *критические пути* используют для планирования основных поставок в ходе проекта. В нашем примере длительность проекта не может быть меньше, чем 10,7 месяцев. Кроме того, любая задержка в одной из работ, попавшей на *критический путь*, обязательно вызовет задержку проекта в целом, а значит, такие работы требуют повышенного внимания во время проекта.

- Расписание работ удобно изображать с помощью **диаграмм Ганта (Gantt chart)**. Эта диаграмма показывает и связи между работами, и их длительность во времени.
- Показаны 4 выделенных группы работ: начало, проектирование, реализация и передача в использование.



- На следующем шаге (после вынесения предварительной оценки трудоемкости и продолжительности работ) нужно привязать их к имеющемуся в проекте персоналу и другим *ресурсам* (оборудованию, материалам и пр.). При этом может оказаться, что некоторые независимые работы не могут проводиться одновременно, поскольку для этого не хватает *ресурсов*.
- Допустим, что первичное планирование и разработка маркетинговых документов требуют полной вовлеченности *менеджера* проекта. При этом вторая работа может быть начата только после окончания первой. В разработке пользовательского интерфейса могут быть задействованы аналитик, проектировщик интерфейсов и программист. При ограниченности *команды* проекта это потребует увеличить время на выполнение работы по реализации. Итоговая картина после перепланирования рассматриваемого проекта для *команды* из *менеджера*, аналитика-тестировщика, архитектора, трех программистов, один из которых также является проектировщиком пользовательского интерфейса, и еще одного тестировщика, способного писать техническую документацию, Она отличается от предыдущей изменениями, удлиннившими проект на 0.5 месяца.
- Естественно, после учета доступных *ресурсов критические пути* проекта могут измениться.
- Кроме того, риски, связанные с непредвиденными ситуациями, неаккуратной оценкой возможностей персонала или технологий и пр., требуют наличия определенных временных и ресурсных резервов при планировании проектов.
- В ходе самого проекта необходимо тщательно следить за выполнением запланированных работ в срок, за доступностью *ресурсов*, отслеживать проблемы, способные привести к срыву графика, а также неучтенные при первоначальном планировании факторы. Реальные длительности и трудоемкости работ могут отличаться от оценочных, что потребует построения новых, уточненных планов проекта. Грамотная отработка изменений в планах и расписаниях — ничуть не менее важная задача, чем их первоначальное составление. В частности, чем быстрее *менеджер* проекта проинформирует о возможных или необходимых изменениях *заказчика*, *спонсора* и других лиц, тем более он может рассчитывать на их понимание и помощь в решении проблем проекта.

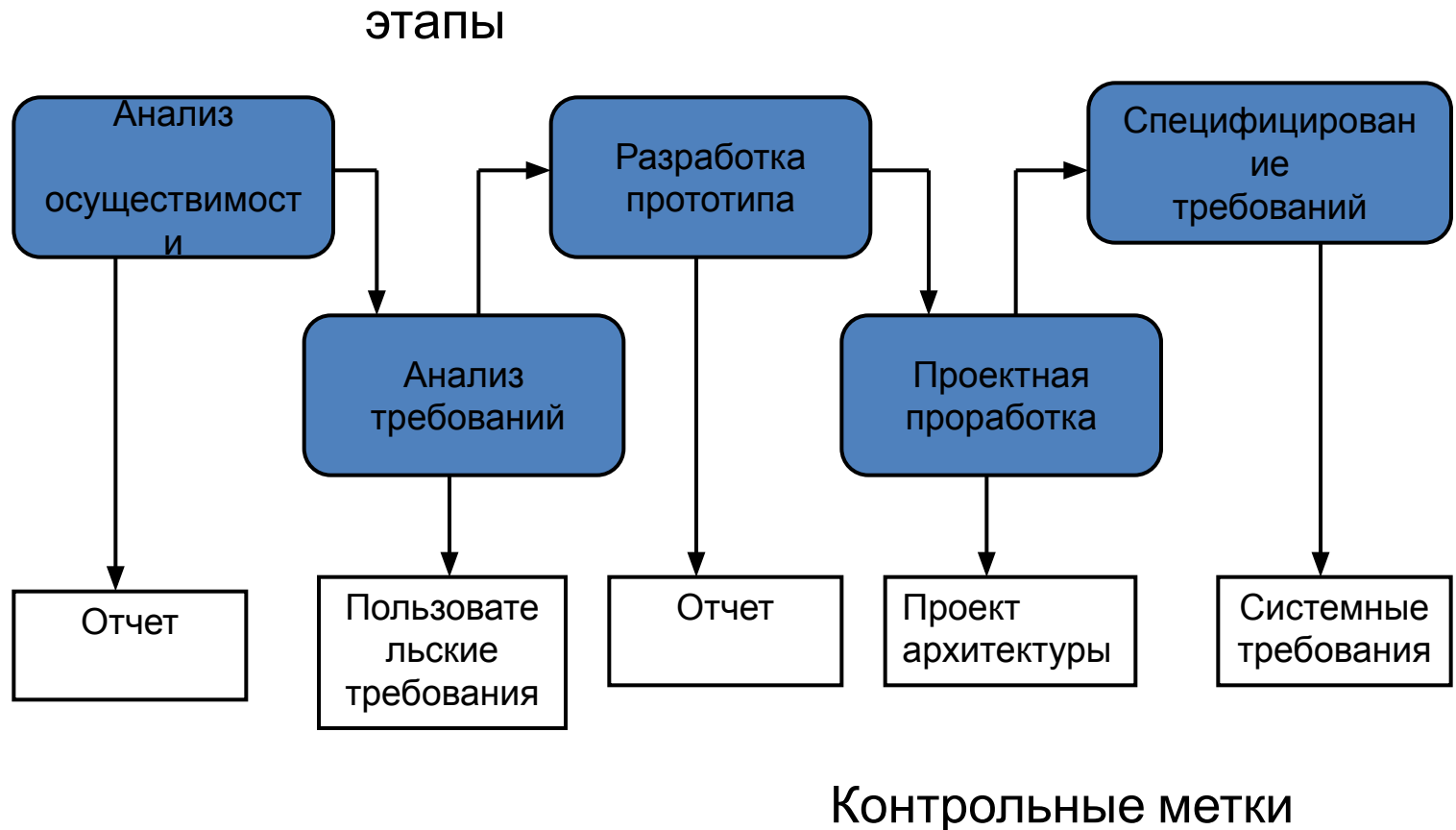
Промежуточные результаты

- Специальным видом КТ является создание **промежуточного результата** (ПР, *deliverable*) – некоторого продукта, который передается заказчику.
- Например:
 - спецификация требований к ПО;
 - проект (модель) ПО;
 - Подсистема и т.п.
- Обычно требуемые ПР фиксируются в контракте (хоз. договоре) проекта.

Контрольные отметки

Контрольные отметки— вехи, отмечающие окончание определенного этапа работ.

По завершении основных больших этапов, таких как разработка спецификации, проектирование и т.п., заказчику ПО предоставляются результаты их выполнения, так называемые *контрольные проектные элементы*.



Планирование управления рисками

- Программный проект является сложным мероприятием, которое требует
 - создание ПО высокого качества
 - в заданные сроки и
 - за заданные деньги (бюджет).
- На выполнение программного проекта отрицательное влияние могут оказать не предвиденные события.
- Управление рисками это попытка минимизировать вероятность неудачи, вызванной не запланированными событиями.
- Управление рисками это не отказ от проекта, который имеет риски, а работа по предвидению, выявлению и минимизации рисков в выполняемом

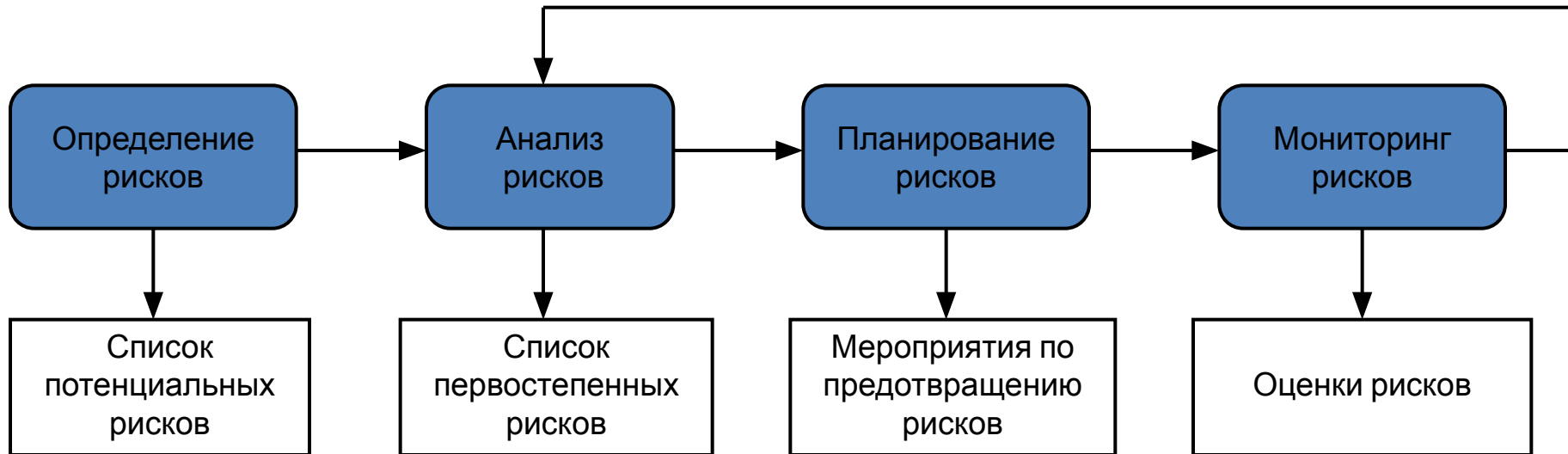
Основные понятия управления рисками

- Риск это возможность подвергнуться повреждению или поражению.
- Т.е., риск предполагает, что существует возможность, что может случиться что-то нежелательное (негативное).
- Управление рисками это вид деятельности, которая гарантирует, что влияние рисков на стоимость, качество и сроки выполнения проекта будет минимальным.
- Управление рисками начинается тогда, когда заканчивается нормальный ход управления проектом

Управление рисками

1. *Определение рисков.* Определяются возможные риски для проекта, для разрабатываемого продукта и бизнес-риски.
2. *Анализ рисков.* Оценивается вероятность и последовательность появления рисков ситуаций.
3. *Планирование рисков.* Планируются мероприятия по предотвращению рисков или минимизации их воздействия на проект.
4. *Мониторинг рисков.* Постоянное оценивание вероятностей рисков и выполнение мероприятий по смягчению последствий проявления рисков ситуаций.

Схема процесса управления рисками



Основные риски и методы управления ими

№	Вид риска	Метод управления риском
1	Недостаток сотрудников	Поиск талантливых сотрудников; соответствие способностей сотрудника его работе; создание команд; обучение; и т.п.
2	Не реальные сроки и финансы (бюджет)	Детальное оценивание затрат графика работ; проектирование согласно заданной стоимости; итеративная разработка; повторное использование компонент; уточнение требований.
3	Разработка не верных функций ПО	Анализ организации; обследование пользователей; прототипирование; подготовка инструкций пользователей заранее.
4	Разработка не верного интерфейса пользователей	Прототипирование; сценарии; анализ задач; определение характеристик пользователей.
5	Не нужные украшения	Уточнение требований; прототипирование; анализ стоимости и эффективности; проектирование согласно заданной стоимости

Анализ рисков

Существует три категории стратегий управления рисками:

- 1. Стратегии предотвращения рисков.* Согласно этим стратегиям следует проводить мероприятия, снижающие вероятность проявления рисков. Примером может служить стратегия исключения потенциально дефектных компонентов.
- 2. Минимизационные стратегии.* Направлены на уменьшение возможного ущерба от рисков. Примером служит стратегия уменьшения ущерба от болезни членов команды разработчиков.
- 3. Планирование "аварийных" ситуаций.* Согласно этим стратегиям необходимо иметь план мероприятий, которые следует выполнить в случае проявления рисков ситуации. В табл. 4.7 это стратегия поведения при возникновении финансовых проблем у организации разработчика.

Показатели рисков

Тип риска	Возможные индикаторы
Технологический	Поздняя доставка техн. или прогр. обеспечения; много сообщений о технических проблемах.
Люди	Плохое моральное состояние сотрудников; плохие отношения между членами команды разработчиков; большая смена текучесть кадров.
Организационный	Организационные слухи; недостаток действий руководства организации.
Инструментальный	Нежелание членов команды использовать инструменты; жалобы на CASE инструменты; требования более мощных рабочих компьютеров.
Требования	Много отчетов о изменении требований; жалобы пользователей.
Оценки	Сложность составить согласованное расписание; трудность ясно определить дефекты.

OpenProj – бесплатный аналог Microsoft Project, предназначенный для планирования проектов (<http://sourceforge.net/projects/openproj/> или <http://openproj.ru.uptodown.com/> – свободное скачивание).