
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И
РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

УТВЕРЖДАЮ

Заведующий кафедрой
«Управление инновациями»

_____ /*А.Ф.Уваров*
(подпись) (ФИО)
" _____ " _____ 2012 г.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ ЗАНЯТИЯМ
И К ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ**

по дисциплине

**ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА
ПРОЕКТИРОВАНИЯ БИЗНЕС-ПРОЦЕССОВ**

Составлена кафедрой

«Управление инновациями»

Для студентов, обучающихся по специальности 220601.65 «Управление инновациями»

Форма обучения

очная

Составитель
к.ф.-м.н., доцент

_____ П.Н. Дробот

инженер

_____ О.В. Штымова

Томск 2012 г.

ОГЛАВЛЕНИЕ

Аннотация	3
Лабораторная работа №1 <i>Методология управления проектами</i>	3
Лабораторная работа №2 Моделирование бизнес-процессов.....	17
Лабораторная работа №3 Ознакомление с CASE-средством Rational rose.....	21
Лабораторная работа №4 Создание модели вариантов использования	25
Лабораторная работа №5 Календарное планирование проектов при помощи MS Project	27
Лабораторная работа №6 Анализ и оптимизация плана проекта при помощи MS Project	33
Список рекомендуемой литературы	39

Аннотация

Изучение инструментальных средств проектирования бизнес-процессов лучше всего осуществлять на практике, выполняя лабораторные, модельные задания, при выполнении которых студент учится основам проектирования бизнес-процессов, постепенно переходя к задачам возрастающей сложности. В настоящем учебно-методическом пособии описаны соответствующие лабораторные работы, в следующем составе: 1) методология управления проектами; 2) моделирование бизнес процессов; 3) ознакомление с CASE-средством Rational Rose; 4) создание модели вариантов использования; 5) календарное планирование проектов при помощи MS Project; 6) анализ и оптимизация плана проекта при помощи MS Project.

В пособии разъясняется ход выполнения работ, даются конкретные этапные задания выполнения каждой лабораторной работы. Методические рекомендации по организации самостоятельной работы студентов, в соответствии с учебным планом и с рабочей программой дисциплины разъясняют студентам методику углубленного самостоятельного изучения лекционного материала и отдельных вопросов курса. Даются рекомендации по самостоятельному изучению тем лабораторных работ.

Лабораторная работа №1 *Методология управления проектами*

1. Цель работы:

Изучение методологии управления проектами. Получение навыков по применению данных методологий для планирования проекта.

2. Методические указания

Лабораторная работа направлена на ознакомление с основными понятиями методологии управления проектами, получение навыков по применению данных понятий при построении плана проекта, построения графика работ, распределения исполнителей, управления рисками. Планирование проекта

Эффективное управление программным проектом напрямую зависит от правильного планирования работ, необходимых для его выполнения. План помогает менеджеру предвидеть проблемы, которые могут возникнуть на каких-либо этапах создания ПО, и разработать превентивные меры для их предупреждения или решения. План, разработанный на начальном этапе проекта, рассматривается всеми его участниками как руководящий документ, выполнение которого должно привести к успешному завершению проекта. Этот первоначальный план должен максимально подробно описывать все этапы реализации проекта.

Кроме разработки плана проекта, на менеджера ложится обязанность разработки других видов планов. Эти виды планов кратко описаны в табл. 1.

Таблица 1 - Виды планов

План	Описание
План качества	Описывает стандарты и мероприятия по поддержке качества разрабатываемого ПО
План аттестации	Описывает способы, ресурсы и перечень работ, необходимых для аттестации программной системы
План управления конфигурацией	Описывает структуру и процессы управления конфигурацией

План сопровождения ПО	Предлагает план мероприятий, требующихся для сопровождения ПО в процессе его эксплуатации, а также расчет стоимости сопровождения и необходимые для этого ресурсы
План по управлению персоналом	Описывает мероприятия, направленные на повышение квалификации членов команды разработчиков

В листинге 1 показан процесс планирования создания ПО в виде псевдокода. Здесь сделан акцент на том, что планирование — это итерационный процесс. Поскольку в процессе выполнения проекта постоянно поступает новая информация, план должен регулярно пересматриваться. Важными факторами, которые должны учитываться при разработке плана, являются финансовые и деловые обязательства организации. Если они изменяются, эти изменения также должны найти отражение в плане работ.

Листинг 1. Процесс планирования проекта

Определение проектных ограничений

Первоначальная оценка параметров проекта

Определение этапов выполнения проекта и контрольных отметок

while пока проект не завершится или не будет остановлен loop

Составление графика работ

Начало выполнения работ

Ожидание окончания очередного этапа работ

Отслеживание хода выполнения работ

Пересмотр оценок параметров проекта

Изменение графика работ

Пересмотр проектных ограничений

if (возникла проблема) then

Пересмотр технических или организационных параметров проекта

end if

end loop

Процесс планирования начинается с определения проектных ограничений (временные ограничения, возможности наличного персонала, бюджетные ограничения и т.д.). Эти ограничения должны определяться параллельно с оцениванием проектных параметров, таких как структура и размер проекта, а также распределением функций среди исполнителей. Затем определяются этапы разработки и то, какие результаты документация, прототипы, подсистемы или версии программного продукта) должны быть

получены по окончании этих этапов. Далее начинается циклическая часть планирования. Сначала разрабатывается график работ по выполнению проекта или дается разрешение на продолжение использования ранее созданного графика. После этого (обычно через 2-3 недели) проводится контроль выполнения работ и отмечаются расхождения между реальным и плановым ходом работ.

Далее, по мере поступления новой информации о ходе выполнения проекта, возможен пересмотр первоначальных оценок параметров проекта. Это, в свою очередь, может привести к изменению графика работ. Если в результате этих изменений нарушаются сроки завершения проекта, должны быть пересмотрены (и согласованы с заказчиком ПО) проектные ограничения.

Конечно, большинство менеджеров проектов не думают, что реализация их проектов пройдет гладко, без всяких проблем. Желательно описать возможные проблемы еще до того, как они проявят себя в ходе выполнения проекта. Поэтому лучше составлять "пессимистические" графики работ, чем "оптимистические". Но, конечно, невозможно построить план, учитывающий все, в том числе случайные, проблемы и задержки выполнения проекта, поэтому и возникает необходимость периодического пересмотра проектных ограничений и этапов создания программного продукта.

План проекта

План проекта должен четко показать ресурсы, необходимые для реализации проекта, разделение работ на этапы и временной график выполнения этих этапов. В некоторых организациях план проекта составляется как единый документ, содержащий все виды планов, описанных выше. В других случаях план проекта описывает только технологический процесс создания ПО. В таком плане обязательно присутствуют ссылки на планы других видов, но они разрабатываются отдельно от плана проекта.

План, представленные ниже, принадлежит именно к последнему типу планов. Детализация планов проектов очень разнится в зависимости от типа разрабатываемого программного продукта и организации-разработчика. Но в любом случае большинство планов содержат следующие разделы.

Введение. Краткое описание целей проекта и проектных ограничений (бюджетных, временных и т.д.), которые важны для управления проектом.

Организация выполнения проекта. Описание способа подбора команды разработчиков и распределение обязанностей между членами команды.

Анализ рисков. Описание возможных проектных рисков, вероятности их проявления и стратегий, направленных на их уменьшение. Тема управления рисками рассмотрена ниже.

Аппаратные и программные ресурсы, необходимые для реализации проекта. Перечень аппаратных средств и программного обеспечения, необходимого для разработки программного продукта. Если аппаратные средства требуется закупать, приводится их стоимость совместно с графиком закупки и поставки.

Разбиение работ на этапы. Процесс реализации проекта разбивается на отдельные процессы, определяются этапы выполнения проекта, приводится описание результатов ("выходов") каждого этапа и контрольные отметки. Эта тема представлена ниже.

График работ. В этом графике отображаются зависимости между отдельными процессами (этапами) разработки ПО, оценки времени их выполнения и распределение членов команды разработчиков по отдельным этапам.

Механизмы мониторинга и контроля за ходом выполнения проекта. Описываются предоставляемые менеджером отчеты о ходе выполнения работ, сроки их предоставления, а также механизмы мониторинга всего проекта.

План должен регулярно пересматриваться в процессе реализации проекта. Одни части плана, например график работ, изменяются часто, другие более стабильны. Для внесения изменений в план требуется специальная организация документопотока, позволяющая отслеживать эти изменения.

Контрольные отметки этапов работ

Менеджеру для организации процесса создания ПО и управления им необходима информация. Поскольку само программное обеспечение неосознано, эта управленческая информация может быть получена только в виде документов, отображающих выполнение очередного этапа разработки программного продукта. Без этой информации нельзя судить о степени готовности создаваемого продукта, невозможно оценить произведенные затраты или изменить график работ.

При планировании процесса определяются контрольные отметки— вехи, отмечающие окончание определенного этапа работ. Для каждой контрольной отметки создается отчет, который предоставляется руководству проекта. Эти отчеты не должны быть большими объемными документами; они должны подводить краткие итоги окончания отдельного логически завершенного этапа проекта. Этапом не может быть, например, "Написание 80% кода программ", поскольку невозможно проверить завершение такого "этапа"; кроме того, подобная информация практически бесполезна для управления, поскольку здесь не отображается связь этого "этапа" с другими этапами создания ПО.

Обычно по завершении основных больших этапов, таких как разработка спецификации, проектирование и т.п., заказчику ПО предоставляются результаты их выполнения, так называемые контрольные проектные элементы. Это может быть документация, прототип программного продукта, законченные подсистемы ПО и т.д. Контрольные проектные элементы, предоставляемые заказчику ПО, могут совпадать с контрольными отметками (точнее, с результатами выполнения какого-либо этапа). Но обратное утверждение неверно. Контрольные отметки — это внутренние проектные результаты, которые используются для контроля за ходом выполнения проекта, и они, как правило, не предоставляются заказчику ПО.

Для определения контрольных отметок весь процесс создания ПО должен быть разбит на отдельные этапы с указанным "выходом" (результатом) каждого этапа. Например, на рис. 1 показаны этапы разработки спецификации требований в случае, когда для ее проверки используется прототип системы, а также представлены выходные результаты (контрольные отметки) каждого этапа. Здесь контрольными проектными элементами являются требования и спецификация требований.

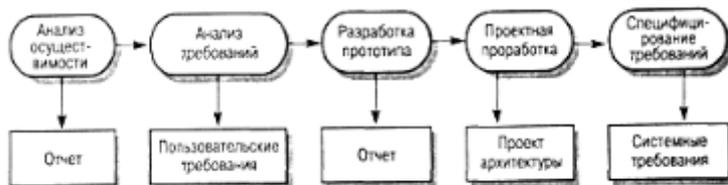


Рис. 1. Этапы процесса разработки спецификации
График работ

Составление графика - одна из самых ответственных работ, выполняемых менеджером проекта. Здесь менеджер оценивает длительность проекта, определяет ресурсы, необходимые для реализации отдельных этапов работ, и представляет их (этапы) в виде

согласованной последовательности. Если данный проект подобен ранее реализованному, то график работ последнего проекта можно взять за основу для данного проекта. Но затем следует учесть, что на отдельных этапах нового проекта могут использоваться методы и подходы, отличные от использованных ранее.

Если проект является инновационным, первоначальные оценки длительности и требуемых ресурсов наверняка будут слишком оптимистичными, даже если менеджер попытается предусмотреть все возможные неожиданности. С этой точки зрения проекты создания ПО не отличаются от больших инновационных технических проектов. Новые аэропорты, мосты и даже новые автомобили, как правило, появляются позже первоначально объявленных сроков их сдачи или поступления на рынок, чему причиной являются неожиданно возникшие проблемы и трудности. Именно поэтому графики работ необходимо постоянно обновлять по мере поступления новой информации о ходе выполнения проекта.

В процессе составления графика (рис. 2) весь массив работ, необходимых для реализации проекта, разбивается на отдельные этапы и оценивается время, требующееся для выполнения каждого этапа. Обычно многие этапы выполняются параллельно. График работ должен предусматривать это и распределять производственные ресурсы между ними наиболее оптимальным образом. Нехватка ресурсов для выполнения какого-либо критического этапа - частая причина задержки выполнения всего проекта.

Длительность этапов обычно должна быть не меньше недели. Если она будет меньше, то окажется ниже точности временных оценок этапов, что может привести к частому пересмотру графика работ. Также целесообразно (в аспекте управления проектом) установить максимальную длительность этапов, не превышающую 8 или 10 недель. Если есть этапы, имеющие большую длительность, их следует разбить на этапы меньшей длительности.

При расчете длительности этапов менеджер должен учитывать, что выполнение любого этапа не обойдется без больших или маленьких проблем и задержек. Разработчики могут допускать ошибки или задерживать свою работу, техника может выйти из строя либо аппаратные или программные средства поддержки процесса разработки могут поступить с опозданием. Если проект инновационный и технически сложный, это становится дополнительным фактором появления непредвиденных проблем и увеличения длительности реализации проекта по сравнению с первоначальными оценками.

Требования к ПО

Диаграммы процессов и временные диаграммы



Таблица 2 - Этапы проекта

Этап	Длительность (дни)	Зависимость
T1	8	
T2	15	
T3	15	T1 (M1)
T4	10	
T5	10	T2, T4 (M2)
T6	5	T1, T2 (M3)
T7	20	T1 (M1)
T8	25	T4 (M5)
T9	15	T3, T6 (M4)
T10	15	T5, T7 (M7)
T11	7	T9 (M6)
T12	10	T11 (M8)

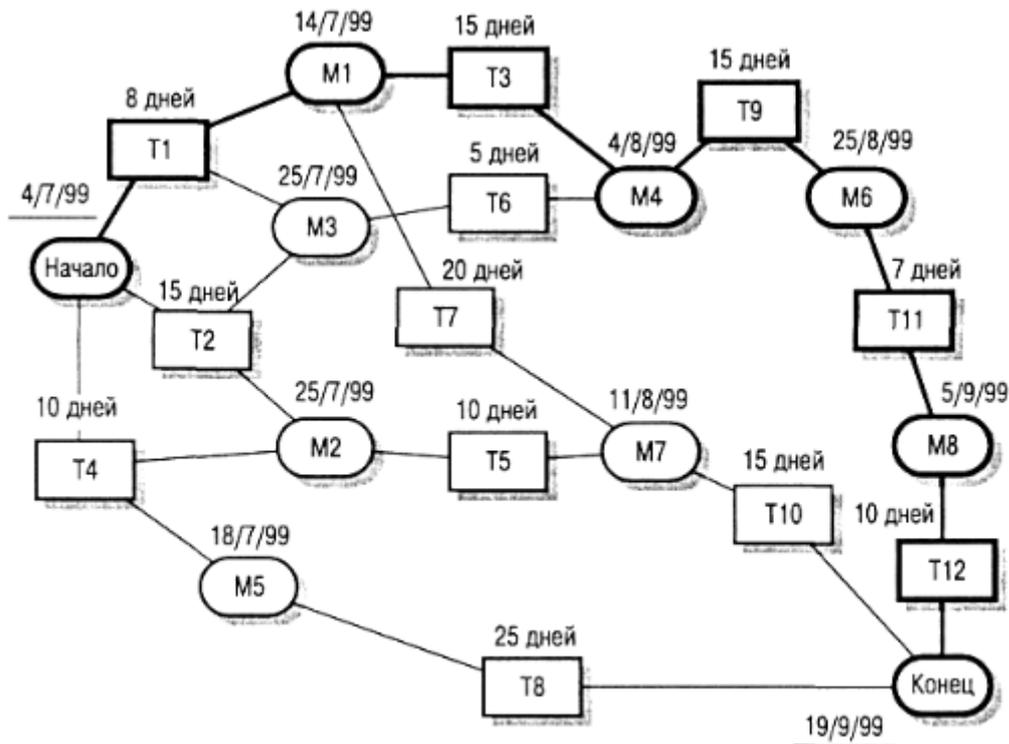


Рис. 3. Сетевая диаграмма этапов

Если для создания сетевой диаграммы используются программные средства поддержки управления проектом, каждый этап должен заканчиваться контрольной отметкой. Очередной этап может начаться только тогда, когда будет получена контрольная отметка (которая может зависеть от нескольких предшествующих этапов). Поэтому в третьем столбце табл. 2 приведены контрольные отметки; они будут достигнуты только тогда, когда будет завершен этап, в строке которого помещена соответствующая контрольная отметка.

Любой этап не может начаться, пока не выполнены все этапы на всех путях, ведущих от начала проекта к данному этапу. Например, этап T9 не может начаться, пока не будут завершены этапы T3 и T6. Отметим, что в данном случае достижение контрольной отметки M4 говорит о том, что эти этапы завершены.

Минимальное время выполнения всего проекта можно рассчитать, просуммировав в сетевой диаграмме длительности этапов на самом длинном пути (длина пути здесь измеряется не количеством этапов на пути, а суммарной длительностью этих этапов) от начала проекта до его окончания (это так называемый критический путь). В нашем случае

продолжительность проекта составляет 11 недель или 55 рабочих дней. На рис. 3 критический путь показан более толстыми линиями, чем остальные пути. Таким образом, общая продолжительность реализации проекта зависит от этапов работ, находящихся на критическом пути. Любая задержка в завершении любого этапа на критическом пути приведет к задержке всего проекта.

Задержка в завершении этапов, не входящих в критический путь, не влияет на продолжительность всего проекта до тех пор, пока суммарная длительность этих этапов (с учетом задержек) на каком-нибудь пути не превысит продолжительности работ на критическом пути. Например, задержка этапа T8 на срок, меньший 20 дней, никак не влияет на общую продолжительность проекта. На рис. 4 представлена временная диаграмма, на которой показаны возможные задержки на каждом этапе.

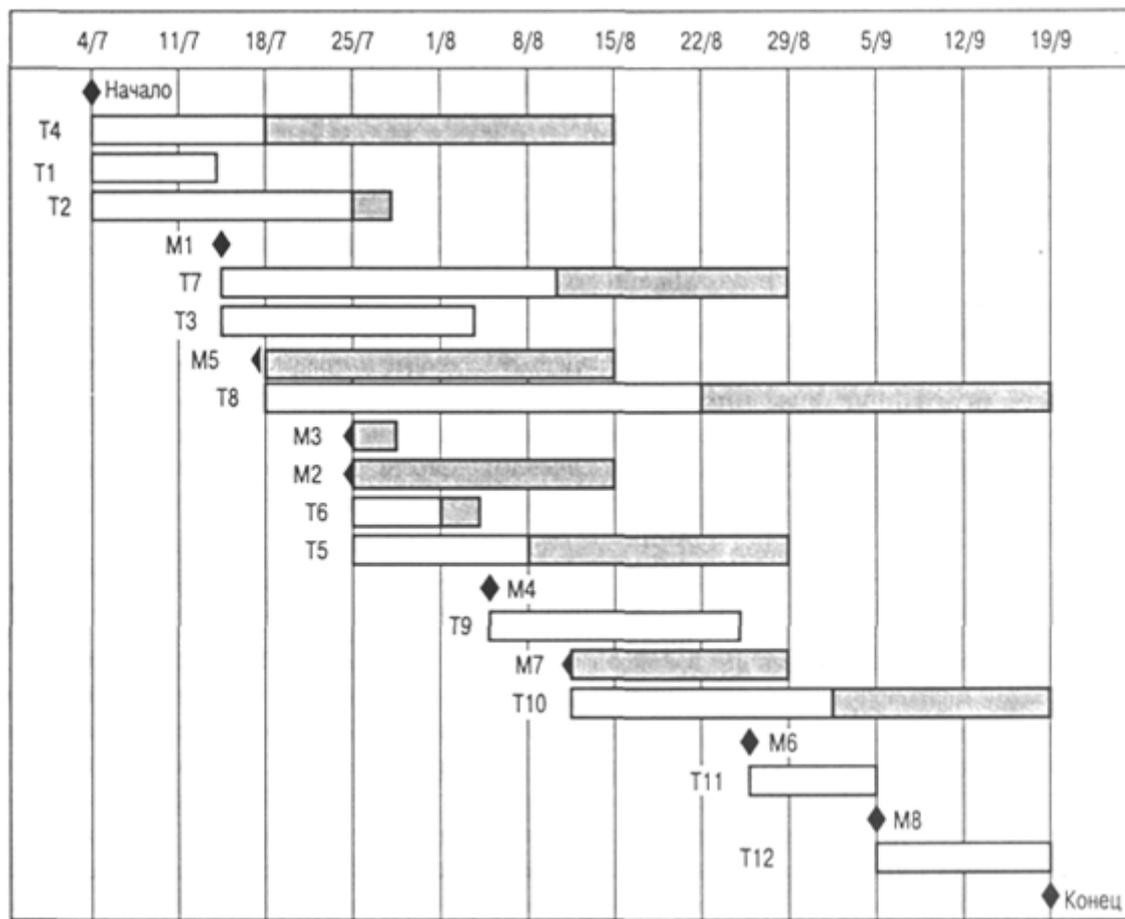


Рис. 4. Временная диаграмма длительности этапов

Сетевая диаграмма позволяет увидеть в зависимости этапов значимость того или иного этапа для реализации всего проекта. Внимание к этапам критического пути часто позволяет найти способы их изменения с тем, чтобы сократить длительность всего проекта. Менеджеры используют сетевую диаграмму для распределения работ.

На рис. 4 показано другое представление графика работ. Это временная диаграмма (иногда называемая по имени ее изобретателя диаграммой Гантта) может быть построена программными средствами поддержки процесса управления. Она показывает длительность выполнения каждого этапа и возможные их задержки (показаны затененными прямоугольниками), а также даты начала и окончания каждого этапа. Этапы критического пути не имеют затененных прямоугольников; это означает, что задержка с завершением данных этапов приведет к увеличению длительности всего проекта.

Подобно распределению времени выполнения этапов, менеджер должен рассчитать распределение ресурсов по этапам, в частности назначить исполнителей на каждый этап. В табл. 3 приведено распределение разработчиков на каждый этап, представленный на рис. 4.

Таблица 3 - Распределение исполнителей по этапам

Этап	Исполнитель
T1	Джейн
T2	Анна
T3	Джейн
T4	Фред
T5	Мэри
T6	Анна
T7	Джим
T8	Фред
T9	Джейн
T10	Анна
T11	Фред
T12	Фред

Приведенная таблица может быть использована программными средствами поддержки процесса управления для построения временной диаграммы занятости сотрудников на определенных этапах работ (рис. 5). Персонал не занят в работе над проектом все время его реализации. В течение периода незанятости сотрудники могут быть в отпуске, работать над другими проектами, проходить обучение и т.д.

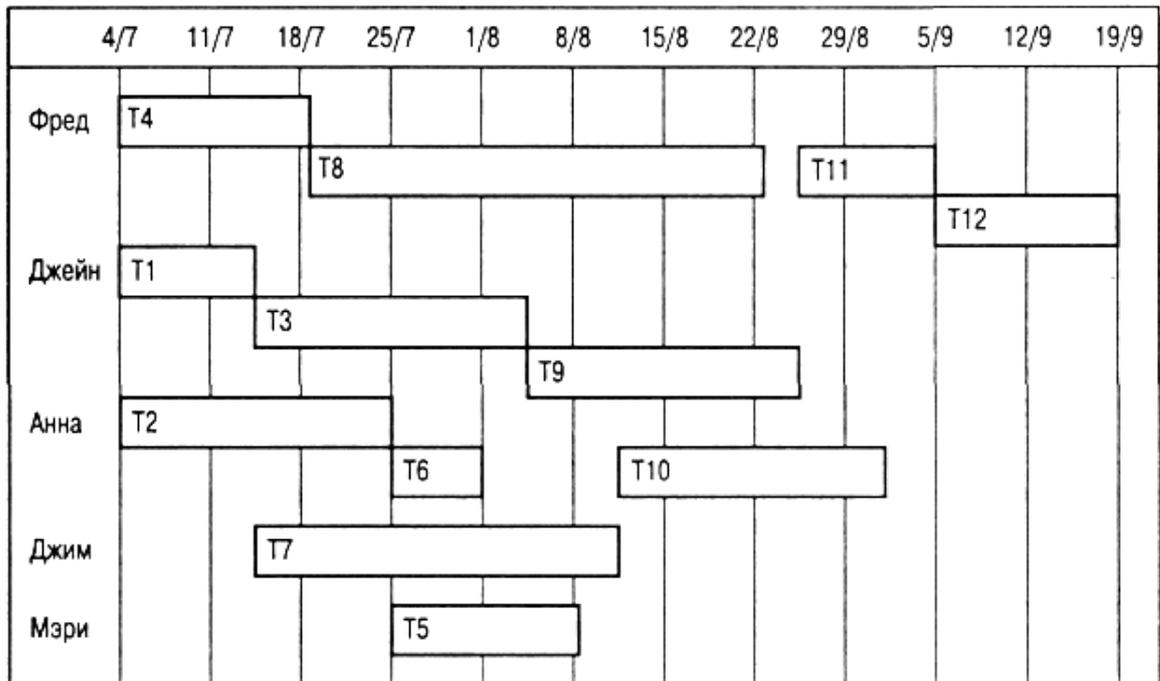


Рис. 5. Временная диаграмма распределения работников по этапам

В больших организациях обычно работает много специалистов, которые задействуются в проекте по мере необходимости. Конечно, такой подход может создать определенные

проблемы для менеджеров проектов. Например, если специалист занят в проекте, который задерживается, это может создать прямые сложности для других проектов, где он также должен участвовать.

Первоначальный график работ неизбежно содержит какие-нибудь ошибки или недоработки. По мере реализации проекта рассчитанные оценки длительности выполнения этапов работ должны сравниваться с реальными сроками выполнения этих этапов. Результаты сравнения должны использоваться в качестве основы для пересмотра графика работ еще не реализованных этапов проекта, в частности для того, чтобы попытаться уменьшить длительность этапов критического пути.

Управление рисками

Важной частью работы менеджера проекта является оценка рисков, которые могут повлиять на график работ или на качество создаваемого программного продукта, и разработка мероприятий по предотвращению рисков. Результаты анализа рисков должны быть отражены в плане проекта. Определение рисков и разработка мероприятий по уменьшению их влияния на ход выполнения проекта называется управлением рисками.

Упрощенно риск можно понимать как вероятность проявления каких-либо неблагоприятных обстоятельств, негативно влияющих на реализацию проекта. Риски могут угрожать проекту в целом, создаваемому программному продукту или организации-разработчику. Можно выделить три типа рисков.

Риски для проекта, которые влияют на график работ или ресурсы, необходимые для выполнения проекта.

Риски для разрабатываемого продукта, влияющие на качество или производительность разрабатываемого программного продукта.

Бизнес-риски, относящиеся к организации-разработчику или поставщикам.

Конечно, эти типы рисков могут пересекаться. Например, если опытный программист покидает проект, это будет риском для проекта (поскольку задерживается срок сдачи готового продукта), риском для продукта (так как новый программист, заменивший ушедшего, может оказаться не слишком опытным и сделать ошибки в программе) и бизнес-риском (поскольку задержка данного проекта может негативно повлиять на будущие деловые контакты между заказчиком и организацией-разработчиком).

Конкретные типы рисков, которые могут оказать влияние на данный проект, зависят от вида создаваемого программного продукта и от организационного окружения, где реализуется программный проект. Вместе с тем многие типы рисков способны повлиять на любые программные проекты, эти риски приведены в табл. 4.

Таблица 4 - Возможные риски программных проектов

Риск	Типы риска	Описание риска
Текучесть разработчиков	Риск для проекта	Опытные разработчики покидают проект до его завершения
Изменение в управлении организацией	Риск для проекта	Организация меняет свои приоритеты в управлении проектом
Неготовность аппаратных средств	Риск для проекта	Аппаратные средства, которые необходимы для проекта, не поступили вовремя или не готовы к эксплуатации
Изменение требований	Риск для проекта и для разрабатываемого продукта	Появление большого количества непредвиденных изменений в требованиях, предъявляемых к разрабатываемому ПО
Задержка в разработке спецификации	Риск для проекта и для разрабатываемого продукта	Спецификации основных интерфейсов подсистем не поступили к разработчикам в соответствии с графиком работ
Недооценка размера разрабатываемой системы	Риск для проекта и для разрабатываемого продукта	Размер системы значительно превысил первоначальную оценку
Недостаточная эффективность CASE-средств	Риск для разрабатываемого продукта	CASE-средства, предназначенные для поддержки проекта, оказались менее эффективными, чем ожидалось
Изменения в технологии разработки ПО	Бизнес-риск	Основные технологии построения программной системы заменяются новыми
Появление конкурирующего программного продукта	Бизнес-риск	На рынке программных продуктов до окончания проекта появилась конкурирующая программная система

Схема процесса управления рисками показана на рис. 6. Этот процесс состоит из четырех стадий.

Определение рисков. Определяются возможные риски для проекта, для разрабатываемого продукта и бизнес-риски.

Анализ рисков. Оценивается вероятность и последовательность появления рисков ситуаций.

Планирование рисков. Планируются мероприятия по предотвращению рисков или минимизации их воздействия на проект.

Мониторинг рисков. Постоянное оценивание вероятностей рисков и выполнение мероприятий по смягчению последствий проявления рисков ситуаций.



Рис. 6. Процесс управления рисками

Процесс управления рисками, как и другие процессы планирования, является итерационным, выполняемым в течение всего срока реализации проекта. Сначала разрабатываются планы управления рисками, затем постоянно отслеживается ситуация вокруг реализации проекта. При поступлении новой информации о возможных рисках заново проводится анализ рисков и первостепенное внимание уделяется новым рискам. По мере поступления новой информации также изменяются планы мероприятий по предотвращению и смягчению рисков.

Результаты процесса управления рисками документируются в виде планов управления рисками. Они должны включать описание возможных проектных рисков, их анализ и перечень мероприятий, необходимых для управления рисками.

Определение рисков

Определение рисков — первая стадия процесса управления рисками. На этой стадии описываются риски, которые могут проявиться при реализации проекта. В принципе на этой стадии не должна оцениваться вероятность и значимость рисков, но на практике маловероятные риски с незначительными последствиями обычно отбрасываются сразу.

Определение рисков может выполняться в режиме командной работы с использованием подхода "мозговой штурм" либо основываться на опыте менеджера. При определении рисков может помочь приведенный ниже список возможных категорий рисков.

Технологические риски. Проистекают из программных и аппаратных технологий, на основе которых разрабатывается система.

Риски, связанные с персоналом. Связаны с членами команды разработчиков.

Организационные риски. Проистекают из организационного окружения, в котором выполняется проект.

Инструментальные риски. Связаны с используемыми CASE-средствами и другими средствами поддержки процесса создания ПО.

Риски, связанные с системными требованиями. Проявляются при изменении требований, предъявляемых к разрабатываемой системе.

Риски оценивания. Связаны с оцениванием характеристик программной системы и ресурсов, необходимых для реализации проекта.

В табл. 5 представлены некоторые примеры, относящиеся к каждой из описанных категорий рисков. Результатом этапа определения рисков будет длинный список возможных рисков, которые могут повлиять на разрабатываемый программный продукт, проект или организацию-разработчика.

Таблица 5 - Категории рисков

Категория рисков	Примеры рисков
Технологические риски	База данных, которая используется в программной системе, не обеспечивает обработку ожидаемого объема транзакций. Программные компоненты, которые используются повторно, имеют дефекты, ограничивающие их функциональные возможности
Риски, связанные с персоналом	Невозможно подобрать работников с требуемым профессиональным уровнем. Ведущий разработчик заболел в самое критическое время. Невозможно организовать необходимое обучение персонала

Организационные риски	В организации, выполняющей разработку ПО, произошла реорганизация, в результате чего изменились приоритеты в управлении проектом. Финансовые затруднения в организации привели к уменьшению бюджета проекта
Инструментальные риски	Программный код, генерируемый CASE-средствами, не эффективен. CASE-средства невозможно интегрировать с другими средствами поддержки проекта
Риски, связанные с системными требованиями	Изменения требований приводят к значительным повторным темными требованиями работам по проектированию системы. Первоначальная нечеткая формулировка пользовательских требований привела к значительным изменениям системных требований, проявившихся на поздних стадиях разработки проекта
Риски оценивания	Недооценки времени выполнения проекта. Скорость выявления дефектов в системе ниже ранее запланированной. Размер системы значительно превышает первоначально рассчитанный

Анализ рисков

При анализе для каждого определенного риска подсчитывается вероятность его проявления и ущерб, который он может нанести. Не существует простых методов выполнения анализа рисков — в значительной мере он основан на мнении и опыте менеджера. Можно привести следующую шкалу вероятностей рисков и их последствий.

Вероятность риска считается очень низкой, если она имеет значение менее 10%; низкой, если ее значение от 10 до 25 %; средней при значениях от 25 до 50%; высокой, если значение колеблется от 50 до 75%; очень высокой при значениях более 75%.

Возможный ущерб от рискованных ситуаций можно подразделить на катастрофический, серьезный, терпимый и незначительный.

Результаты анализа рисков должны быть представлены в виде таблицы рисков, упорядоченных по степени возможного ущерба. В табл. 6 приведен упорядоченный список рисков, описанных в табл. 5; там же указаны вероятности этих рисков. Здесь вероятности рисков и степень ущерба от них указаны произвольно. На практике для их определения необходима подробная информация о проекте, технологии создания ПО, команде разработчиков и о самой организации.

Таблица 6 - Список рисков после проведения их анализа

Риск	Вероятность	Степень ущерба
Финансовые затруднения в организации привели к уменьшению бюджета проекта	Низкая	Катастрофическая
Невозможно подобрать работников с требуемым профессиональным уровнем	Высокая	Катастрофическая
Ведущий разработчик заболел в самое критическое время	Средняя	Серьезная
Программные компоненты, используемые повторно, имеют дефекты, ограничивающие их функциональные возможности	Средняя	Серьезная
Изменения требований приводят к значительным повторным работам по проектированию системы	Средняя	Серьезная
В организации, выполняющей разработку ПО, произошла реорганизация, в результате чего изменились приоритеты в управлении проектом	Высокая	Серьезная
База данных, которая используется в программной системе, не обеспечивает обработку ожидаемого объема транзакций	Средняя	Серьезная
Недооценки времени выполнения проекта	Высокая	Серьезная
CASE-средства невозможно интегрировать с другими средствами поддержки проекта	Высокая	Терпимая
Первоначальная нечеткая формулировка пользовательских требований привела к значительным изменениям системных требований, проявившихся на поздних стадиях разработки проекта	Средняя	Терпимая
Невозможно организовать необходимое обучение персонала	Средняя	Терпимая
Скорость выявления дефектов в системе ниже ранее спланированной	Средняя	Терпимая
Размер системы значительно превышает первоначально рассчитанный	Высокая	Терпимая
Программный код, генерируемый CASE-средствами, неэффективен	Средняя	Незначительная

Конечно, как вероятность рисков, так и возможный ущерб от них должны пересматриваться при поступлении дополнительной информации об этих рисках и по мере реализации мероприятий по управлению ими. Поэтому подобные таблицы рисков должны переделываться на каждой итерации процесса управления рисками.

После проведения анализа рисков определяются наиболее значимые риски, которые затем отслеживаются на протяжении всего срока выполнения проекта. Определение этих значимых рисков зависит от их вероятностей и возможного ущерба. В общем случае всегда отслеживаются риски с катастрофическими последствиями, а также риски с серьезным ущербом, значение вероятности которых выше среднего.

В некоторых статьях рекомендуется определить и отслеживать "10 верхних" рисков, но это не всегда обоснованная рекомендация. Количество рисков, которые необходимо отслеживать, зависит от конкретного проекта. Это может быть пять рисков, а может — пятнадцать. Но, конечно, количество рисков, по которым проводится мониторинг, должно быть обозримым. Большое количество отслеживаемых рисков потребует огромного количества собираемой информации. Из списка рисков, представленных в табл. 6, для мониторинга следует отобрать те риски, которые могут привести к катастрофическим и серьезным последствиям для вашего проекта.

Планирование рисков

Планирование заключается в определении стратегии управления каждым значимым риском, отобранным для мониторинга после анализа рисков. Здесь также не существует общепринятых подходов для разработки таких стратегий — многое основывается на "чутье" и опыте менеджера проекта. В табл. 7 показаны возможные стратегии управления основными рисками, приведенными в табл. 6.

Таблица 7 - Стратегии управления рисками

Риск	Стратегия
Финансовые проблемы организации	Подготовить краткий документ для руководства организации, показывающий важность данного проекта для достижения финансовых целей организации
Проблемы неквалифицированного персонала	Предупредить заказчика о потенциальных трудностях и возможной задержке проекта, рассмотреть вопрос о покупке компонентов системы
Болезни персонала	Реорганизовать работу команды разработчиков таким образом, чтобы обязанности и работа членов команды перекрывали друг друга, вследствие этого разработчики будут знать и понимать задачи, выполняемые другими сотрудниками
Дефектные системные компоненты	Заменить потенциально дефектные системные компоненты покупными компонентами, гарантирующими качество работы
Изменения требований	Попытаться определить требования, наиболее вероятно подверженные изменениям; в структуре системы не отображать детальную информацию
Реорганизация компании-разработчика	Подготовить краткий документ для руководства компании, показывающий важность данного проекта для достижения финансовых целей компании
Недостаточная производительность базы данных	Рассмотреть возможность покупки более производительной базы данных
Недооценки времени выполнения проекта	Рассмотреть вопрос о покупке системных компонентов, исследовать возможность использования генератора программного кода

Существует три категории стратегий управления рисками.

Стратегии предотвращения рисков. Согласно этим стратегиям следует проводить мероприятия, снижающие вероятность проявления рисков. Примером может служить стратегия исключения потенциально дефектных компонентов, описанная в табл. 7.

Минимизационные стратегии. Направлены на уменьшение возможного ущерба от рисков. Примером служит стратегия уменьшения ущерба от болезни членов команды разработчиков (см. табл. 7).

Планирование "аварийных" ситуаций. Согласно этим стратегиям необходимо иметь план мероприятий, которые следует выполнить в случае проявления рисков ситуации. В табл. 7 это стратегия поведения при возникновении финансовых проблем у организации-разработчика.

Мониторинг рисков

Мониторинг рисков заключается в регулярном пересчете вероятностей рисков и ущерба, который они могут нанести. Для этого необходимо постоянно отслеживать факторы, которые влияют на вероятность рисков и возможный ущерб. Эти факторы зависят от типов риска. В табл. 8 приведены признаки, которые помогают определить тип риска.

Таблица 8 - Признаки рисков

Тип риска	Признаки
Технологические риски	Задержки в поставке оборудования или программных средств поддержки процесса создания ПО, многочисленные документированные технологические проблемы
Риски, связанные с персоналом	Низкое моральное состояние персонала, натянутые отношения между членами команды разработчиков, низкое качество выполненной работы
Организационные риски	Разговоры среди персонала о пассивности и недостаточной компетентности высшего руководства организации
Инструментальные риски	Нежелание разработчиков использовать программные средства поддержки, неодобрительные отзывы о CASE-средствах, запросы на более мощные инструментальные средства
Риски, связанные с системными требованиями	Необходимость пересмотра многих системных требований, недовольство заказчика ПО
Риски оценивания	Изменения графика работ, многочисленные отчеты о нарушении графика работ

Мониторинг рисков должен быть непрерывным процессом, отслеживающим ход выполнения мероприятий по управлению рисками, при этом каждый основной риск должен рассматриваться отдельно.

4. Порядок выполнения работы

Изучить предлагаемый теоретический материал.

Построить временную и сетевую диаграммы для выбранного проекта.

Построить диаграмму распределения участников группы по этапам.

Построить список возможных рисков с указанием названия риска, его описание и типа.

Провести анализ рисков.

Описать стратегию планирования рисков.

Построить отчет, включающий все полученные диаграммы и описание стратегии планирования рисков.

Лабораторная работа №2 Моделирование бизнес-процессов

Цель работы: Освоение основных концепций моделирования с использованием BPMN.

Задание: Используя BizAgi Process Modeler, постройте диаграммы на русском языке.

Упражнение 1. Продажа аппаратуры: подготовка товара для пересылки.

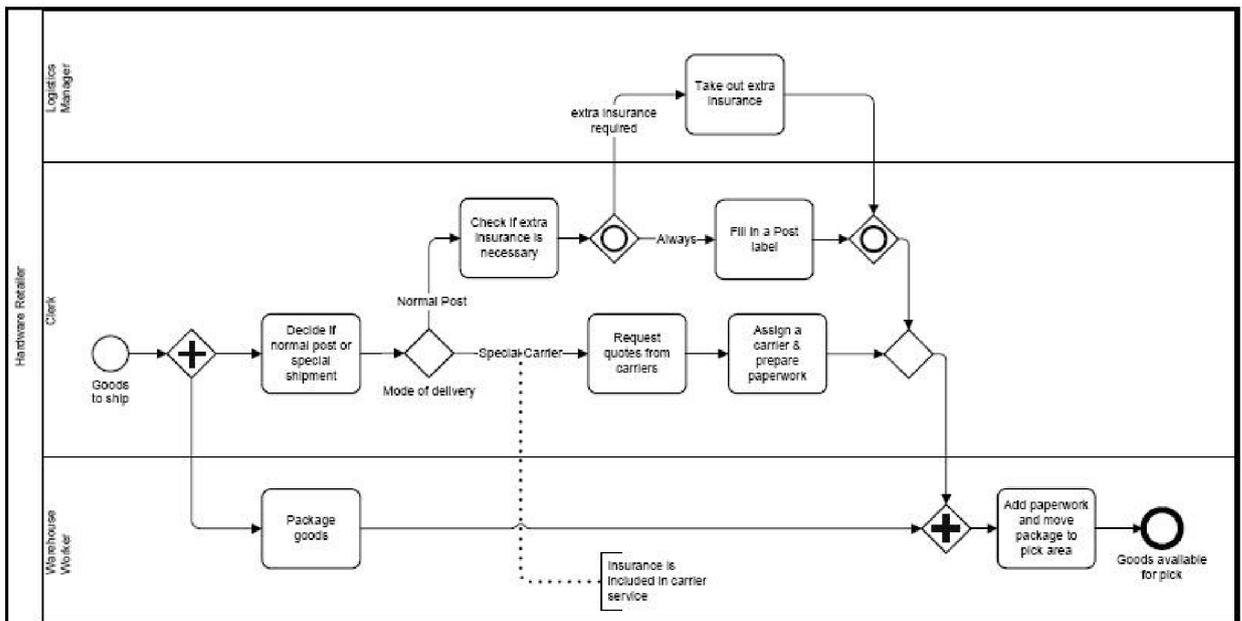


Figure 5.1: Shipment Process of a hardware retailer

На данном рисунке показаны подготовительные шаги, которые должны быть выполнены торговой организацией перед тем, как заказанный товар будет готов к пересылке покупателю.

В данном процессе используется только один пул и только три линии для участников процесса: менеджера логистики, клерка и кладовщика. При этом мы не рассматриваем явно коммуникацию между участниками.

Простое стартовое событие «Goods to ship» указывает начало процесса. Сразу после инициализации процесса параллельный шлюз начинает две параллельных операции: пока клерк решает какой тип пересылки применить (обычный или специальный), кладовщик начинает упаковку товара. Задача клерка приводит к исключаяющему шлюзу «mode of delivery», что является хорошим примером рекомендуемого использования шлюзов: шлюз не является ответственным за принятие решения о типе пересылки, - это решение принимается во время предыдущей активности. Шлюз же выполняет только функцию маршрутизации, перенаправляя поток по альтернативным путям в зависимости от результатов предыдущих процессов

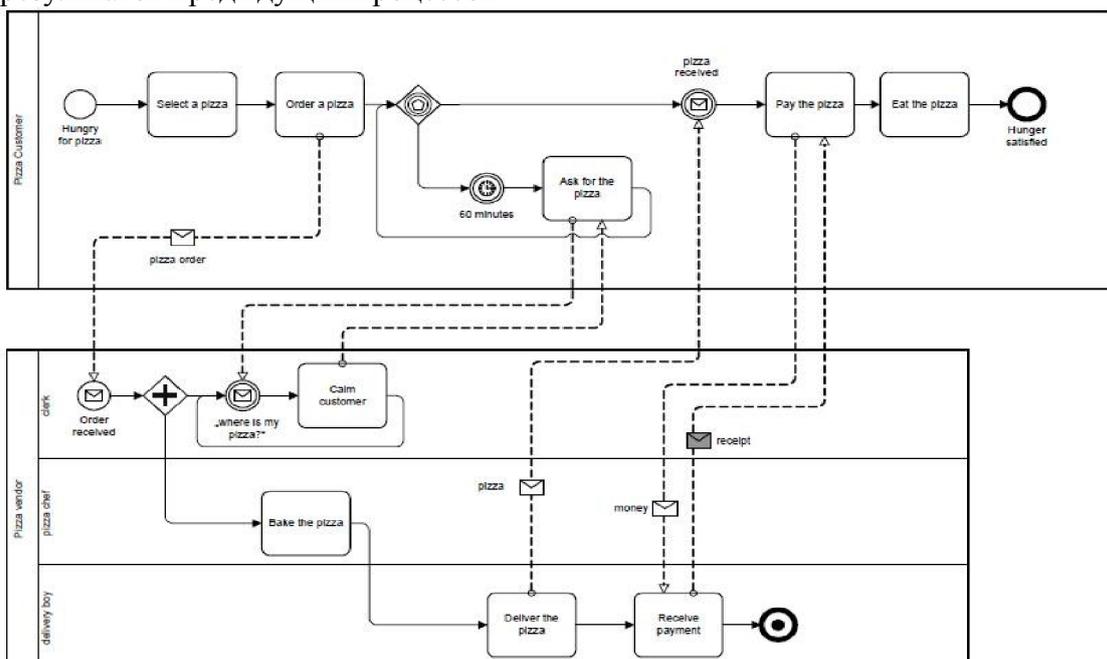


Figure 5.2: Ordering and delivering pizza

Упражнение 3. Выполнение заказа и дополнительные приобретения.

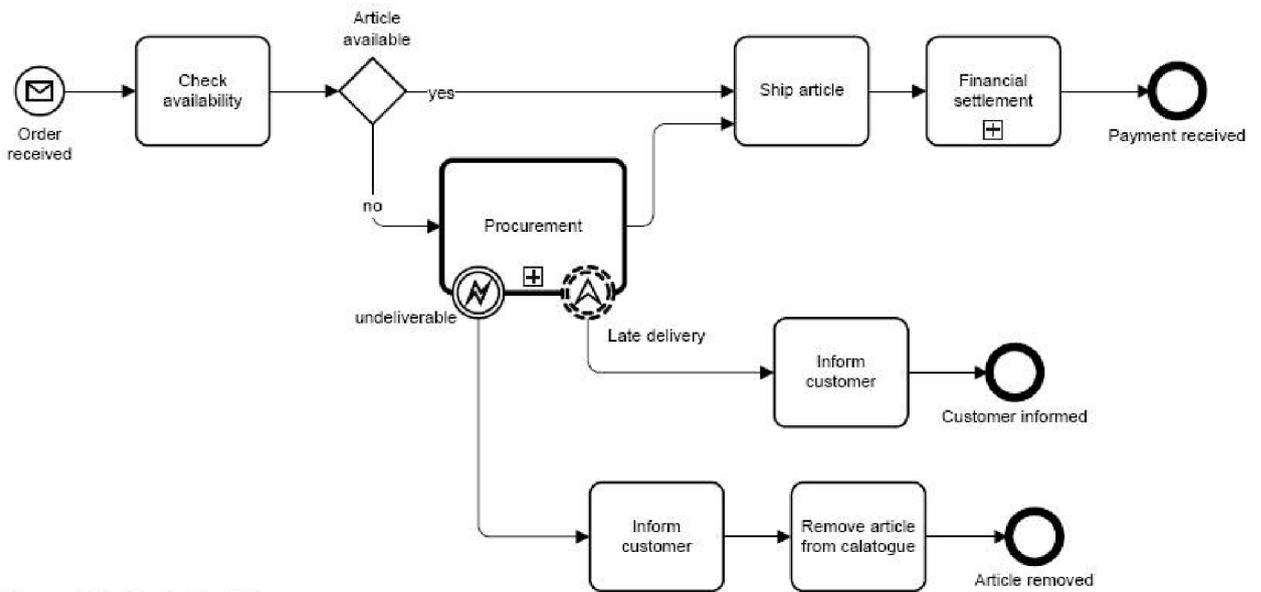


Figure 5.3: Order Fulfillment

В данном упражнении рассматривается процесс выполнения заказа: если указанная в заказе статья имеется в наличии, то она отправляется заказчику с последующими финансовыми процедурами, которые на рисунке ниже свернуты и скрыты как под-процесс. Если же статьи такого типа нет в наличии, то запускается под-процесс приобретения. Обратите внимание на широкую границу этого под-процесса на диаграмме, что означает активизацию внешнего вызова. Это - как обозначение глобально выполняемой задачи, т. е. под-процесса. Другой характерной особенностью под-процесса дополнительного приобретения статей является наличие двух связанных с ним событий. Используя эти присоединенные события можно управлять любыми событиями, возникающими спонтанно при выполнении под-процесса.

Таким образом, мы различаем прерывающие и не-прерывающие присоединенные события. Оба события перехватывают и обрабатывают возникающие в под-процессе события, но только непрерывающее (в нашем случае «late delivery») не завершает активность к которой оно присоединено. При возникновении прерывающего события текущая активность немедленно завершается.

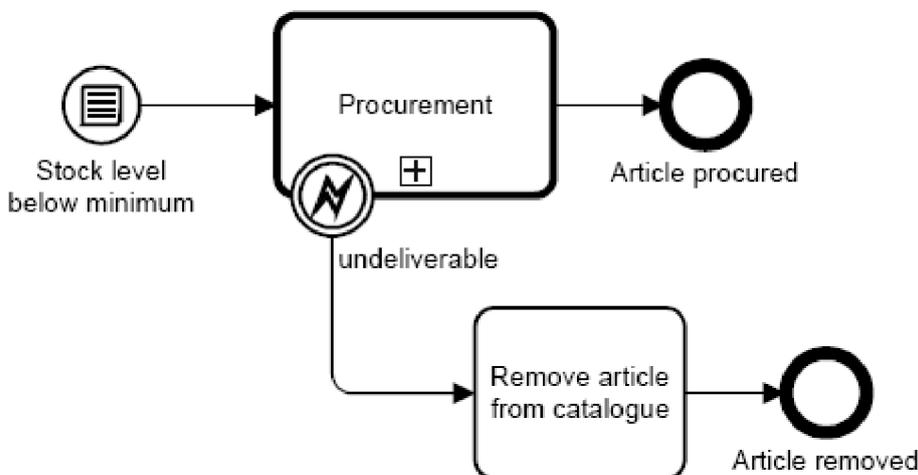


Figure 5.4: Stock maintenance process

Процесс управления складом на этом рисунке инициируется условным стартовым событием, т.е. когда возникает ситуация со значением условия TRUE, например, в нашем случае, когда запасы на складе оказываются ниже допустимого минимума. Для пополнения запасов должен быть запущен процесс дополнительного приобретения (procurement). Таким образом здесь мы используем тот же процесс дополнительного приобретения, что и при обработке заказов, и мы просто вызываем под-процесс Procurement, что обозначается жирной рамкой. Также как в процессе выполнения заказа, этот процесс обрабатывает исключительную ситуацию удалением статьи из каталога. Но в данном процессе управления складом нет необходимости в обработке события «late delivery» (задержка поставки), которое остается необработанным. Если процесс пополнения запасов завершается успешно, уровень запасов выше минимального, то процесс управления складом завершается событием «article procured» (статья пополнена). Теперь заглянем внутрь под-процесса пополнения.

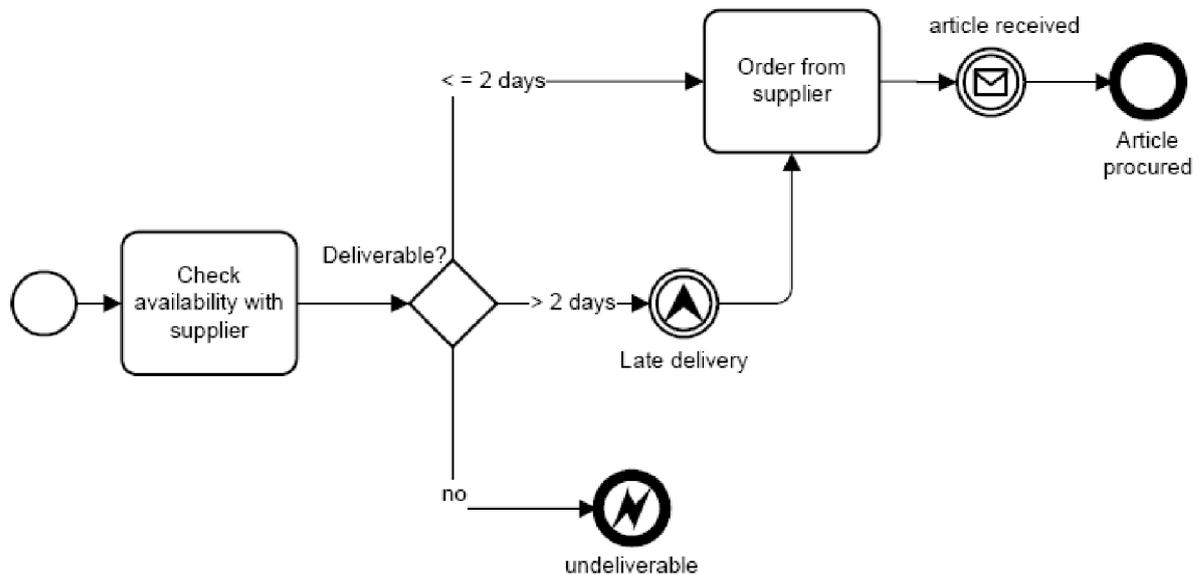


Figure 5.5: Procurement sub-process

Поскольку это под-процесс, то стартовое событие простое, что означает отсутствие какого-либо триггера от внешнего события. Просто внешний процесс ссылается на этот подпроцесс. Первой задачей этого под-процесса является проверка наличия пополняемой статьи у поставщика. Если нет, то под-процесс выбрасывает исключение «not deliverable», которое перехватывается как при обработке заказа, так и при пополнении склада. При задержки поставки на 2 дня под-процесс вырабатывает событие сообщаящее основному процессу о задержке поставки. Такое (escalation event) событие имеет escalationCode необходимый для связывания выброса и перехвата. В отличие от ошибки при возникновении такого события текущий поток выполнения никогда не прерывается и на него не влияют промежуточные escalation события. Далее, процесс пополнения продолжается ожидая поставки. Но вырабатываемые события обрабатываются ближайшим родительским процессом с присоединенным кодом escalationCode, совпадающим с кодом этого события. В процессе обработки заказа событие «late delivery» присоединенное к под-процессу «Procurement» перехватывает выбрасываемое событие «late delivery», но теперь это не-прерывающее событие. Поэтому создается новый образец, соответствующий произошедшим событиям и предназначенный для уведомления попупателя о том, что заказанная статья будет выслана позже.

По завершении под процесса пополнения, процесс обработки заказа продолжается отправкой статьи и выполнением финансовых операций.

Лабораторная работа №3 Ознакомление с CASE-средством RATIONAL ROSE

Цель работы. Изучить интерфейс Rational Rose и принципы работы с ним.

Основные понятия

Rational Rose – это CASE-средство фирмы Rational Software Corporation (США), предназначенное для автоматизации этапов анализа и проектирования программного обеспечения, для генерации кодов на различных языках и выпуска проектной документации. Rational Rose использует методологию объектно-ориентированного анализа и проектирования, основанную на подходах трех ведущих специалистов в данной области: Буча, Рамбо и Джекобсона. Разработанная ими универсальная нотация для моделирования объектов (UML – Unified Modeling Language) претендует на роль стандарта в области объектно-ориентированного анализа и проектирования. Конкретный вариант Rational Rose определяется языком, на котором генерируются коды программ (C++, Smalltalk, PowerBuilder, Ada, SQLWindows и ObjectPro). Основной вариант – Rational Rose/C++ – позволяет разрабатывать проектную документацию в виде диаграмм и спецификаций, а также генерировать программные коды на C++. Кроме того, Rational Rose содержит средства реинжиниринга программ, обеспечивающие повторное использование программных компонент в новых проектах.

Структура и функции

В основе работы Rational Rose лежит построение различного рода диаграмм и спецификаций, определяющих логическую и физическую структуры модели, ее статические и динамические аспекты. В их число входят диаграммы классов, состояний, сценариев, модулей, процессов. В составе Rational Rose можно выделить 6 основных структурных компонент:

- репозиторий,
- графический интерфейс пользователя,
- средства просмотра проекта (browser),
- средства контроля проекта,
- средства сбора статистики
- генератор документов.

К ним добавляются генератор кодов (индивидуальный для каждого языка) и анализатор для C++, обеспечивающий реинжиниринг – восстановление модели проекта по исходным текстам программ. Репозиторий представляет собой объектно-ориентированную базу данных.

Средства просмотра обеспечивают "навигацию" по проекту, в том числе: перемещение по иерархиям классов и подсистем, переключение от одного вида диаграмм к другому и т. д. Средства контроля и сбора статистики дают возможность находить и устранять ошибки по мереразвития проекта, а не после завершения его описания. Генератор отчетов формирует тексты выходных документов на основе содержащейся в репозитории информации. Средства автоматической генерации кодов программ на языке C++, используя информацию, содержащуюся в логической и физической моделях проекта, формируют файлы заголовков и файлы описаний классов и объектов. Создаваемый таким образом скелет программы может быть уточнен путем прямого программирования на языке C++. Анализатор кодов C++ реализован в виде отдельного программного модуля. Его назначение состоит в том, чтобы создавать модули проектов в форме Rational Rose на основе информации, содержащейся в определяемых пользователем

исходных текстах на C++. В процессе работы анализатор осуществляет контроль правильности исходных текстов и диагностику ошибок. Модель, полученная в результате его работы, может целиком или фрагментарно использоваться в различных проектах. Анализатор обладает широкими возможностями настройки по входу и выходу. Например, можно определить типы исходных файлов, базовый компилятор, задать, какая информация должна быть включена в формируемую модель и какие элементы выходной модели следует выводить на экран.

Таким образом, Rational Rose/C++ обеспечивает возможность повторного использования программных компонент.

В результате разработки проекта с помощью CASE-средства Rational Rose формируются следующие документы:

- диаграммы классов;
- диаграммы состояний;
- диаграммы сценариев;
- диаграммы модулей;
- диаграммы процессов;
- спецификации классов, объектов, атрибутов и операций;
- заготовки текстов программ;
- модель разрабатываемой программной системы.

Последний из перечисленных документов является текстовым файлом, содержащим всю необходимую информацию о проекте (в том числе необходимую для получения всех диаграмм и спецификаций).

Взаимодействие с другими средствами и организация групповой работы

Rational Rose интегрируется со средством PVCS для организации групповой работы и управления проектом и со средством SoDA – для документирования проектов. Интеграция Rational Rose и SoDA обеспечивается средствами SoDA. Для организации групповой работы в Rational Rose возможно разбиение модели на управляемые подмодели. Каждая из них независимо сохраняется на диске или загружается в модель. В качестве подмодели может выступать категория классов или подсистема. Для управляемой подмодели предусмотрены операции:

- загрузка подмодели в память;
- выгрузка подмодели из памяти;
- сохранение подмодели на диске в виде отдельного файла;
- установка защиты от модификации;
- замена подмодели в памяти на новую.

Наиболее эффективно групповая работа организуется при интеграции Rational Rose со специальными средствами управления конфигурацией и контроля версий (PVCS). В этом случае защита от модификации устанавливается на все управляемые подмодели, кроме тех, которые выделены конкретному разработчику. В этом случае признак защиты от записи устанавливается для файлов, которые содержат подмодели, поэтому при считывании "чужих" подмоделей защита их от модификации сохраняется и случайные воздействия окажутся невозможными.

Среда функционирования

Rational Rose функционирует на различных платформах: IBM PC (в среде Windows), Sun SPARC stations (UNIX, Solaris, SunOS), Hewlett-Packard (HP UX), IBM RS/6000 (AIX). Для работы системы необходимо выполнение следующих требований:

Платформа Windows – процессор 80386SX или выше (рекомендуется 80486), память 8 Мб (рекомендуется 12 Мб), пространство на диске 8Мб + 1–3 Мб для одной модели.

Платформа UNIX – память 32 + (16* число пользователей) Мб, пространство на диске 30 Мб + 20 при инсталляции + 1–3 Мб для одной модели. Совместимость по версиям обеспечивается на уровне моделей.

Технология выполнения работы

Запуск программы

1. Вызвать кнопкой Пуск Главное меню.
2. Найти в программах Rational Rose Enterprise Edition и выбрать Rational Rose Enterprise Edition
3. Запустить программу.
4. Программа загрузится и появится окно с набором стандартных проектов. Нажать на Cancel.

CASE – средство Rational Rose имеет простой и понятный пользовательский интерфейс для построения требуемых логических и физических моделей данных. Он зависит от используемой технологии. В любом случае при запуске средства моделирования появляются:

- меню;
- основная панель инструментов;
- панель специальных инструментов;

– навигатор моделей.

Окно пакета при запуске

Основная панель инструментов содержит следующие главные кнопки:

- создание новой модели;
- открытие имеющейся модели;
- сохранение построенной модели;
- копирование модели;
- печать модели;
- масштабирование.

Навигатор модели показывает состав модели по уровням разработки. С его помощью можно легко и быстро переходить от одной модели к другой. Работа с навигатором модели аналогична работе с проводником системы Windows. Навигатор поддерживает четыре

представления:

- использования;
- логическое;
- компонентов;
- размещения.

Панель специальных инструментов содержит основные кнопки для создания выбранной диаграммы, например для построения диаграммы прецедентов представления использования:

- создание субъекта;
- создание аспекта;
- создание ассоциации субъектов и аспектов;
- создание обобщения;

Для логического представления:

- создание класса;
- создание ассоциации классов.

Окно модели является местом создания логической или физической модели данных исследуемой системы.

Задания

1. Запустить Rational Rose.
2. Посмотреть навигацию по проекту.
3. Создать любой элемент, дать ему название и комментарий к нему.
4. Сохранить проект.

Лабораторная работа №4 Создание модели вариантов использования

Цель работы: ознакомиться с созданием функциональной модели использования; изучить нотации, применяемые при построении диаграмм, и освоить их применение в процессе постановки задачи.

Основные понятия

Моделирование в Ration Rose проводится как спуск от концептуальной модели к логической, а затем к физической модели программной системы. Концептуальная модель выражается в виде диаграмм вариантов использования (Use – case diagram). Этот тип диаграмм служит для проведения итерационного цикла общей постановки задачи вместе с заказчиком.

Вариант использования представляет собой последовательность действий, выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом). Вариант использования описывает типичное взаимодействие между пользователем и системой. В простейшем случае вариант использования определяется в процессе обсуждения с пользователем тех функций, которые он хотел реализовать. Эти диаграммы служат основой для достижения взаимопонимания между программистами-профессионалами, разрабатывающими проект, и заказчиками проекта. Внутри каждого варианта использования (прецедента) могут быть определены:

- вложенная диаграмма использования,
- диаграмма взаимодействия объектов,
- диаграмма последовательности взаимодействия,
- диаграмма классов,
- диаграмма перехода состояния.

Действующее лицо (Actor) – это роль, которую пользователь играет

по отношению к системе. Действующие лица представляют собой роли, а не конкретных людей или наименования работ. Действующее лицо

может быть внешней системой, которой необходима информация от данной системы. На рис. 2 приводится вариант использования, описывающий одну из функций системы управления проектами – обратную связь между менеджером проекта и исполнителем. Нотации представления использования (диаграмма прецедентов) Каждое представление строится из диаграмм, которые используют свои нотации (обозначения). Для представления использования применяются следующие нотации:

- субъект как внешняя сущность, взаимодействующая с системой; им может быть и человек, и устройство, и другая система;
- аспект использования как определенное средство, предоставляемое системой;
- односторонняя ассоциация, как взаимодействие, направленное от одного субъекта или аспекта к другим;
- обобщение от одного субъекта или аспекта к другому;

Примеры обобщения показаны на рис. 1. Это сильный инструмент построения диаграмм. Так, один клиент, другой клиент обслуживающей фирмы обобщаются в клиента фирмы. Пример. Менеджер модифицирует план, назначает ресурс и получает отчеты от исполнителей, сотрудников и субподрядчиков проекта. Информационную систему назовем "Управление проектами". На рис. 2 показаны функции менеджера относительно выполнения проекта.

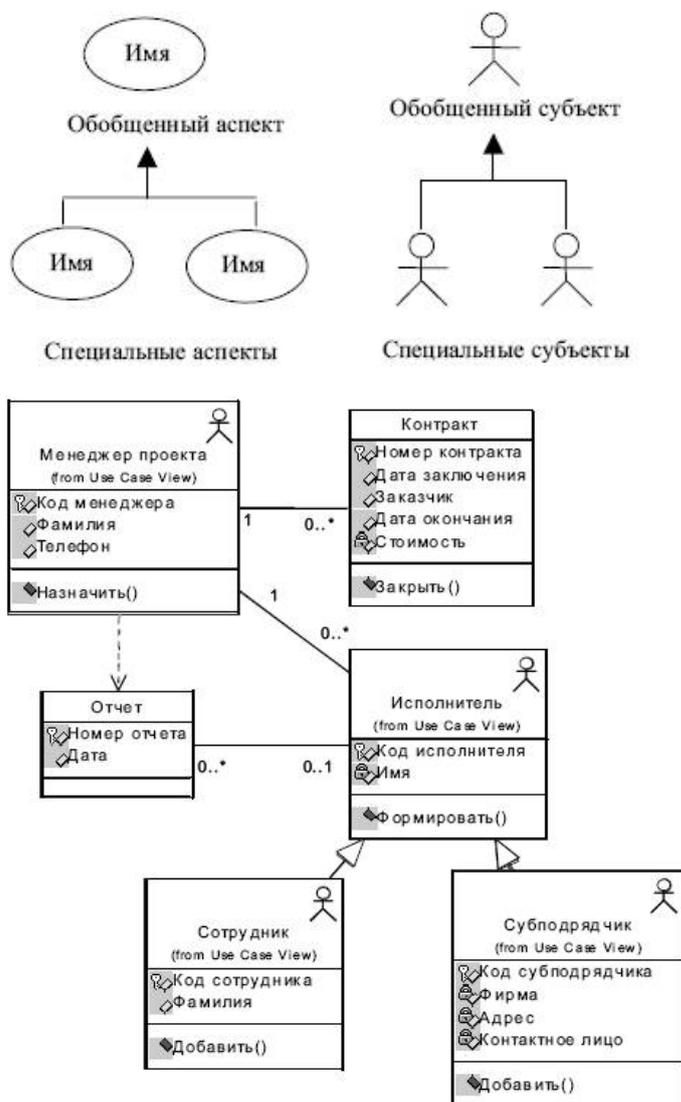


Рис. 2. Диаграмма использования. Управление проектами

Создание ассоциации

- a. Нажать кнопку создания ассоциации.
- b. Нарисовать стрелку от одного элемента диаграммы прецедентов к другому.
- c. Отрегулировать размещение элементов диаграммы прецедентов.

Задания

для построения представления варианта использования:

Построить диаграмму прецедентов по разработанному техническому заданию.

1. Присвоить имя диаграмме согласно предметной области и решаемой задаче.
2. Определить субъектов (актеров) и прецедентов и присвоить им имена согласно предметной области.
3. Определить ассоциации между ними.
4. Построить обобщения между субъектами и прецедентами

Лабораторная работа №5 Календарное планирование проектов при помощи MS Project

Содержание работы

Общее знакомство с интерфейсом и предварительные настройки

Открытие нового проекта

Настройка параметров проекта

Формирование иерархии и взаимосвязи задач проекта

Назначение ресурсов задачам

Построение сетевого графика с выделением критического пути и вывод основных временных характеристик работ

Построение диаграммы Ганта с выделением критического пути и вывод основных временных характеристик работ

Формирование базового проекта

Порядок выполнения работы

Общее знакомство с интерфейсом и предварительные настройки

Запускаем Project и рассмотрим окно программы. Окно программы имеет стандартный вид, как и у всех продуктов фирмы Microsoft. В самом верху окна – строка меню и панели инструментов, ниже – строка ввода, рабочая область программы и строка состояния.

Рабочая область MS Project содержит Область задач (крайнее слева), Панель представлений (в середине) и собственно поле, где отображаются проектные данные (крайнее справа). Если Область задач или Панель представлений закрыты, их необходимо открыть, выполнив, соответственно, команды

Вид →Панели инструментов →Область задач

Вид → Панель представлений

Необходимо настроить MS Project таким образом, чтобы при открытии программы отображались панели Область задач, Консультант и Панель представлений (рис.5).

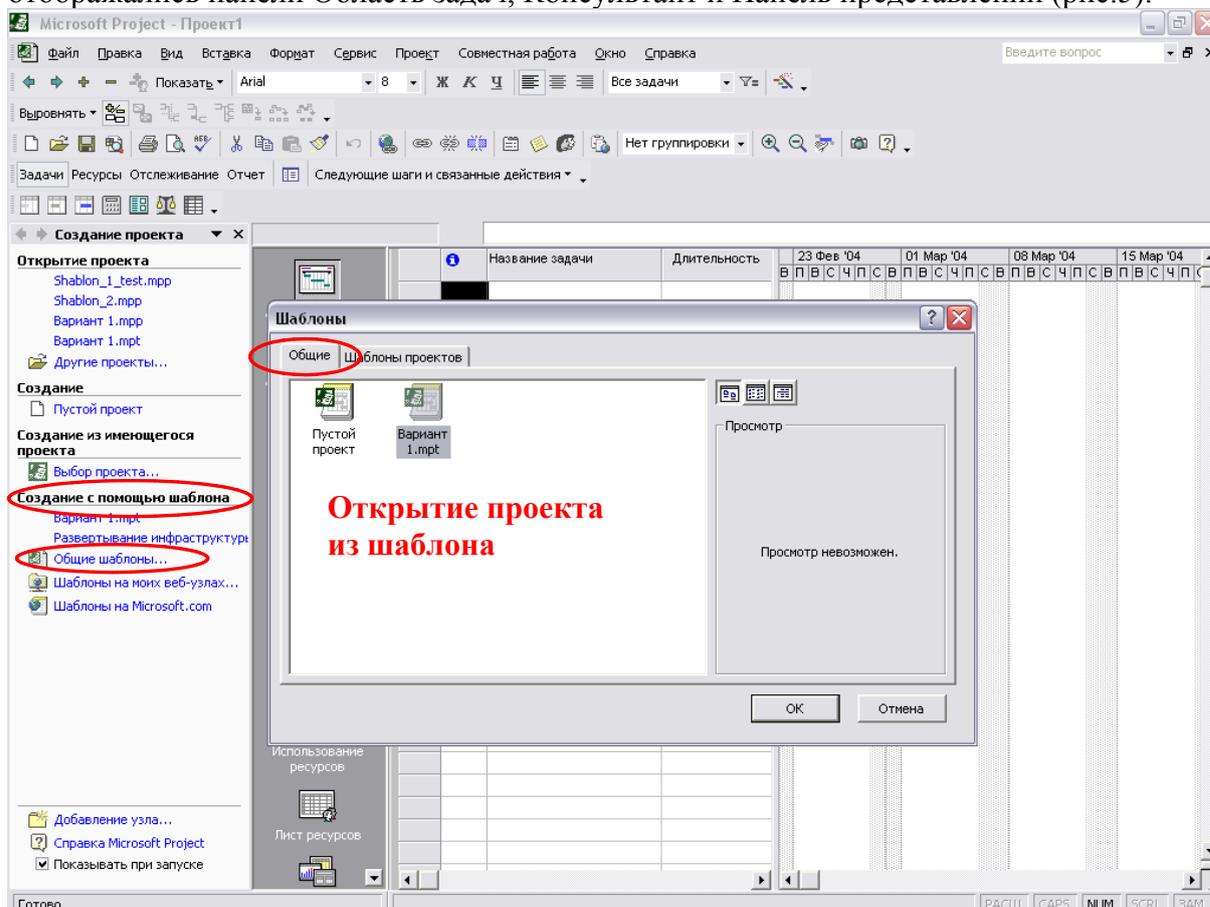


Рисунок 5. Пользовательский интерфейс MS Project.

Открытие нового проекта, используя шаблон заданного варианта задания

В Области задач в подразделе Создание с помощью шаблона необходимо открыть Общие шаблоны... и в открывшемся окне на вкладке Общие открыть шаблон задания, выданного преподавателем (рис. 5).

Необходимо проверить правильность установки основных сведений о проекте.

Выполняем команды Проект→Сведения о проекте

В открывшемся окне должны быть следующие установки (если нет, то установить именно такие):

Дата начала – определяется в Рекомендации для составления сетевого плана проекта (в задании)

Дата окончания - не активна

Планирование – от даты начала проекта

Текущая дата – сегодняшняя дата

Дата отчета – НД (то есть не определена)

Календарь – стандартный (то есть 8-часовой рабочий день, суббота и воскресенье – выходные)

Приоритет – 500

Когда установки соответствуют требуемым, принимаем их, нажав ОК.

Настройка параметров проекта

Произведем настройки MS Project. Выберем команду меню

Сервис→ Параметры и будем работать в открывшемся диалоговом окне с восьмью вкладками (рис.6).

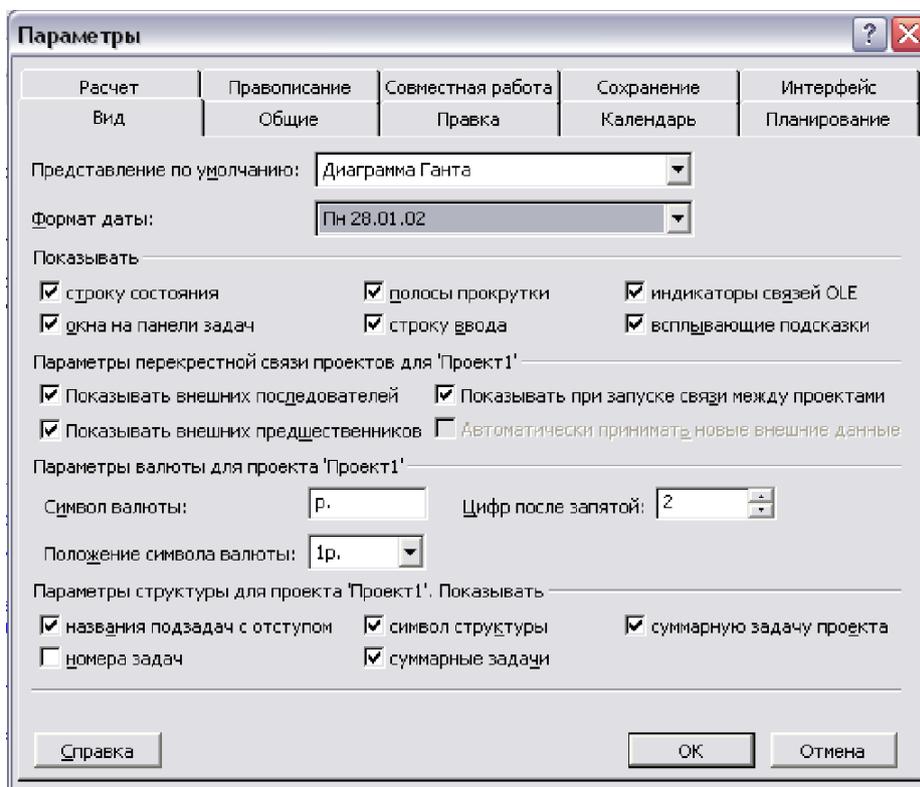


Рисунок 6. Панель настройки параметров проекта.

Вкладка Общие

Вводим имя пользователя (свое) и сбрасываем флажки Открывать последний файл при запуске, Включать автофильтр для новых проектов.

Все остальные чекбоксы должны быть помечены, ставки работ вводить не надо, а количество запоминаемых файлов пусть будет максимально и равно 9.

Вкладка Правка

Все чекбоксы должны быть помечены, с предлагаемыми цветами гиперссылок и параметрами отображения временных интервалов можно согласиться по умолчанию.

Вкладка Календарь

По умолчанию предлагаются параметры стандартной 40-часовой рабочей недели с 8 часовым рабочим днем с 9 до 18 часов. С этим предлагаемым вариантом тоже соглашаемся.

Вкладка Планирование

Здесь необходимо обязательно сбросить флажок Автоматическое связывание вставленных или перемещенных задач, так как в противном случае все задачи любого списка будут рассматриваться как последовательные. Остальные флажки нужно оставить, а в окошках должны быть следующие надписи

Новые задачи – начинаются в день начала проекта

Длительность – в днях

Трудозатраты – в днях

Тип задач по умолчанию – фиксированный объем ресурсов

Вкладка Интерфейс

Здесь все прямоугольные чекбоксы должны быть помечены, а для круглых чекбоксов выбираем варианты Стандартная функция Microsoft Project и Стандартное содержимое Microsoft Project.

Вкладка Сохранение

Здесь для Сохранять файлы Microsoft Project как надо выбирать вариант.

Проект (*.mpr), расположение файлов оставить по умолчанию.

Также надо пометить чекбокс Автосохранение, задав значение этого параметра в 10 минут, и пометить чекбокс Запрос перед сохранением. Из оставшихся чекбоксов необходимо отметить только Сохранять только активный проект.

Вкладка Совместная работа

Пропускается, так как эта опция не используется.

Вкладка Правописание

Необходимо пометить все чекбоксы и оставить все неизменным в Поле для проверки.

Вкладка Расчет

В разделе Параметры расчета для Microsoft Project должны быть выбраны варианты Режим расчета – автоматический и Расчет – в активном проекте. Остальные чекбоксы (за исключением двух нижних) оставляем заполненными по умолчанию, нижние же должны быть следующими: Рассчитывать насколько критических путей - не помечается и Считать критическими задачи, имеющие резерв 0 дней.

Вкладка Вид

Должны быть произведены следующие установки:

Представление по умолчанию – диаграмма Ганта

Формат даты – по умолчанию

Подраздел Показывать – отмечать все чекбоксы

Подраздел Параметры перекрестной связи проектов - отмечать все чекбоксы

Символ валюты – р

Положение символа валюты – 1р.

Цифр после запятой – 2

Подраздел Параметры структуры для проекта - отмечать все чекбоксы, кроме Показывать Номер задач.

Присвоение индивидуальных данных проекту

Произвести запись в задаче плана проекта с «0»-ым номером по образцу: «Вариант 2 - гр.7777 - Иванов М, Петров П., Сидоров П.». Результаты работ принимаются преподавателем только при наличии этих данных.

Формирование иерархии и взаимосвязи задач нового проекта

Производится по шаблону заданию и исходным данным по заданному преподавателем варианту

Определение необходимого типа каждой задачи (в шаблоне по умолчанию все задачи представлены, как простые)

Выделение задачи, как суммарной

Необходимо выделить задачу в столбце Название задачи, лежащую ниже суммарной и ближайшую к ней, которая будет входить в суммарную и воспользоваться стрелкой На уровень ниже на панели инструментов. Требуемая задача будет выделена как суммарная (полужирный шрифт и специальный символ), а входящая в нее сдвинется в ячейке вправо от своего прежнего положения. Аналогичную процедуру проделать для всех остальных задач, которые должны входить в требуемую суммарную.

Выделение задачи как события или вехи.

Необходимо выделить задачу в столбце Название задачи. Произвести действия

Проект→Сведения о задаче→Дополнительно и пометить чекбокс Пометить задачу как веху и выставить длительность задачи, равную 0 дней.

Формирование связи между задачами

Произведем действия Проект→Сведения о задаче→Предшественники. На основании информации Рекомендаций для составления сетевого плана проекта (в задании) и информации п.2.1 в столбце Предшественники необходимо задать задачу-предшественника, а в столбце Тип – выбрать необходимый тип связи. Суммарные задачи связывать не надо.

Ввод длительности задач

Произведем действия Проект→Сведения о задаче→Общие и в окне Длительность необходимо ввести длительность задачи в днях и на вкладке Дополнительно снять флажок в чекбоксе Фиксированный объем работ

Важно помнить, что для суммарных задач длительности определяются по составным (вводить их длительность не надо), а длительности событий (вех) - нулевые

Назначение задачам имеющегося ресурса

Назначение ресурса осуществляется для каждой простой задачи.

В таблице, с которой производится работа, необходимо пометить задачу и на панели инструментов нажать кнопку Назначить ресурсы.

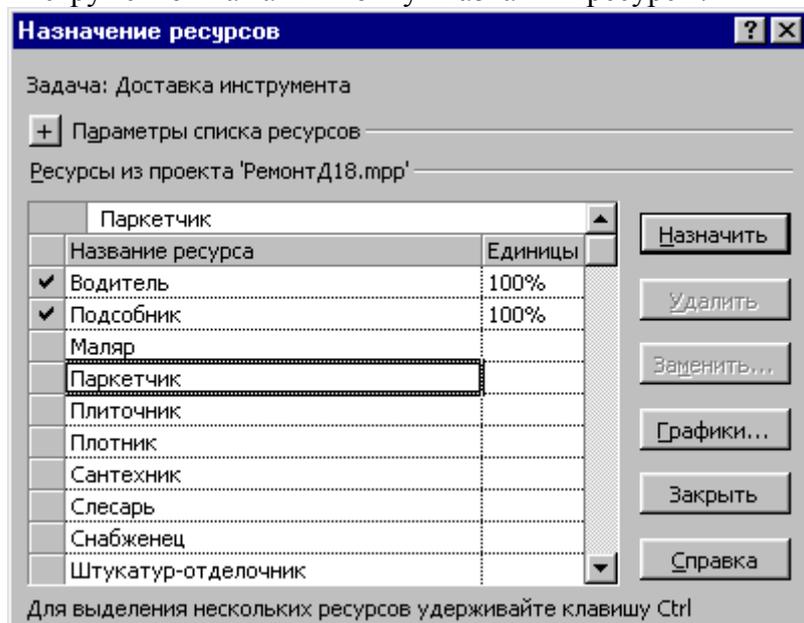


Рисунок 7. Панель “Назначение ресурсов”.

В открывшемся окне Назначение ресурсов (рис.7) в соответствии с информацией таблицы задания выделить требуемый ресурс и в столбце Единицы выбрать 100%. Далее нажать последовательно кнопки Назначить и далее Закрывать. Для вех и суммарных задач ресурсы не назначаются.

Добавление столбцов временных характеристик каждой задачи проекта

Щелкаем правой кнопкой мыши по столбцу, перед которым необходимо вставить дополнительный столбец и в появившемся контекстном меню выбираем вариант Вставить столбец. В открывшемся окне в графе Имя поля выбираем соответственно Общий временной резерв (рис.8). Для определенности предлагается вставить столбец Общий временной резерв после столбца Окончание

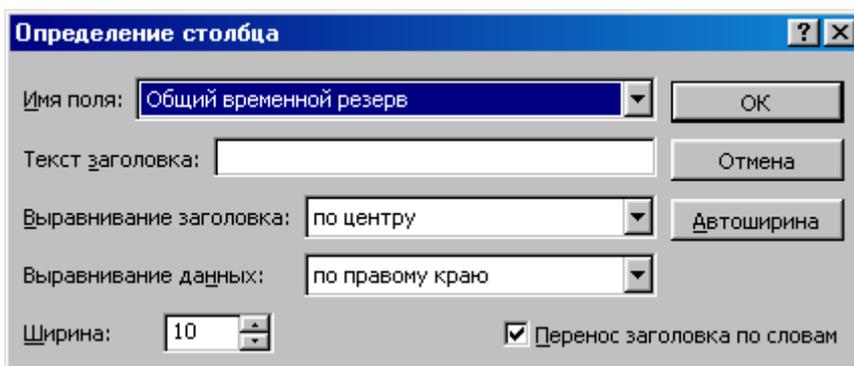


Рисунок 8. Панель добавления столбца.

Аналогичным образом вставляем столбцы Раннее начало, Раннее окончание, Позднее начало и Позднее окончание и нажимаем ОК. Для определенности предлагается вставить столбец в следующем порядке : Раннее начало, Позднее начало, Раннее окончание, Позднее окончание после столбца Общий временной резерв.

Сохранение файла проекта, как базового

Производим действия Сервис→Отслеживание→Сохранить базовый план.

В открывшемся окне Сохранение базового плана помечаем чекбокс Сохранить базовый план и чекбокс Всего проекта. Далее нажимаем ОК. План сохранится как базовый

Построение сетевого графика с выделением критического пути и вывод основных временных характеристик работ

Переходим от представления Вид→ диаграмма Ганта к Вид→ Сетевой график при этом результаты подготовленной таблицы представляются в виде сетевого графика типа “работа-вершина” и на графике красным цветом оказываются выделены критические задачи (работы). В каждом блоке, соответствующем задаче имеется информация о ней. Для сохранения для дальнейшего анализа сетевого графика в виде файла с последующей его распечаткой необходимо произвести следующие действия:

Со строки меню открываем

Вид → Панели инструментов → Сетевой график

и в открывшейся строке инструментов нажимаем кнопки Убрать суммарные задачи (чтобы не загромождать график) и Показать надписи связей.

Уменьшаем масштаб сетевого графика, чтобы он весь разместился на экране монитора (продельываем действия Вид→Масштаб и выбираем масштаб примерно 50%).

Чтобы максимально использовать площадь экрана можно закрыть Панель представлений и Область задач, выполнив уже известные действия

Вид →Панели инструментов →Область задач

Вид →Панель представлений

На панели инструментов нажимаем кнопку Копировать рисунок и из открывшейся панели выбираем вариант Для печати. Должны быть отмечены чекбоксы В Копировать – Отображаемые строки и в Шкала времени – Как на экране. По нажатии ОК, график скопируется в буфер.

Открываем в Microsoft Word пустой документ в альбомном формате и скопируем туда из буфера сетевой график. С ним можно теперь обращаться как с рисунком, растягивать, увеличивать для придания размеров, удобных для анализа и для последующей печати. Сохраняем результаты работы в виде *.doc файла на своей дискете.

Построение диаграммы Ганта с выделением критического пути и вывод основных временных характеристик работ

Переходим от представления Вид→ Сетевой график к

Вид→ Диаграмма Ганта с отслеживанием. При этом красным цветом производится выделение критических задач. Для сохранения для дальнейшего анализа сохраняем таблицу и диаграмму, произведя следующие действия:

На панели инструментов нажимаем кнопку Копировать рисунок и из открывшейся панели выбираем вариант Для печати. Должны быть отмечены чекбоксы в Копировать – Отображаемые строки и в Шкала времени – Как на экране. По нажатии ОК, таблица и диаграмма копируется в буфер. Для того, чтобы на сохраняемом рисунке были видны необходимые столбцы с ранними/поздними временами начала/окончания и временным резервом перед копированием можно скрыть лишние столбцы таблицы, кликнув мышью по заголовку скрываемого столбца и из открываемого правой кнопкой мыши контекстного меню выбрать Скрыть столбец. Также можно при помощи мыши сдвигать вертикальный разделитель между таблицей и диаграммой Ганта.

Открываем в Microsoft Word пустой документ в альбомном формате и скопируем туда из буфера таблицу и диаграмму Ганта. С полученным изображением можно теперь обращаться как с рисунком, растягивать, увеличивать для придания размеров, удобных для анализа.

Сохраняем результаты работы в виде *.doc файла на своей дискете. Примечание. Если таблица и диаграмма не умещаются на один рисунок, можно отдельно копировать таблицу и отдельно диаграмму (как два рисунка), сдвигая необходимым образом перед копированием вертикальный разделитель между таблицей и диаграммой Ганта.

Лабораторная работа №6 Анализ и оптимизация плана проекта при помощи MS Project

Содержание работы

Оптимизация плана проекта по методу критического пути при существовании ограничений

Сравнение оптимизированного плана проекта по методу критического пути с базовым планом

Анализ и оптимизация плана проекта по методу PERT

Сравнение оптимизированного плана проекта по методу PERT с базовым планом

Порядок выполнения работы

Оптимизация плана проекта по методу критического пути при существовании ограничений

В Области задач в подразделе Создание с помощью шаблона необходимо открыть Общие шаблоны... и в открывшемся окне на вкладке Общие открыть шаблон задания, выданного преподавателем (рис. 9).

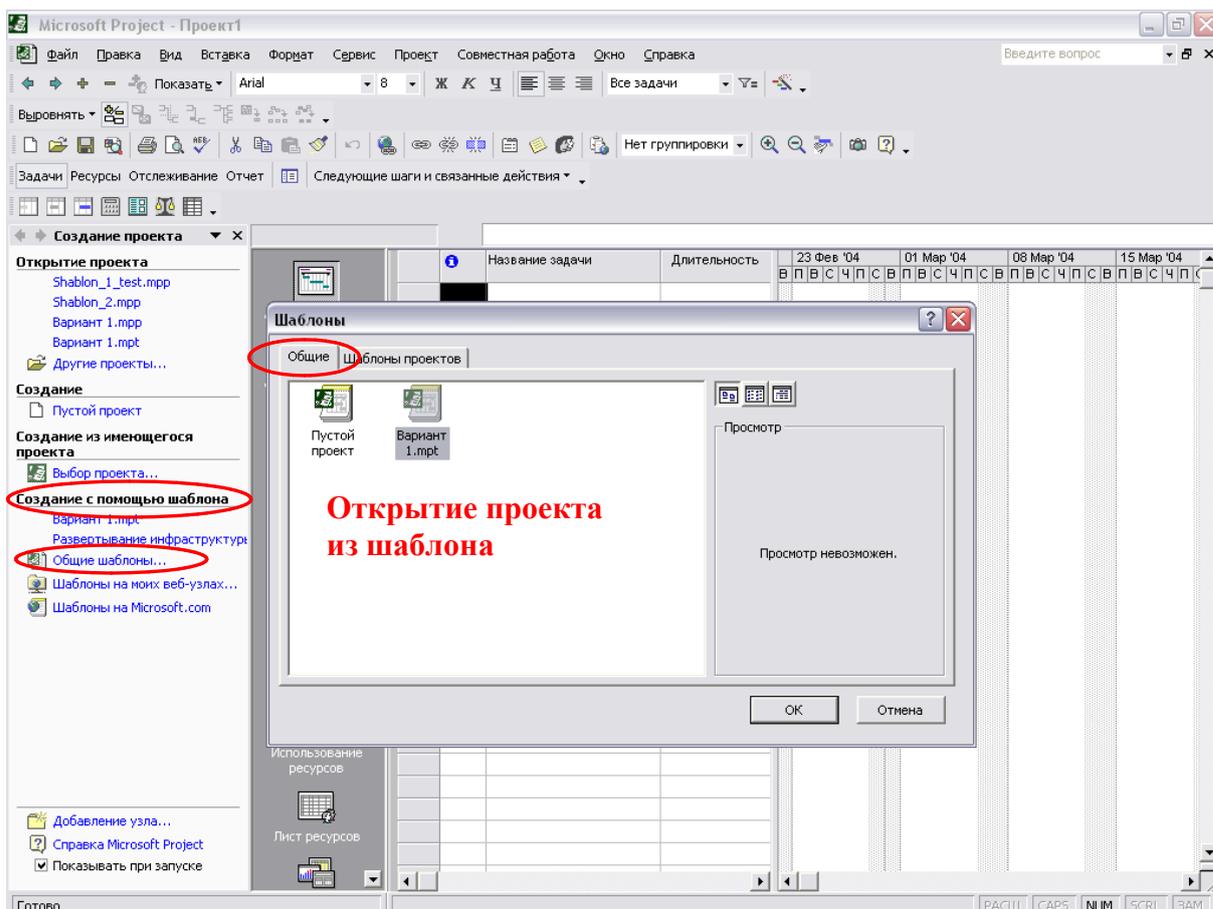


Рисунок 9. Открытие файла проекта.

В рамках данной лабораторной работы работа будет производиться с представлением Диаграмма Ганта с отслеживанием, к которому необходимо перейти аналогично тому, как это делалось в лабораторной работе № 1. По выданному преподавателем заданию производим оптимизацию плана проекта, учитывая ограничения и резервные ресурсы: дата окончания проекта должна быть на несколько рабочих дней ранее базовой; распараллеливание - для последовательных, критических работ в проекте используются резервные временные ресурсы, то есть ряд задач могут выполняться параллельно; сокращение длительности - для параллельных, последовательных, критических работ в проекте используются резервные трудовые ресурсы, то есть ряд задач могут выполняться за более короткие сроки за счет дополнительных исполнителей.

Сокращение длительности работ

В соответствии с условиями выданного задания, необходимо дважды, для одной и для второй выбранной задачи, назначить дополнительных исполнителей. Для выбранных работ надо увеличить загрузку во столько раз, сколько дополнительных исполнителей есть в резерве. Изменить загрузку можно, выполнив команды: Проект→Сведения о задаче→Ресурсы (рис 10).

В поле Единицы напротив ресурса выставить соответствующую загрузку (рис. 8). Например, если в задании указано, что, загрузка увеличивается в три раза, то она становится не 100%, а 300%.

Проанализировать сокращение длительности каждой из задач после назначения дополнительных ресурсов.

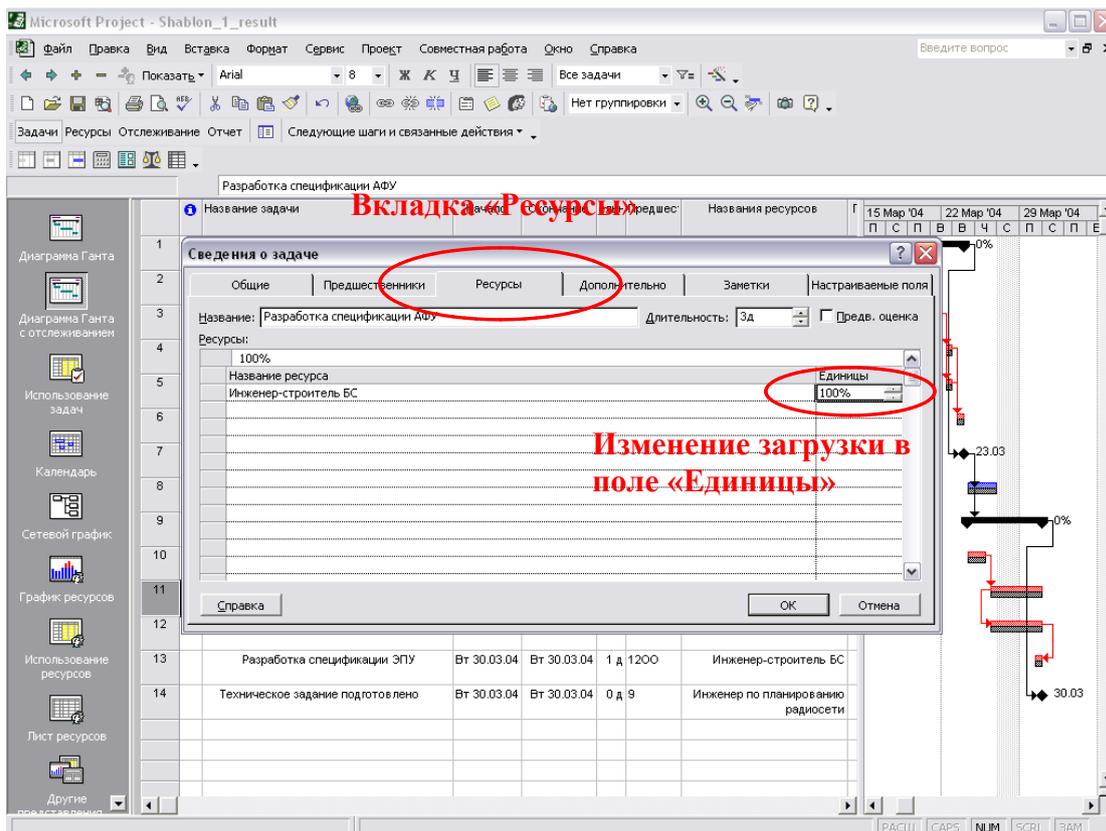


Рисунок 10. Переход на вкладку Ресурсы и изменение загрузки ресурса в поле Единицы.

Распараллеливание работ

В задании, выданном преподавателем, указаны задачи, которые могут выполняться параллельно. Для указанных задач следует изменить тип связи между ними, тем самым сократив длительность выполнения проекта. Изменить тип связи для задачи можно, выполнив команды: Проект→Сведения о задаче→Предшественники

В выпадающем списке столбца Тип задать соответствующие заданию типы связей (рис. 11).

Проанализировать изменение длительности плана проекта после изменения загрузки и распараллеливания работ. Эти изменения будут заметны в сравнении с базовым планом проекта, который показан в виде полосы черного цвета для соответствующих задач (рис. 11).

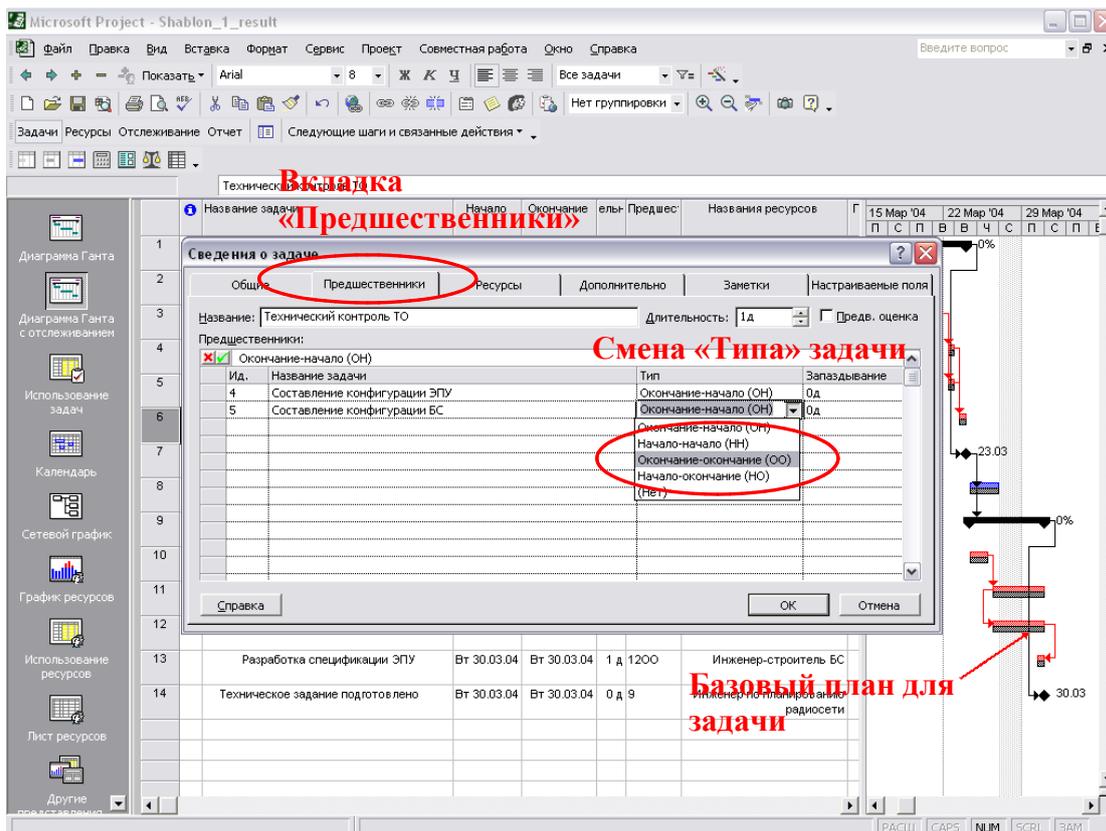


Рисунок 11. Переход на вкладку Предшественники в Сведениях о задаче и смена типа связи в выпадающем списке Тип.

Сравнение оптимизированного плана проекта по методу критического пути с исходным планом

По измененной после выполнения предыдущего пункта диаграммы Ганта сравнить полученный оптимизированный план проекта по методу критического пути с базовым планом. Проверить, выполнено ли условие окончания плана проекта.

На панели инструментов, нажав кнопку Копировать рисунок, выбрать вариант Для отображения на экране. Предварительно необходимо настроить столбцы (можно менять ширину столбцов, следя, чтобы все нужные столбцы попали в картинку) и рисунок (нужно, чтобы на диаграмме был виден весь проект, для этого следует правой кнопкой мыши щелкнуть в сером поле заголовка диаграммы, выбрать Масштаб, выставить Весь проект и нажать ОК), сохранить диаграмму в буфер обмена. Открыть новый документ в Microsoft Word, вставить, сохраненный в буфере фрагмент. С полученным изображением можно теперь обращаться как с рисунком, растягивать, увеличивать для придания размеров, удобных для анализа. Сохраняются результаты работы в виде *.doc файла на своей дискете

Анализ и оптимизация плана проекта по методу PERT

В Области задач в подразделе Создание с помощью шаблона необходимо открыть Общие шаблоны... и в открывшемся окне на вкладке Общие открыть шаблон задания, выданного преподавателем (рис. 9).

Настройки проекта должны соответствовать произведенным в лабораторной работе № 1. Для удобства можно работать непосредственно с панелью инструментов, для этого следует произвести действия Вид→Панели инструментов→Анализ по методу PERT.

На панели инструментов появится новый набор кнопок, позволяющих работать в режиме PERT - анализа

Отобразить таблицу ввода данных по срокам с прибавками для анализа, нажав кнопку, Лист ввода PERT. Ввести в колонки Оптимистическая длительность, Ожидаемая

длительность, Пессимистическая длительность соответствующие длительности с учетом прибавок из соответствующих полей таблицы задания к исходной длительности из колонки Длительность.

На панели инструментов, нажав кнопку Копировать рисунок, выбрать вариант Для отображении на экране и сохранить таблицу в виде рисунка в файл формата *.doc.

Провести анализ по методу PERT при заданных программой весовых коэффициентах
Нажать кнопку Задание весовых коэффициентов по методу PERT.

Программа откроет новое окно и предложит следующие коэффициенты для оптимистической длительности -1;

для ожидаемой длительности – 4;

для пессимистической длительности – 1.

Если преподавателем не предложен иной вариант (в любом случае сумма коэффициентов должна быть равна 6) согласиться, нажав ОК

Произвести анализ, нажав кнопку Вычисление по методу PERT. Программа рассчитает реалистический план выполнения проекта.

На панели инструментов, нажав кнопку Копировать рисунок, выбрать вариант Для отображении на экране и сохранить таблицу в виде рисунка в файл формата *.doc.

Перейти в представление Диаграмма Ганта с отслеживанием.

На панели инструментов, нажав кнопку Копировать рисунок, выбрать вариант Для отображении на экране и сохранить таблицу в виде рисунка в файл формата *.doc.

Проанализировать и сравнить полученный оптимизированный план проекта по методу PERT с базовым планом. Проверить, нарушилась ли заданная дата окончания проекта.

Методические указания по самостоятельному изучению курса

Самостоятельная работа студентов заключается в углубленном и разностороннем изучении тем, обозначенных в лекционном курсе. Также необходимо проявлять знания для выполнения задания по самостоятельному изучению отдельных вопросов освещенных в лекционном курсе, которые выполняются либо в виде конспекта по рекомендуемой литературе, либо в виде поиска необходимой информации во всемирной сети и других доступных источниках информации.

Результаты выполнения самостоятельной работы докладываются студентами во время аудиторных занятий, в форме отчета.

Содержание отчета

В отчете следует указать:

Цель работы

Введение

Программно-аппаратные средства, используемые при выполнении работы.

Основную часть (описание самой работы)

Заключение (выводы)

Список используемой литературы

В процессе самостоятельного изучения необходимо:

Построить временную и сетевую диаграммы для выбранного проекта.

Построить диаграмму распределения участников группы по этапам.

Построить список возможных рисков с указанием названия риска, его описание и типа.

Провести анализ рисков.

Описать стратегию планирования рисков.

Построить отчёт, включающий все полученные диаграммы и описание стратегии планирования рисков.

Построить функциональную модель системы, описанной в Лабораторной работе № 3 так, чтобы она отвечала всем предъявленным к системе требованиям, представляла полный функционал системы (каждой функции в описании системы должен соответствовать по крайней мере один функциональный блок) и её основные бизнес-процессы:

с помощью методологии IDEF0 построить контекстную диаграмму;

с помощью методологии IDEF0 построить диаграмму 1-го уровня (A0) – модель окружения;

с помощью методологии IDEF3 декомпозировать функциональные блоки модели окружения на 1-2 уровня вглубь до потоков, связи с внешними системами и

на каждой диаграмме 2-го уровня должно быть не менее 4-х функциональных блоков;

на каждой диаграмме 3-го уровня и далее не менее 2-х функциональных блоков.

Построить отчёт, включающий все полученные уровни модели, описание функциональных блоков, потоков данных, хранилищ и внешних объектов.

Список рекомендуемой литературы

3.1. Основная литература

1. Б. В. Черников. Информационные технологии управления : Учебник / Б. В. Черников. - М.: Форум, 2008 ; М. : Инфра-М, 2008. - 351 с. - (Высшее образование) .– (10 экз. в библ. ТУСУР)
2. И. Л. Туккель Управление инновационными проектами : учебник для студентов вузов, обучающихся по направлению подготовки "Инноватика" / И. Л. Туккель, А. В. Сурина, Н. Б. Культин ; ред. И. Л. Туккель. - СПб. : БХВ-Петербург, 2011. - 396 с. : ил (20 экз. в библ. ТУСУР)
3. В. Д. Ковалева Автоматизированное рабочее место экономиста : [Электронный ресурс] учеб. пособие/ В. Д. Ковалев, В. В. Хисамутдинов.-М.: Финансы и статистика, ИНФРА-М, 2010, 336 с.:ил /- on-line URL: <http://e.lanbook.com/view/book/1016/>

3.2. Дополнительная литература

1. О.Н.Бекетова Бизнес-план: теория и практика : учебное пособие для вузов / О. Н. Бекетова, В. И. Найденков. - М. : Приор-издат, 2009. - 284 с. : ил. (10 экз. в библ. ТУСУР)
2. В. В. Покровский Математические методы в бизнесе и менеджменте : учебное пособие / В. В. Покровский. - 2-е изд., испр. . - М. : БИНОМ. Лаборатория знаний, 2008. - 109 с. : ил. (1 экз. в библиотеке ТУСУР)
3. О. Н. Граничин Информационные технологии в управлении : учебное пособие для вузов / О. Н. Граничин, В. И. Кияев. - М. : Интернет-Университет Информационных Технологий, 2008 ; М. : БИНОМ. Лаборатория знаний, 2008. - 335 с. : ил. (1 экз. в библ. ТУСУР)
4. А. И. Исакова Информационные технологии в экономике : учебное пособие для студентов специальности 080801 / А. И. Исакова, М. Н. Исаков ; Федеральное агентство по образованию, Томский государственный университет систем управления и радиоэлектроники, Кафедра автоматизированных систем управления. - Томск : ТМЦДО, 2008. - 206 с : ил (12 экз. в библ. ТУСУР)
5. Информационные системы и технологии в экономике и управлении : Учебное пособие для вузов / Санкт-Петербургский государственный университет экономики и финансов ; ред. : В. В. Трофимов. - 2-е изд., перераб. и доп. - М. : Высшее образование, 2007. - 480 с. : ил. (20 экз. в библ. ТУСУР)