Федеральное агентство по образованию РФ

ГОУ ВПО Нижегородский государственный университет им. Н.И. Лобачевского

Факультет Вычислительной математики и кибернетики

Кафедра Математического обеспечения ЭВМ

УЧЕБНЫЙ КУРС

**«Объектно-ориентированный анализ и проектирование»**

для подготовки по направлению «Информационные технологии»

Нижний Новгород  
2006

###### Содержание[[1]](#footnote-1)

[1. Рамки проекта 3](#_Toc413347588)

[1.1. Матрица компромиссов проекта 3](#_Toc413347589)

[1.2. Вехи проекта 4](#_Toc413347590)

[1.3. Сметы проекта 5](#_Toc413347591)

[1.4. План-график проекта 5](#_Toc413347592)

[2. Роли и ответственности 5](#_Toc413347593)

[2.1. Знания, умения и навыки 6](#_Toc413347594)

[2.2. Структура команды 6](#_Toc413347595)

[3. Протоколы проекта 6](#_Toc413347596)

[3.1. Управление конфигурацией 6](#_Toc413347597)

[3.2. Управление изменениями 7](#_Toc413347598)

[3.3. Управление внедрениями 7](#_Toc413347599)

[3.4. Достижение качества проекта 7](#_Toc413347600)

[3.5. Рабочая среда проекта 7](#_Toc413347601)

Документ “Структура проекта” включает в себя информацию об организации проектной группы, персонификации ролей и ответственности. Также документ разъясняет схемы взаимодействия проектной группы с заказчиком и заказчика – с проектной группой.

# Рамки проекта

Рамки (scope) определяют пространство параметров, в котором будет создаваться решение, детализируя функциональность, определяя, что останется за рамками решения и указывая критерии, по которым заинтересованные лица будут судить о готовности решения. Рамки создаются на основе единого видения, являются результатом компромисса между сформулированными целями и условиями реальности и отражают приоритезацию заказчиком имеющихся требований к создаваемому решению. Частью процесса определения рамок проекта является вынесение менее важной функциональности из текущего проекта в планы на будущее.

Рамки проекта *(*project scope*)* определяют объем работ, который должен быть выполнен проектной группой для поставки заказчику каждого из элементов, определенного рамками решения.

Управление рамками проекта критично для его успеха. Необходимо определить и фиксировать рамки решения и проекта, используя треугольник компромиссов и матрицу компромиссов проекта.

## Матрица компромиссов проекта

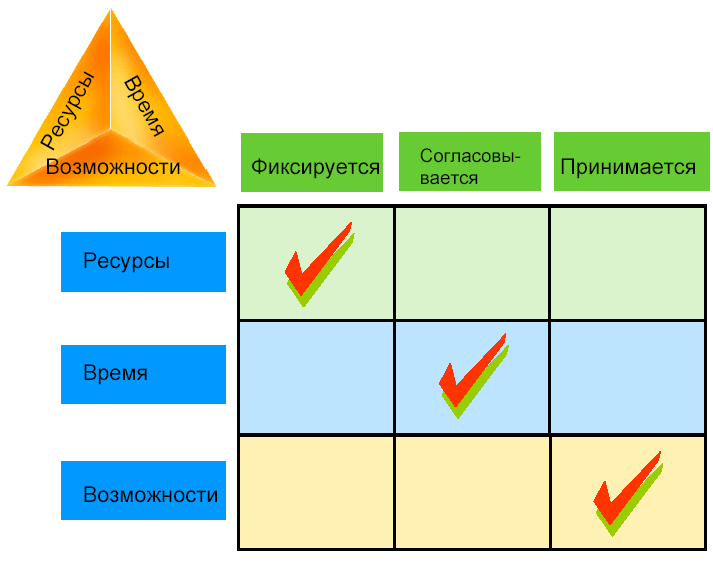
Хорошо известна взаимозависимость между ресурсами проекта (людскими и финансовыми), его календарным графиком (временем) и реализуемыми возможностями (рамками). Эти три переменные образуют треугольник компромисов (tradeoff triangle), показанный на рис. 1.

После достижения равновесия в этом треугольнике изменение на любой из его сторон для поддержания баланса требует модификаций на другой (двух других) сторонах и/или на изначально измененной стороне.



1. Треугольник компромиссов[[2]](#footnote-2)\*

Нахождение верного баланса между ресурсами, временем разработки и возможностями – ключевой момент в построении решения, должным образом отвечающего нуждам заказчика.



1. Матрица компромиссов\*

Другое весьма полезное средство для управления проектными компромиссами – матрица компромиссов проекта (project tradeoff matrix), показанная на  
рис. 2. Она отражает достигнутое на ранних этапах проекта соглашение между проектной группой и заказчиком о выборе приоритетов в возможных в будущем компромиссных решениях. В определенных случаях из этой приоритезации могут делаться исключения, но в целом следование ей облегчает достижение соглашений по спорным вопросам.

Рис. 2 показывает матрицу компромиссов проекта, используемую обычно проектными группами Майкрософт. Она помогает обозначить проектное ограничение, воздействие на которое практически невозможно (колонка “Фиксируется”), фактор, являющийся в проекте приоритетным (колонка “Согласовывается”), и третий параметр, значение которого должно быть принято в соответствии с установленными значениями первых двух величин (колонка “Принимается”).

Принципиально важно наличие у проектной группы и заказчика единого, однозначного взгляда на матрицу компромиссов проекта.

Укажите здесь оценки (в выбранных вами единицах) параметров треугольника компромиссов: ресурсы, которыми располагает ваш проект; имеющееся для реализации проекта время; возможности решения, которые, согласно описанию в документе “Концепция проекта”, будут вами реализованы. Расставьте приоритеты и постройте на их основе матрицу компромиссов.

Программный продукт для фирмы «YourNewTaxi» должен быть готов к окончанию 2016 года. Любые попытки расширить временные рамки гарантируют несоразмерное и вопиюще несправедливое увеличение налагаемых на команду разработчиков санкций, худшая из которых может повлечь за собой годовое посещение государственного спортивно-оздоровительного лагеря в стиле «милитари». В свою очередь, оптимизация проекта по времени, то есть, завершение оного ранее указанной временной черты едва ли подразумевает под собой наличие какого бы то ни было поощрения, если не считать таковым моральное удовлетворение разработчиков и заказчика.

Итог: время фиксировано.

В требованиях к программному продукту указан лишь базовый его функционал. Однако, команда разработчиков, будучи подверженной жажде самосовершенствования, готова выполнять заказ за один лишь бесценный опыт. Разработка только базового функционала отнюдь не способствует стремительному устранению невежества, а значит, в случае планомерного приближения продукта к готовому состоянию допускается совершенствование его функционала.

В частности, рассматриваются такие дополнительные возможности как:

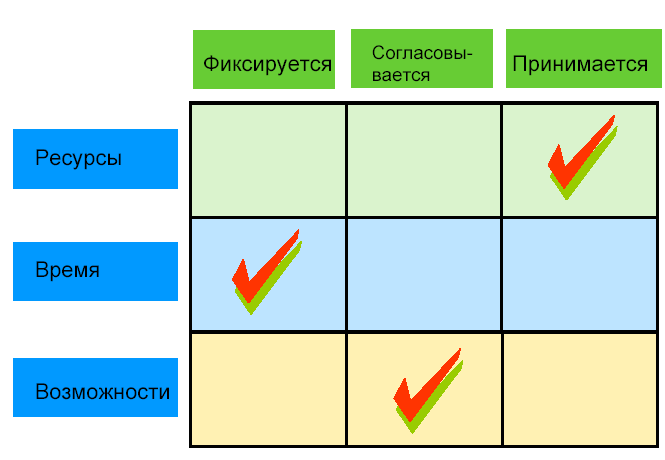
* Информирование клиента о примерном времени ожидания такси;
* Наличие у клиента дополнительных требований к транспортному средству: наличие подъёмника, багажника, количество мест в транспортном средстве, и т.д.;
* Наличие обратной связи;
* Организация «Чёрного списка» для клиентов, и системы наказаний для недобросовестных водителей такси;
* Наличие системы поощрения водителей на основе отзывов клиентов;
* Возможность заказа такси к определённому времени;

и др.

Итог: возможности согласовываются.

Так как в первую очередь были очерчены границы для времени и возможностей, ресурсы принимаются в том объёме, в котором это будет возможно с учётом жертв в пользу двух упомянутых выше приоритетов.

Итог: ресурсы принимаются.



## Вехи проекта

Вехи проекта – это существенные события в жизни проекта. На фазе выработки концепции обычно “выставляются” внешние вехи, которые показывают достижение видимых целей проекта. На самом верхнем уровне в качестве внешних вех могут рассматриваться окончания фаз выполнения проекта. В зависимости от природы проекта вехи могут быть основаны на финансовых затратах, на прогрессе в проекте, на временных интервалах и т.д.

Раннее определение вех позволяет установить точки на временном графике проекта, по которым заинтересованные стороны и проектная группа будут судить о ходе его выполнения.

Выберите и аргументируйте подход к определению вех в вашем проекте. Определите и опишите здесь вехи, на которые вы будете ориентироваться в ходе выполнения проекта.

Время и финансы не являются определяющими факторами, однозначно описывающими фазу выполнения продукта. В самом деле, скорость разработки не является величиной постоянной, а значит, выбрать осмысленные промежутки времени в качестве вех крайне проблематично, а неосмысленные вообще рассматривать не стоит, ведь не получится заранее сказать, каков в этот промежуток будет прогресс и будет ли он вообще. В качестве финансов выступает кристально чистый энтузиазм, для измерения которого не существует однозначного эталона сравнения.

Итак, вехами следует считать стадии готовности продукта.

1. Оболочка.

Имеется возможность запустить прототипы серверного и клиентского приложений с возможностью соединения их между собой. Графический интерфейс не пригоден для демонстрации оного заказчику или отсутствует вообще.

2. Базовый интерфейс и зачатки функционала.

Имеется возможность запустить клиентское приложение и лицезреть графический интерфейс, некоторые элементы которого реализуют простейший демонстрационный функционал.

3. Базовый интерфейс и развитый функционал.

Имеется возможность так или иначе наблюдать большую часть требуемых заказчиком возможностей программ, интерфейс может не покрывать эти возможности полностью и подвергаться изменениям.

4. Развитый интерфейс и развитый функционал.

Можно пользоваться программным продуктом, основной функционал которого доступен в пользовательском интерфейсе. Эта версия продукта может быть использована как демонстрационная.

5. Отлаженный продукт и доработанный интерфейс.

В продукте устранены ошибки и очевидные недочёты, интерфейс обретает конечный вид. Эта версия продукта готова для представления её заказчику.

6. Расширенный функционал.

Опциональная стадия, заключающаяся в добавлении новых возможностей в программный продукт с последующим исправлением новоявленных неисправностей.

## Сметы проекта

Выполнение проекта предполагает использование ресурсов и подразумевает наличие затрат. Ресурсы включают в себя людей, оборудование, различные расходные материалы и т.д. Затраты рассчитываются на основе тарифов (расценок) на каждый вид требуемого ресурса. В данном разделе должна быть представлена информация о:

* списке видов ресурсов,
* требуемом количестве каждого ресурса,
* тарифе на каждый вид ресурса,
* общей стоимости каждого ресурса,
* общей стоимости всех ресурсов, необходимых проектной группе.

На основе информации данного раздела должен рассчитываться бюджет проекта. Также этот этап – хорошая возможность идентифицировать специфические ресурсы, которые могут потребоваться для выполнения проекта.

Укажите, какие виды ресурсов необходимы вашей проектной группе для выполнения проекта, определите в выбранных единицах количество каждого ресурса, оцените общие ожидаемые затраты по ресурсам.

Виды ресурсов:

* Программное обеспечение
* Оборудование

ПО включает в себя операционную систему, среду разработки и платные программные библиотеки.

Оборудование главным образом включает в себя рабочие станции (компьютеры) и сопутствующую аппаратуру.

|  |  |
| --- | --- |
| IntelliJ IDEA Ultimate |  |
| Windows 10 Professional |  |
| Github Premium |  |
| Lenovo ThinkPad P70 |  |
| Logitech MX Master Mouse |  |

## План-график проекта

На этом этапе строится первый вариант графика выполнения проекта на основе выделенных вех и уже сформулированных задач, для каждой из которых задаются даты начала и окончания.

Процесс построения графика проекта итеративен. На фазе выработки концепции график строится на основных вехах проекта. На фазе планирования график становится более детальным в процессе выделения отдельных задач проекта.

На основе выделенных выше вех, а также сформулированных целей и задач определите ожидаемую временную протяженность проекта и постройте предварительный график выполнения проекта.

По вертикальной оси на графике расположены описанные выше вехи.

# Роли и ответственности

В данном разделе описывается организация проектной группы. Четкие требования к квалификации, ролям и ответственностям членов проектной группы позволяют менеджеру проекта правильно подобрать команду и дают каждому понимание его личного вклада в общий успех проекта.

## Знания, умения и навыки

Участники проектной группы должны удовлетворять определенным требованиям для успешного выполнения проекта. Эти требования выражаются в терминах знаний, умений и навыков и должны включать технические, управленческие и иные возможности.

На основе имеющейся к данному моменту информации о разрабатываемом решении и способе его реализации сформулируйте требования к знаниям, умениям и навыкам, которые необходимы участникам проектной группы. В процессе формулирования учитывайте, что требования должны касаться всех ролевых кластеров команды.

Все участники должны:

* Уметь работать в команде
* Иметь способности к коммуникации
* Уметь планировать свои действия
* Проводить анализ требований
* Уметь работать с UML
* Иметь знания о объектно-ориентированном программировании, разработке веб-приложений и работе с базами данных

Ответственные за логику приложения должны:

* Знать C#
* Владеть практическими навыками объектно-ориентированного программирования

Ответственные за графический интерфейс должны:

* Знать C#
* Владеть практическими навыками разработки GUI

Ответственные за сервер должны:

* Знать структуру TCP/IP
* Владеть практическими навыками создания и настройки сервера

Ответственные за базу данных:

* Знать SQL
* Владеть практическими навыками создания, редактирования и обработки баз данных

## Структура команды

Структура команды определяет организационные единицы (менеджер проекта, спонсоры, лидеры команд, и т.д.), задает отношения между ними и зоны их ответственности.

Распределите роли, в вашей команде и опишите здесь это распределение.

Руководитель проекта - Метелёв Евгений

Back-end разработчик - Сахаров Алексей

Front-end разработчик - Дружин Игорь

Мобильный разработчик - Карчков Денис

Дизайнер - Метелёв Евгений

Системный администратор - Карчков Денис

Тестировщики: Метелёв Евгений, Сахаров Алексей, Дружин Игорь

# Протоколы проекта

Протоколы проекта – это набор описаний процессов в проекте, которые должны быть утверждены, чтобы все участники команды действовали в одинаковом ключе. Стандартизация повышает производительность и облегчает общение как внутри проектной группы так и с заинтересованными сторонами.

## Управление конфигурацией

Многие составляющие решения в процессе работы над проектом претерпевают неоднократные изменения, следовательно, проект нуждается в некоторой стратегии управления конфигурацией, определяющей выбранные методы отслеживания изменений, ведения отчетности и др. Управление конфигурацией должно охватывать проектную документацию, среды разработки и тестирования и любые изменения в проектной среде в целом.

Данный раздел включает:

* описание методов и средств управления конфигурацией;
* описание шагов по запросу и принятию изменений в конфигурации;
* роли и ответственности в управлении конфигурацией;
* выбор и/или описание требований к системе контроля версий.

Опишите стратегию управления конфигурацией в проекте в соответствии с представленными выше пунктами.

Наш коллектив использует методологию RUP и руководствуется его основными принципами:

* Ранняя идентификация и непрерывное устранение основных рисков.
* Концентрация на выполнении требований заказчиков к исполняемой программе.
* Ожидание изменений в требованиях, проектных решениях и реализации в процессе разработки.
* Компонентная архитектура, реализуемая и тестируемая на ранних стадиях проекта.
* Постоянное обеспечение качества на всех этапах разработки проекта.
* Работа над проектом в сплочённой команде, ключевая роль в которой принадлежит архитекторам.

Команда состоит из четырех человек и располагается в общем офисе. Работаем 40 часов в неделю, пн-пт 10:00 - 18:00. Мы работаем в сплоченной команде, как и завещал RUP. В начале каждого рабочего дня проводится собрание, для обсуждения текущих результатов. Каждую неделю происходит встреча с заказчиком, где ему предоставляется подробная информация о текущем прогрессе. Однако, встречи с заказчиком не ограничены одним днем в неделю и если имеется необходимость, встреча может быть организована в любой рабочий день. Основная проектная документация пишется во время фазы “Уточнение”. Инструкция по программному комплексу пишется одновременно с кодом и дополняется по мере реализации новых функций. Для контроля версий используется система Git.

## Управление изменениями

Одна из существенных идей, лежащих в основе объектно-ориентированного подхода – ориентированность на изменения. В то же время порядок внесения предложений, утверждения и реализации изменений и отслеживания результатов должен быть определен и зафиксирован. Кроме того, все изменения в проекте должны проводиться в соответствии с принятыми и утвержденными рамками решения и рамками проекта. В данном разделе должны быть описаны:

* процесс управления изменениями;
* форма запроса на изменение;
* роли и ответственности в процессе управления изменениями;
* действия в случае, если предложенные изменения могут отразиться на контракте с заказчиком (в том числе, если эти действия были инициированы самим заказчиком, см. треугольник компромиссов).

Опишите стратегию управления изменениями в проекте в соответствии с представленными выше пунктами.

Заказчик может подать предложение (в устной форме) на изменение или реализацию новых функций во время проведения организационной встречи с ним. Т.к. мы придерживаемся методологии RUP, то мы этого и ожидаем. В такой ситуации мы примем во внимание пожелания заказчика и постараемся скорректировать текущее решение. При этом соответствующая проектная документация будет дополнена и обновлена.

## Управление внедрениями

Работа над решением не заканчивается с окончанием этапа разработки. Важным аспектом, требующим внимания, является и процесс внедрения как пилотных так и финальных версий решения. В данном разделе должны быть описаны способы и средства по подготовке релизов и управлению их внедрением, как в тестовые, так и в производственные окружения.

Сформулируйте и опишите здесь мероприятия, необходимые для будущего внедрения решения, которое будет разработано в ходе проекта.

Планируется организационное мероприятие для водителей, в ходе которого будут рассказаны все преимущества работы с данным продуктом и предоставлена подробная инструкция по эксплуатации. Далее будет открыт доступ к бета-тестированию в ходе которого приложение будет периодически обновляться, а так же будут собираться отзывы водителей и клиентов. Как только всех все будет устраивать, обновление завершится и будет выпущена финальная версия продукта.

## Достижение качества проекта

Данный раздел описывает, каким образом в ходе выполнения проекта будут удовлетворяться ожидания заказчика и будущих пользователей к качеству создаваемого решения, как с точки зрения разработчиков, так и со стороны управления. Достижение требуемого качества решения должно быть описано через:

* ожидания к качеству решения;
* процесс проверки качества;
* процесс управления достижением качества;
* роли и ответственности в процессе достижения качества.

Сформулируйте и опишите стратегию по достижению требуемого качества проекта.

Как для клиента, так и для водителя основными показателями качества являются: интуитивно понятный интерфейс, быстрая, стабильная и корректная работа приложения.

Для обеспечения качества используется подход тестирования программного обеспечения. Для проверки правильности работы приложения применяется функциональное тестирование. Для проверки производительности и стабильности – нагрузочное и стресс-тестирование. Юзабилити-тестирование для определения приемлемости графического интерфейса. А также, как было указано выше, будет применен подход бета-тестирования, уже после проведения всех других тестов.

## Рабочая среда проекта

В данном разделе должен быть описан подход к созданию рабочей среды проекта, определяющий как организационные требования (размещение членов команды, комнаты для митингов и т.д.), так и требования к оборудованию (компьютеры, столы, телефоны и т.д.). Здесь же должны определяться требования к инструментам и системам, таким как системы контроля версий, среды разработки, инструменты тестирования и др.

Сформулируйте и опишите требования к рабочей среде проекта.

Организационные требования:

Офис с наличием wi-fi на четырех человек 18 м2. Собрания проводятся в этом же офисе.

Оборудование:

|  |
| --- |
| Lenovo ThinkPad P70 |
| Logitech MX Master Mouse  Стол  Стул |

Система контроля версий – Git

Среда разработки - Microsoft Visual Studio 2015

Тестирование производится с помощью инструментов Visual Studio

1. В документе использованы материалы белых книг (white papers) “MSF Process Model”, “MSF Risk Management Discipline”, “MSF Team Model” (<http://www.microsoft.com/msf>), их переводов “Модель процессов MSF”, “Дисциплина управления рисками MSF”, “Модель проектной группы MSF” выполненных в 2003 году корпораций eLine Software (<http://www.elinesoftware.com>), а также официальных курсов Microsoft 2710B и 1846A. [↑](#footnote-ref-1)
2. \* Рисунок заимствован из белой книги (white paper) “Модель процессов MSF” в переводе корпорации eLine Software 2003 года. [↑](#footnote-ref-2)