**Лекция 7 СПЕЦИФИКАЦИЯ ТРЕБОВАНИЙ К ПО**

Прочитав эту лекцию, вы узнаете:

* *В чем заключается объектно-ориентированный подход к спецификации требований.*

## Содержание

[Содержание 1](#_Toc242075605)

[СПЕЦИФИКАЦИЯ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ 2](#_Toc242075606)

[ОСНОВЫ СПЕЦИФИКАЦИИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ 2](#_Toc242075607)

[3.4.2. ПРИМЕР СПЕЦИФИКАЦИИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ 10](#_Toc242075608)

[! Следует запомнить. 16](#_Toc242075609)

[S Основные понятия 16](#_Toc242075610)

[? Вопросы для самоконтроля 16](#_Toc242075611)

## СПЕЦИФИКАЦИЯ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ[[1]](#footnote-1)

### ОСНОВЫ СПЕЦИФИКАЦИИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Спецификация требований к ПО является составной частью процесса управления требованиями. В качестве повторения отметим, что все требования к ПО делятся на функциональные и нефункциональные.

***Функциональные требования*** определяют действия, которые должна выполнять система, без учета ограничений, связанных с ее реализацией. ***Нефункциональные требования*** описывают атрибуты системы (технические характеристики) и ее окружения.

Для выявления требований используются (в различных сочетаниях) следующие методы:

* собеседование (интервьюирование);
* анкетирование;
* моделирование и анализ бизнес-процессов (далее будет рассмотрена методика перехода от объектной бизнес-модели к системным требованиям);
* сессии по выявлению требований (мозговой штурм);
* создание и демонстрация пользователям работающих прототипов приложений (для выявления замечаний и дополнительных требований).

***Собеседование*** *-* наиболее часто используемый и обычно наиболее полезный метод выявления требований. При использовании интервью может быть поставлено несколько целей, таких, как выяснение, проверка и толкование фактов, формирование заинтересованности, привлечение пользователей к работе, сбор предложений и мнений. Но метод собеседования требует для эффективного контакта хороших навыков общения с людьми, имеющими различные ценности, приоритеты, мнения, побуждения и индивидуальные особенности. Преимущества метода собеседования заключаются в следующем:

* опрашиваемое лицо может свободно и открыто отвечать на вопросы и почувствовать себя участником проекта;
* лицо, проводящее собеседование, может наблюдать за поведением опрашиваемого лица, изменять ход опроса, переформулировать или иначе строить вопросы во время собеседования.

Недостатки метода собеседования:

* метод трудоемкий и дорогой, поэтому может оказаться непрактичным;
* успех собеседования зависит от навыков общения лица, проводящего собеседование, и от желания опрашиваемых лиц участвовать в интервью.

Существуют два типа интервью: неструктурированное и структурированное. *Неструктурированные интервью* проводят с одной общей целью, которая явно не высказывается, и с помощью нескольких общих вопросов. Лицо, проводящее собеседование, рассчитывает на то, что опрашиваемое лицо должно само определять рамки и направление интервью. В таких интервью часто теряется нить рассуждений, и по этой причине они редко используются при проектировании систем.

В *структурированных интервью* лицо, проводящее собеседование, заранее подготавливает конкретный ряд вопросов к опрашиваемому лицу. В зависимости от ответов в процессе собеседования могут быть заданы дополнительные вопросы для уточнения или разъяснения некоторых тем. *Вопросы без подразумеваемых ответов* позволяют опрашиваемому лицу выбрать ответ, наиболее подходящий с его точки зрения. *Вопросы, допускающие единственный ответ,* ограничивают выбор возможного ответа или требуют коротких, прямых ответов.

***Анкетирование*** позволяет с помощью анкет получать сведения от большого количества людей, контролируя правильность их ответов. При работе с большой аудиторией этот метод наиболее эффективен. Преимущества метода анкетирования заключаются в следующем:

* люди могут заполнять и возвращать анкеты в удобное для них время;
* люди склонны сообщать в ответах действительные факты, если анкетирование анонимное;
* это относительно недорогой способ сбора данных с участием большого количества людей.

Недостатки метода анкетирования:

* не все могут согласиться ответить на вопросы, анкеты могут возвращаться не заполненными;
* нет возможности пояснить или переформулировать неправильно понятые вопросы, наблюдать и анализировать реакцию респондента на отдельные вопросы;
* подготовка анкет может потребовать много времени.

***Мозговой штурм*** особенно полезен в ситуациях, когда обсуждаются новые, нестандартные решения незнакомой проблемы. Большинство новаторских идей очень часто бывают результатом комбинации нескольких, на первый взгляд, не связанных друг с другом идей. Мозговой штурм включает в себя генерацию идей и расстановку приоритетов. Процесс генерации идей обычно подчиняется правилам, соблюдение которых преследует одну цель — выработку как можно большего числа идей. Критика и споры в этот период не поощряются; существующие пределы и ограничения снимаются, можно изменять и комбинировать идеи.

Основная идея ***создания прототипа*** состоит в быстрой постройке модели системы, которая, как предполагается, нужна пользователю. Обычно в прототипах многие детали опускаются, в том числе процедуры контроля входных данных, обработки ошибок, резервного копирования и восстановления данных. Не учитываются также производительность и масштабируемость. Создание прототипа никак не противоречит и не препятствует применению других методов получения требований от пользователя. Однако иногда только с помощью прототипа удается заставить клиента начать обсуждение требований, которые при другом подходе кажутся слишком абстрактными. Кроме того, работа с прототипом часто помогает получить требования, которые в противном случае так и остались бы неизвестными. Прототип хорош еще и тем, что он является неким осязаемым доказательством (или иллюзией) прогресса в разработке системы. А в некоторых случаях он может даже поддерживать какие-то ограниченные рабочие возможности системы. К сожалению, как отмечалось, реальный опыт работы с прототипами свидетельствует, что их создание обычно препятствует любому официальному документированию требований, следовательно, требования формулируются только в виде кода. Детали работы системы, игнорируемые прототипом (обработка ошибок, резервное копирование и др.), могут так и не появиться в виде требований.

Выявленные в результате применения перечисленных методов требования к ПО оформляются в виде ряда документов и моделей. К основным документам, регламентируемым технологией Rational Unified Process, относятся:

* Концепция - определяет глобальные цели проекта и основные особенности разрабатываемой системы. Существенной частью концепции является постановка задачи разработки, определяющая требования к выполняемым системой функциям.
* Словарь предметной области (глоссарий) — определяет общую терминологию для всех моделей и описаний требований к системе. Глоссарий предназначен для описания терминологии предметной области и может быть использован как словарь данных системы.
* Дополнительные спецификации (технические требования) — содержит описание нефункциональных требований к системе, таких, как надежность, удобство использования, производительность, сопровождаемость и др.

Примеры документов приведены в подразд. 3.4.2.

Функциональные требования к системе моделируются и документируются с помощью вариантов использования (use case), которые в контексте процесса управления требованиями трактуются следующим образом:

* вариант использования фиксирует соглашение между участниками проекта относительно поведения системы;
* вариант использования описывает поведение системы при различных условиях, когда система отвечает на запрос одного из участников, называемого основным действующим лицом;
* основное действующее лицо инициирует взаимодействие с системой, чтобы добиться некоторой цели. Система отвечает, соблюдая интересы всех участников.

Варианты использования - это вид документации, применяемый, когда требуется сконцентрировать усилия на обсуждении принципиальных требований к разрабатываемой системе, а не на подробном их описании. Стиль их написания зависит от масштаба, количества участников и критичности проекта. Существуют четыре уровня точности[[2]](#footnote-2) при описании вариантов использования (расположенные по степени повышения точности):

* Действующие лица и цели (перечисляются действующие лица и все их цели, которые будет обеспечивать система).
* Краткое изложение варианта использования (в один абзац) или основной поток событий (без анализа возможных ошибок).
* Условия отказа (анализ мест возникновения возможных ошибок в основном потоке событий).
* Обработка отказа (написание альтернативных потоков событий).

Введение перечисленных уровней преследует своей целью грамотное планирование и экономию времени разработки. В итерационном цикле создания системы не следует пытаться за один прием подробно описать все требования, их нужно постепенно уточнять, повышая уровень точности. Выбор первоочередных вариантов использования для уточнения определяется их приоритетами (см. подразд. 1.5).

Методика моделирования вариантов использования в технологии Rational Unified Process предусматривает специальное соглашение, связанное с группировкой структурных элементов и диаграмм модели. Это соглашение включает следующие правила:

* Все действующие лица, варианты использования и диаграммы вариантов использования помещаются в пакет с именем Use Case Model.
* Если моделируется сложная многофункциональная система, то совокупность всех действующих лиц и вариантов использования может разделяться на пакеты. В качестве принципов разделения могут использоваться:
  + структуризации модели в соответствии с типами пользователей (действующих Лиц);
  + функциональная декомпозиция;
  + разделение модели на пакеты между группами разработчиков (в качестве объектов управления конфигурацией).

Все рекомендации по написанию качественных вариантов использования можно изложить в виде набора образцов, которые перечислены в таблице 7.1.

Таблица 7.1 - Образцы написания вариантов использования

|  |  |  |
| --- | --- | --- |
| Наименование образца | Проблема | Предлагаемое решение |
| Формирование команды разработчиков | | |
| Небольшая команда разработчиков | Участие слишком многих людей в написании варианта использования неэффективно; компромисс между многими различными точками зрения может привести к неудовлетворительным результатам при создании системы - | Ограничьте число разработчиков, отрабатывающих детали варианта использования, двумя или тремя людьми. Для подключения дополнительных участников используйте образец Двухуровневое рецензирование - |
| Состав сторонних участников | Невозможно удовлетворить потребности всех заинтересованных лиц, не обмениваясь информацией с ними | Активно привлекайте заказчиков и заинтересованных лиц в вашей организации к разработке вариантов использования |
| Сбалансированная команда | Команда из близких по роду деятельности, одинаково мыслящих людей может создать небольшой набор ограниченных вариантов использования, не удовлетворяющий общим потребностям | Формируйте команду из людей с различными специализациями, представляющими интересы различных заинтересованных лиц. Команда должна включать как разработчиков, так и пользователей |
| Организация процесса разработки | | |
| Глубина после об­щего представле­ния | Невозможно продвигаться вперед и создавать набор согласованных вариантов использования, если тратить впустую время на последовательное написание подробных вариантов использования | Разрабатывайте сначала общее представление вариантов использования, затем постепенно добавляйте детали, работая с группой взаимосвязанных вариантов использования |
| Итерационная разработка | Разработка вариантов использования за один проход затруднительна, усложняет внесение дополнительной информации и затрудняет выявление факторов риска | Разрабатывайте варианты использования итерационно, повышая на каждой итерации их точность и корректность |
| Множество форм | Различные проекты требуют различную степень формальности и различные шаблоны. Использовать один и тот же шаблон вариантов использо­вания непродуктивно | Выбирайте формат вариантов использования, основываясь на проектных рисках и предпочтениях участников |
| Наименование образца | Проблема | Предлагаемое решение |
| Двухуровневое рецензирование | Многие заинтересованные лица могут пожелать принять участие в рецензировании вариантов использования. Это слишком долгое и дорогостоящее занятие | Используйте два вида рецензирования: первое, проводимое небольшой группой внутренних участников, может выполняться многократно; второе, проводимое полной группой, выполняется, как правило, один раз |
| Своевременное завершение | Разработка модели вариантов использования сверх потребностей заинтересованных лиц и разработчиков приводит к напрасным тратам ресурсов и затягивает проект | Прекращайте разработку вариантов использования, как только они достигают необходимой полноты и удовлетворяют потребности участников проекта |
| Свобода творчества | Чрезмерное внимание к стилю написания вариантов использования тормозит работу | Небольшие различия в стиле написания вариантов использования несущественны. Если вариант использования достиг необходимой полноты, его автор имеет право на стилистические отступления |
| Организация набора вариантов использования | | |
| Общепринятая четкая концепция | Недостаточно четкое представление о системе может привести к неуверенности и противоположным мнениям среди заинтересованных лиц и быстро парализовать проект | Необходимо подготовить и утвердить концепцию системы, в которой четко определены ее цели и роль в деятельности организации. Распространите концепцию среди всех участников Проекта |
| Наименование образца | Проблема | Предлагаемое решение |
| Видимые границы | Если вы не определили границы системы, ее масштаб будет расти не­контролируемым образом | Установите четкую и видимую границу между системой и внешней средой, перечислив людей и оборудование, взаимодействующих с данной системой |
| Ясный состав действующих лиц | Если выявлять только пользователей системы, игнорируя роли, которые они играют по отношению к системе, можно упустить существенную часть поведения системы или ввести избыточное поведение | Идентифицируйте действующих лиц, с которыми должна взаимодействовать система, и роли, которые они играют по отношению к системе. Четко опишите каждую роль |
| Транзакции, значимые для пользователей | Система несовершенна, если она не может предоставить пользователям необходимые услуги и не выполняет цели и задачи, определяемые ее концепцией | Идентифицируйте важные, значимые услуги, которые система предоставляет действующим лицам для удовлетворения их потребностей |
| Разворачивающееся представление | Количество шагов в описании поведения системы превышает возможности охвата и интересы различных типов читателей вариантов использования | Создайте иерархическую организацию набора ва­риантов использования, которую можно развернуть для большей детализации, или свернуть, чтобы скрыть детали и показать контекст |
| Отдельный вариант использования | | |
| Законченная единственная цель | Неправильно заданные цели не дают разработчикам четко определить, где заканчивается один вариант использования и начинается другой | Каждый вариант использования должен иметь законченную и четко определенную цель, которая может находиться на любом уровне образца Разворачивающееся представление |
| Наименование образца | Проблема | Предлагаемое решение |
| Имя в виде глагольной фразы | Бессмысленные, слишком общие имена вариантов использования не дают читателям правильного представления, на них сложно ссылаться | Именуйте каждый вариант использования активной глагольной фразой, отражающей цель основного действующего лица |
| Сценарий и фрагменты | Читатели должны иметь возможность легкого следования по интересующему их сценарию, в противном случае они могут утратить интерес или потерять важную информацию | Основной сценарий должен быть предельно простым, без анализа возможных ошибочных ситуаций. Фрагменты сценария, отражающие альтернативы, должны располагаться под ним |
| Исчерпывающие альтернативы | Вариант использования может иметь много альтернатив. Отсутствие некоторых из них означает, что разработчики неправильно понимают поведение системы, и система может получиться несовершенной | Описывайте все альтернативы и ошибочные ситуации, которые могут иметь место в варианте использования |
| Перегрузка ин­формацией | Включение нефункциональных требований в вариант использования может быстро привести его в неопределенное и беспорядочное состояние | Включите дополнительные позиции в шаблон варианта использования за пределами текста сценария, чтобы отразить полезную дополнительную информацию |
| Точность и читае­мость | Варианты использования, слишком сложные для нетехнических специалистов или слишком не точные для разработчиков, несовершенны и могут привести к созданию неадекватной системы - | Сделайте вариант использования достаточно читаемым, чтобы заинтересованные лица могли в нем разобраться и оценить его, и достаточно точным, чтобы разработчики понимали, что им делать |
| Наименование образца | Проблема | Предлагаемое решение |
| Сценарии и шаги | |
| Обнаруживаемые условия | Разработчики вариантов использования всегда решают проблему, сколько и какие условия включить в их описание | Включайте только реально обнаруживаемые условия. Объединяйте условия, которые оказывают на систему одинаковое воздействие |
| Уровни шагов | Чрезмерно крупные или мелкие шаги сценария варианта использования делают вариант использования трудным для восприятия и понимания | Оставляйте в сценарии от трех до девяти шагов. В идеальном случае все шаги должны быть на близких уровнях и на уровне абстракции, следующем за целью варианта использования |
| Выполнение цели действующего лица | У читателей и разработчиков возникают проблемы в понимании поведения системы, если неясно, какое действующее лицо отвечает за выполнение шага сценария, и что оно делает для его завершения | При описании каждого четко указывайте, какое действующее лицо выполняет действие, и что оно получает по его завершении |
| Продвижение впе­ред | Разработчики должны решить, как много поведения включить в каждый шаг. Слишком много деталей делает вариант использования длинным и трудным для чтения | Исключайте или объединяйте шаги, которые не означают никакого движения вперед для действующего лица. Упрощайте фрагменты, которые отвлекают внимание читателя от движения вперед |
| Наименование образца | Проблема | Предлагаемое решение |
| Нейтральность к технологии | Включение технологических ограничений и деталей реализации в описание варианта использования увеличивает сложность и затрудняет понимание его цели | Описание варианта использования должно быть нейтральным по отношению к технологии |
| Связи между вариантами использования | | |
| Общее поведение | Описание одинаковых шагов в различных вариантах использования занимает лишнее время и затрудняет понимание общих процессов в модели вариантов использования | Выражайте общие действия в виде «включаемых» вариантов использования |
| Перемещение аль­тернатив | Длинные или сложные описания альтернатив могут занять доминирующее положение и показаться более важными, чем они заслуживают | Рассмотрите возможность выделения сложных альтернатив в отдельный вариант использования |
| Абстракция | Попытка описать в одном варианте использования две или более различных альтернатив, ни одна из которых не является доминирующей, приводит к проблемам | Создайте обобщенный абстрактный вариант использования. Поместите каждый отдельный вариант сценария, уточняющий абстракцию, в специализированный вариант использования |
| Модификация существующих вариантов использования | | |
| Удаление изли­шеств | Чрезмерно длинный вариант использования громоздок и труден для работы, уводит в сторону внимание пользователей | Переместите длинный, громоздкий фрагмент или слишком сложное расширение в отдельный вариант использования |
| Наименование образца | Проблема | Предлагаемое решение |

|  |  |  |
| --- | --- | --- |
| Объединение фрагментов | Варианты использования, описывающие очень маленькие или изолированные фрагменты поведения, не выражают достаточной для понимания информации относительно услуг, предоставляемых системой (Транзак­ции, значимые для поль­зователей) | Объединяйте небольшие взаимосвязанные варианты использования или фрагменты вариантов использования в варианты использования, связанные общей целью |
| Удаление лишних вариантов исполь­зования | Варианты использования, результаты которых незначительны, отвлекают внимание и затрудняют понимание системы | Удалите варианты использования, которые не дают ничего существенного для системы или выпали из текущего активного списка вариантов использования |

Варианты использования, безусловно, не единственный способ выражения функциональных требований. Это объясняется следующими причинами:

* их описания слишком подробны. Заинтересованные лица зачастую желают видеть лишь краткий перечень основных функций;
* некоторые важные функции системы легче выразить в краткой форме без привязки к именам вариантов использования.

Следовательно, помимо вариантов использования, функции системы можно выразить через ее свойства (system features), представляющие собой высокоуровневые, краткие утверждения. Более строго в контексте Rational Unified Process системное свойство определяется как «наблюдаемая извне и обеспечиваемая системой функция, которая непосредственно удовлетворяет потребности заинтересованного лица». Свойство можно облечь в следующую лингвистическую форму: система будет выполнять «свойство X».

Спецификация требований в технологии Rational Unified Process не требует обязательного моделирования бизнес-процессов организации, для которых создается ПО, однако наличие бизнес-моделей существенно упрощает построение системной модели вариантов использования. При переходе от бизнес-модели к начальной версии модели вариантов использования применяются следующие правила.

* Для каждого исполнителя в модели бизнес-анализа, который в перспективе станет пользователем новой системы, в модели вариантов использования создается действующее лицо с таким же наименованием. В состав действующих лиц включаются также внешние системы, играющие в бизнес-процессах пассивную роль источников информации.
* Варианты использования для данного действующего лица создаются на основе анализа обязанностей соответствующего исполнителя (в простейшем случае для каждой операции исполнителя создается вариант использования, реализующий данную операцию в системе).

Такая начальная версия модели описывает минимальный вариант системы, пользователями которой являются только исполнители бизнес-процессов. Если в дальнейшем в процессе развития системы ее непосредственными пользователями будут становиться действующие лица бизнес-процессов, то модель вариантов использования будет соответствующим образом модифицироваться.

Применение данных правил будет проиллюстрировано в следующем примере.

### ПРИМЕР СПЕЦИФИКАЦИИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

В данном подразделе рассматривается спецификация требований к системе регистрации для учебного заведения, бизнес-модель которого описана выше.

**Уточненная постановка задачи для системы**

Перед руководителем информационной службы университета ставится задача разработки автоматизированной системы регистрации студентов на дополнительные платные курсы. Система должна позволять студентам регистрироваться на курсы и просматривать свои табели успеваемости с персональных компьютеров, подключенных к локальной сети университета. Профессора должны иметь доступ к системе, чтобы указать курсы, которые они будут читать, и проставить оценки за курсы.

Из-за недостатка средств университет не в состоянии заменить всю существующую систему. Остается функционировать в прежнем виде база данных, содержащая всю информацию о курсах (каталог курсов). Эта база данных поддерживается реляционной СУБД. Новая система будет работать с существующей БД в режиме доступа, без обновления.

В начале каждого семестра студенты могут запросить каталог курсов, содержащий список курсов, предлагаемых в данном семестре. Информация о каждом курсе должна включать имя профессора, наименование кафедры и требования к предварительному уровню подготовки (прослушанным курсам).

Новая система должна позволять студентам выбирать 4 курса в предстоящем семестре. В дополнение к этому каждый студент может указать 2 альтернативных курса на тот случай, если какой-либо из выбранных им курсов окажется уже заполненным или отмененным. На каждый курс может записаться не более 10 и не менее 3 студентов (если менее 3, то курс будет отменен). В каждом семестре существует период времени, когда студенты могут изменить свои планы (добавить или отказаться от выбранных курсов). После того, как процесс регистрации некоторого студента завершен, система регистрации направляет информацию в расчетную систему, чтобы студент мог внести плату за семестр. Если курс окажется заполненным в процессе регистрации, студент должен быть извещен об этом до окончательного формирования его личного учебного плана.

В конце семестра студенты должны иметь доступ к системе для просмотра своих электронных табелей успеваемости. Поскольку эта информация конфиденциальная, система должна обеспечивать ее защиту от несанкционированного доступа.

Таблица 7.2 - Глоссарий проекта

|  |  |
| --- | --- |
| ***Глоссарий проекта*** | |
| **Термин** | **Значение** |
| **Курс** | Учебный курс, предлагаемый университетом. |
| **Конкретный курс** | Конкретное чтение данного курса в конкретном семестре (один и тот же курс может вестись в нескольких параллельных сессиях). Включает точные дни недели и время. |
| **Каталог курсов** | Полный каталог всех курсов, предлагаемых университетом. |
| **Расчётная система** | Система обработки информации об оплате за курсы. |
| **Оценка** | Оценка, полученная студентом за конкретный курс. |
| **Профессор** | Преподаватель университета. |
| **Табель успеваемости** | Все оценки за все курсы, полученные студентом в данном семестре. |
| **Список курса** | Список всех студентов, записавшихся на конкретный курс. |
| **Студент** | Личность, проходящая обучение в университете. |
| **Учебный график** | Курсы, выбранные студентом в текущем семестре. |

**Описание дополнительных спецификаций**

*Функциональные возможности:*

* Система должна обеспечивать многопользовательский режим работы.
* Если конкретный курс оказывается заполненным в то время, когда студент формирует свой учебный график, включающий данный курс, то система должна известить его об этом.

*Удобство использования:*

• Пользовательский интерфейс должен быть Windows-совместимым.

*Надежность.*

*•* Система должна быть в работоспособном состоянии 24 часа в день 7 дней в неделю, время простоя - не более 10%.

*Производительность.*

• Система должна поддерживать до 2000 одновременно работающих с центральной базой данных пользователей и до 500 пользователей, одновременно работающих с локальными серверами.

*Безопасность.*

* Система не должна позволять студентам изменять любые учебные графики, кроме своих собственных, а также не должна позволять профессорам модифицировать конкретные курсы, выбранные другими профессорами.
* Только профессора имеют право ставить студентам оценки.
* Только регистратор может изменять любую информацию о студентах.

*Проектные ограничения.*

*•* Система должна быть интегрирована с существующей системой каталога курсов, функционирующей на основе реляционной СУБД.

**Создание начальной версии модели вариантов использования**

***Действующие лица:***

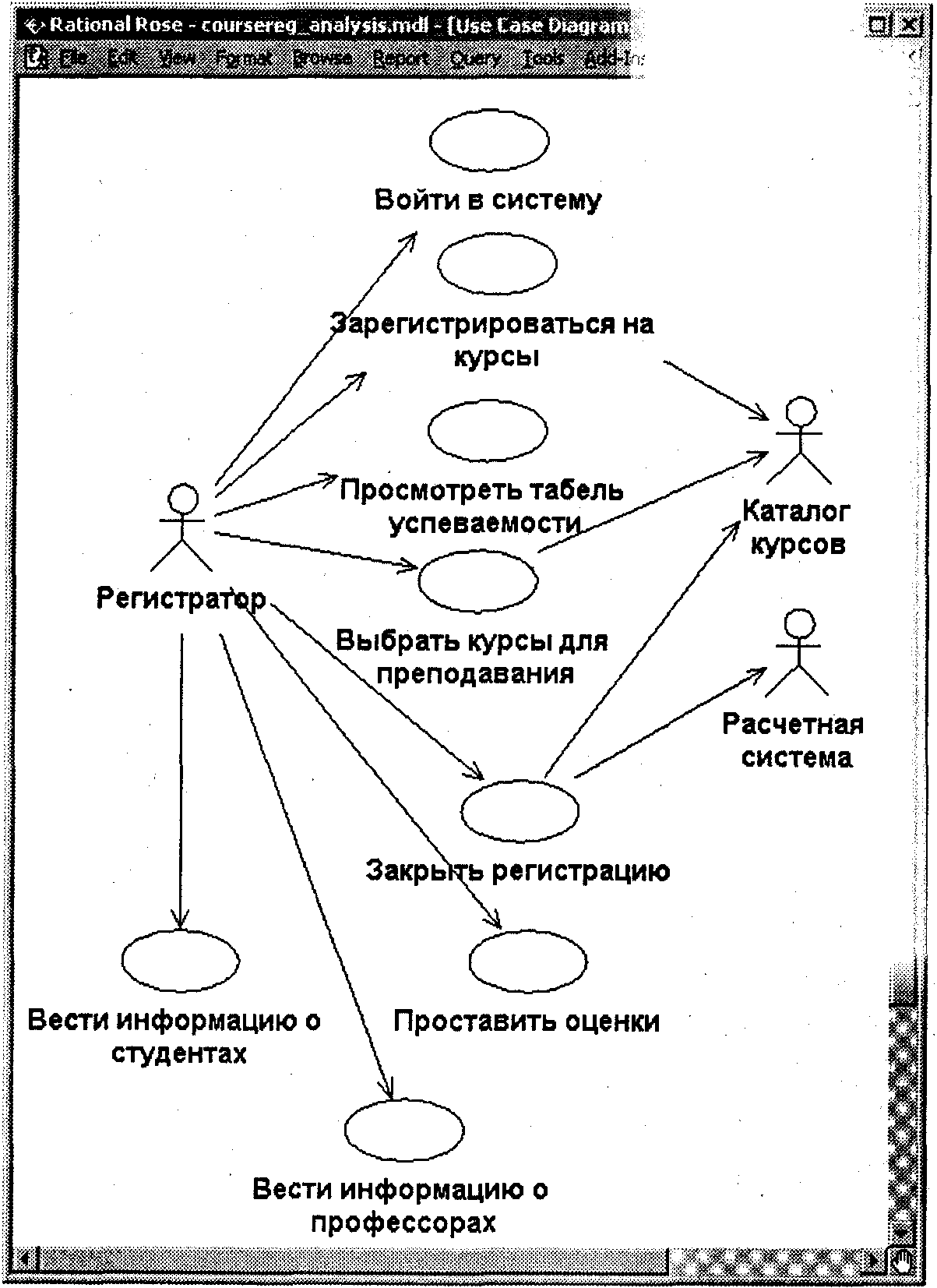
* Регистратор — формирует учебный план и каталог курсов, записывает студентов на курсы, ведет все данные о курсах, профессорах, успеваемости и студентах.
* Расчетная система — получает от данной системы информацию по оплате за курсы.
* Каталог курсов — база данных, содержащая информацию о курсах.

***Варианты использования.***

Исходя из потребностей действующих лиц, выделяются следующие варианты использования:

* Войти в систему;
* Зарегистрировать студента на курсы;
* Вывести табель успеваемости;
* Назначить курсы для преподавания;
* Проставить оценки;
* Вести информацию о профессорах;
* Вести информацию о студентах;
* Закрыть регистрацию.

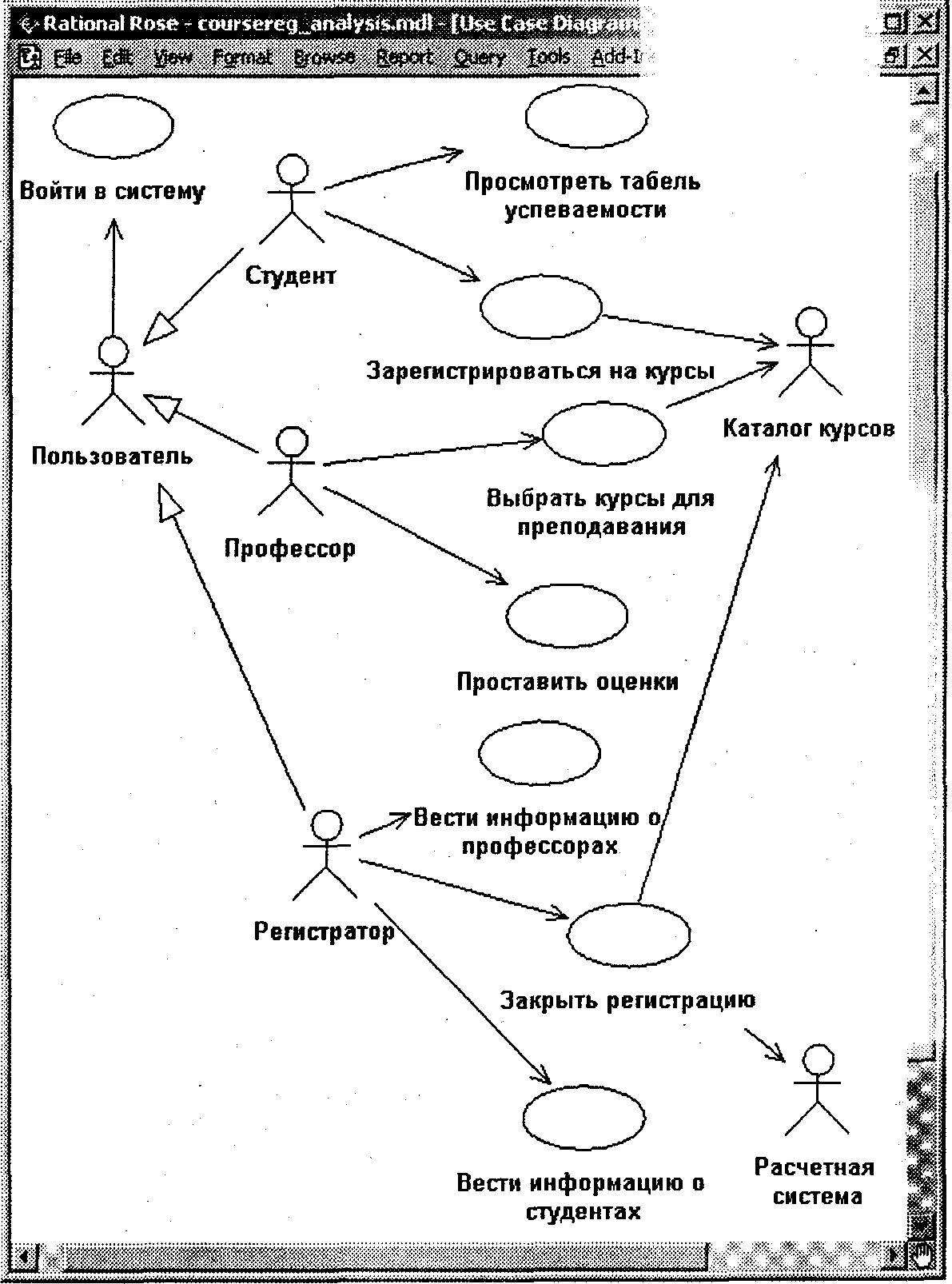
Начальная версия диаграммы вариантов использования показана на рисунке 7.1.



**Рисунок 7.1 -** Начальная версия диаграммы вариантов использования

***Модификация модели вариантов использования***

Согласно постановке задачи в состав пользователей системы следует ввести студентов и профессоров. При этом в описание действующих лиц и вариантов использования вносятся изменения. Модифицированная версия диаграммы вариантов использования показана на рисунке 7.2. Поскольку вход в систему полностью одинаков для регистратора, студента и профессора, их поведение можно обобщить и ввести новое действующее «Пользователь» (супертип) с общим вариантом использования «Войти в систему», подтипами которого являются Регистратор, Студент и Профессор.



**Рисунок 7.2 -** Модифицированная диаграмма вариантов использования для системы регистрации

***Действующие лица:***

* Студент - записывается на курсы и просматривает табель успеваемости.
* Профессор — выбирает курсы для преподавания и ставит оценки.
* Регистратор - формирует учебный план и каталог курсов, ведет все данные о курсах, профессорах и студентах.
* Расчетная система — получает от данной системы информацию об оплате за курсы.
* Каталог курсов - база данных, содержащая информацию о курсах.

***Варианты использования:***

* Войти в систему;
* Зарегистрироваться на курсы;
* Просмотреть табель успеваемости;
* Выбрать курсы для преподавания;
* Проставить оценки;
* Вести информацию о профессорах;
* Вести информацию о студентах;
* Закрыть регистрацию.

***Примеры спецификаций вариантов использования* Вариант использования «Войти в систему».**

*Краткое описание.*

Данный вариант использования описывает вход пользователя в систему регистрации курсов.

*Основной поток событий.*

Данный вариант использования начинает выполняться, когда пользователь хочет войти в систему регистрации курсов.

1. Система запрашивает имя пользователя и пароль.
2. Пользователь вводит имя и пароль.
3. Система подтверждает имя и пароль, после чего открывается доступ в систему.

*Альтернативные потоки. Неправильное имя/пароль.*

Если во время выполнения основного потока обнаружится, что пользователь ввел неправильное имя и/или пароль, система

Моделирование бизнес-процессов и спецификация требований выводит сообщение об ошибке. Пользователь может вернуться к началу основного потока или отказаться от входа в систему, при этом выполнение варианта использования завершается.

*Предусловия.*

Отсутствуют.

*Постусловия.*

Если вариант использования выполнен успешно, пользователь входит в систему. В противном случае состояние системы не изменяется.

**Вариант использования «Зарегистрироваться на курсы»:**

*Краткое описание.*

Данный вариант использования позволяет студенту зарегистрироваться на предлагаемые курсы в текущем семестре. Студент может изменить свой выбор (обновить или удалить курсы), если изменение выполняется в установленное время в начале семест­ра. Система каталога курсов предоставляет список всех конкретных курсов текущего семестра.

*Основной поток событий.*

Данный вариант использования начинает выполняться, когда студент хочет зарегистрироваться на предлагаемые курсы или изменить свой график курсов.

1. Система запрашивает требуемое действие (создать график, обновить график, удалить график).
2. Когда студент указывает действие, выполняется один из подчиненных потоков (создать, обновить, удалить или принять график).

*Создать график.*

1. Система выполняет поиск в каталоге доступных предлагаемых курсов и выводит их список.
2. Студент выбирает из списка 4 основных курса и 2 альтернативных курса.
3. После выбора система создает график студента.
4. Выполняется подчиненный поток «Принять график».

*Обновить график.*

1. Система выводит текущий график студента.
2. Система выполняет поиск в каталоге доступных предлагаемых курсов и выводит их список.
3. Студент может обновить свой выбор курсов, удаляя или добавляя предлагаемые курсы.
4. После выбора система обновляет график.
5. Выполняется подчиненный поток «Принять график».

*Удалить график.*

1. Система выводит текущий график студента.
2. Система запрашивает у студента подтверждения удаления графика.
3. Студент подтверждает удаление.
4. Система удаляет график. Если график включает предлагаемые курсы, на которые записался студент, он должен быть удален из списков этих курсов.

*Принять график.*

Для каждого выбранного, но еще не «зафиксированного» предлагаемого курса в графике система проверяет выполнение студентом предварительных требований (прохождение определенных курсов), факт открытия предлагаемого курса и отсутствие конфликтов графика. Затем система добавляет студента в список выбранного предлагаемого курса. Курс фиксируется в графике и график сохраняется в системе.

*Альтернативные потоки.*

*Сохранить график.*

В любой момент студент может вместо принятия графика сохранить его. В этом случае шаг «Принять график» заменяется на следующий:

1. «Незафиксированные» предлагаемые курсы помечаются в графике как «выбранные».
2. График сохраняется в системе.

*Не выполнены предварительные требования, курс заполнен или имеют место конфликты графика.*

Если во время выполнения подчиненного потока «Принять график» система обнаружит, что студент не выполнил необходимые предварительные требования или выбранный им предлагаемый курс заполнен, или имеют место конфликты графика, то вы­дается сообщение об ошибке. Студент может либо выбрать другой предлагаемый курс и продолжить выполнение варианта использования, либо сохранить график, либо отменить операцию, после чего основной поток начнется с начала.

*График не найден.*

Если во время выполнения подчиненных потоков «Обновить график» или «Удалить график» система не может найти график студента, то выдается сообщение об ошибке. После того, как студент подтвердит это сообщение, основной поток начнется с начала.

*Система каталога курсов недоступна.*

Если окажется, что невозможно установить связь с системой каталога курсов, то будет выдано сообщение об ошибке. После того, как студент подтвердит это сообщение, вариант использования завершится.

*Регистрация на курсы закончена.*

Если в самом начале выполнения варианта использования окажется, что регистрация на текущий семестр закончена, будет выдано сообщение, и вариант использования завершится.

*Удаление отменено.*

Если во время выполнения подчиненного потока «Удалить график» студент решит не удалять его, удаление отменяется, и ос­новной поток начнется с начала.

***Предусловия.***

Перед началом выполнения данного варианта использования студент должен войти в систему.

*Постусловия.*

Если вариант использования завершится успешно, график студента будет создан, обновлен или удален. В противном случае состояние системы не изменится.

**Вариант использования «Закрыть регистрацию».**

*Краткое описание.*

Данный вариант использования позволяет регистратору закрывать процесс регистрации. Предлагаемые курсы, на которые не записалось достаточного количества студентов (менее трех), отменяются. В расчетную систему передается информация о каж­дом студенте по каждому предлагаемому курсу, чтобы студенты могли внести оплату за курсы.

*Основной поток событий.*

Данный вариант использования начинает выполняться, когда регистратор запрашивает прекращение регистрации.

1. Система проверяет состояние процесса регистрации. Если регистрация еще выполняется, выдается сообщение и вариант использования завершается.
2. Для каждого предлагаемого курса система проверяет, ведет ли его какой-либо профессор и записалось ли на него не менее трех студентов. Если эти условия выполняются, система фиксирует предлагаемый курс в каждом графике, который включает данный курс.
3. Для каждого студенческого графика проверяется наличие в нем максимального количества основных курсов; если их недостаточно, система пытается дополнить альтернативными курсами из списка данного графика. Выбирается первый доступный альтернативный курс. Если таких курсов нет, то никакое дополнение не происходит.
4. Система закрывает все предлагаемые курсы. Если в каком-либо предлагаемом курсе оказывается менее трех студентов (с учетом добавлений, сделанных в п.З), система отменяет его и исключает из каждого содержащего его графика.
5. Система рассчитывает плату за обучение для каждого студента в текущем семестре и направляет информацию в расчетную систему. Расчетная система посылает студентам счета для оплаты с копией их окончательных графиков.

*Альтернативные потоки.*

*Предлагаемый курс никто не ведет.*

Если во время выполнения основного потока обнаруживается, что некоторый курс не ведется никаким профессором, то этот курс отменяется. Система исключает данный курс из каждого со­держащего его графика.

*Расчетная система недоступна.*

Если невозможно установить связь с расчетной системой, через некоторое установленное время система вновь попытается связаться с ней. Попытки будут повторяться до тех пор, пока связь не установится.

*Предусловия.*

Перед началом выполнения данного варианта использования регистратор должен войти в систему.

*Постусловия.*

Если вариант использования завершится успешно, регистрация закрывается. В противном случае состояние системы не изменится.

## ! Следует запомнить.

1. Для выявления требований используются (в различных сочетаниях) следующие методы:
   * собеседование (интервьюирование);
   * анкетирование;
   * моделирование и анализ бизнес-процессов;
   * сессии по выявлению требований (мозговой штурм);
   * создание и демонстрация пользователям работающих прототипов приложений (для выявления замечаний и дополнительных требований).

## S Основные понятия

Бизнес-процесс, бизнес-модель, бизнес-правила, действующее лицо бизнес-процессов, вариант использования с точки зрения бизнес-процессов, бизнес-объект.

## ? Вопросы для самоконтроля

1. Что такое бизнес-процесс и бизнес-модель?
2. Что дает построение бизнес-моделей и какие проблемы с ним связаны?
3. Охарактеризуйте достоинства и область применения методики моделирования Rational Unified Process.
4. Какие проблемы могут возникнуть при спецификации требований в случае отсутствия бизнес-моделей?

1. Методической основой данного подраздела является технология Rational Unified Process. [↑](#footnote-ref-1)
2. *Коберн А.* Современные методы описания функциональных требова­ний к системам: Пер. с англ. - М.: ЛОРИ, 2002. [↑](#footnote-ref-2)