



# ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ CASE-технологии

ISO/IEC 12207  
IEEE/EIA 12207  
Software Life Cycle Processes  
Supporting Life Cycle Processes

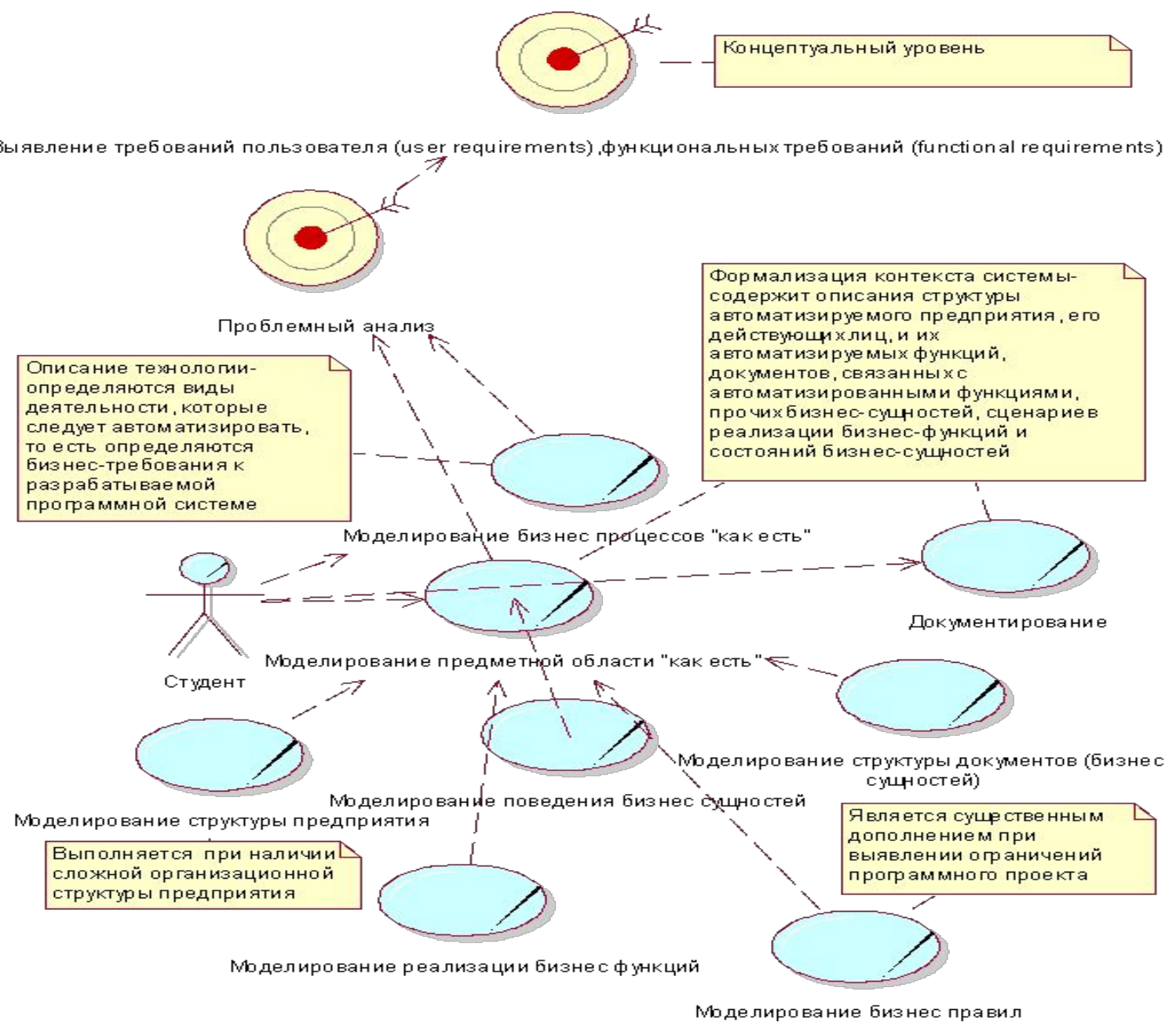
## Лекция \_8\_9

1. Проблемный анализ объекта автоматизации. Концептуальная диаграмма прецедентов
2. Технологический процесс документирования (артефакты объектно-ориентированного бизнес-анализа)
3. Артефакты
4. Анализ и проектирование. Модель «Требования»
5. Модель прецедентов. Диаграммы прецедентов
6. Модель прецедентов. Диаграммы прецедентов. Расширения UML для моделирования бизнес-систем
7. Диаграммы прецедентов. Нотация
8. Диаграммы прецедентов. Сценарий.
9. Рекомендации для написания сценариев реализации прецедентов/вариантов использования
10. Примеры написания сценариев реализации прецедентов/вариантов использования



# Проблемный анализ объекта автоматизации. Концептуальная диаграмма прецедентов

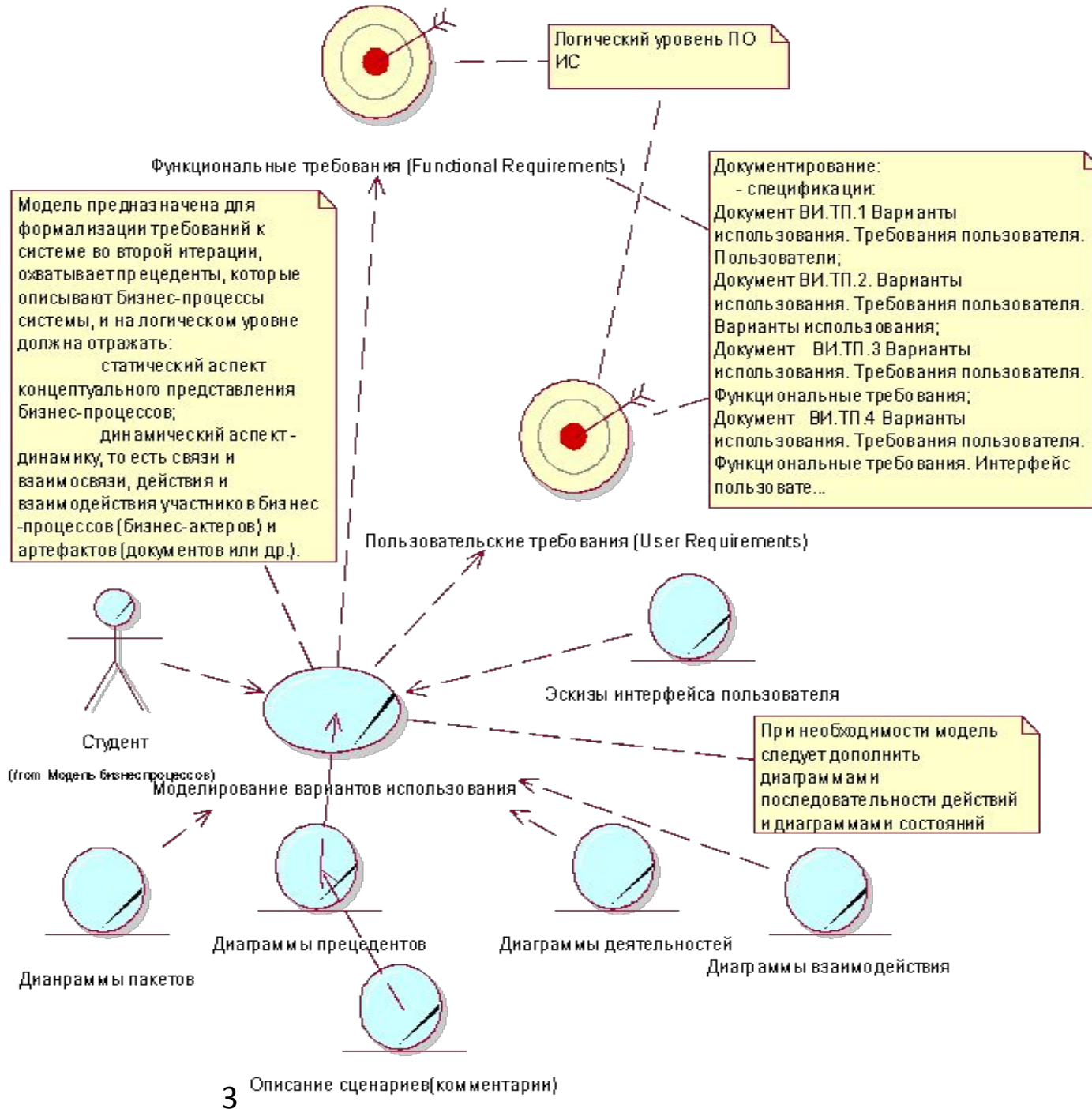
Выявление требований пользователя (user requirements), функциональных требований (functional requirements)



## Идентифицированное подмножество UML разработки бизнес-метамоделей

Мета-класс UML	Стереотипы
Actor	Business Actor
Class	Business Entity, Business Goal, Business Worker, Case Worker
Collaboration	Business Use Case Realization
Constraint	Business Rule
Dependency	owner, supports
Model	Business Use Case Model, Business Analysis Model
Package	Business System
Signal	Business Event
Use Case	Business Use Case

# Проблемный анализ объекта автоматизации. Модель анализа (предметной области)



# Технологический процесс документирования

## Документ Г1.1 Границы проекта. Заинтересованные

Лица		Конкретизация
Имя		
Именование		
Идентификатор		
	Профиль лица* (описание)	
	Классы пользователя	
	Привилегированность	

## Документ Г1.2 Границы проекта. Основные функции

Заинтересованное лицо	Функция	Конкретизация
	Именование	
	Идентификатор	
	Описание	
	Потребности*	
	Конкурирующий продукт	
	Отличие и преимущество	
	Внешний вид	
	Атрибуты качества	
	Приоритет	

# Технологический процесс документирования

Документ ВИ.ТП.1 Варианты использования. Требования пользователя. Пользователи

<b>Заинтересованное лицо (Пользователь)</b>	<b>Конкретизация</b>	<b>Примечание</b>
Наименование		
Идентификатор (ID)		
Профиль лица* (описание)		
Характерное	<i>Типичный представитель (возраст, образование, потребности)</i>	
Уникальное (свойства, атрибуты)	<i>Описание (отличительные черты)</i>	
Класс пользователя	<i>Мера ответственности, права доступа, дополнительно</i>	

# Технологический процесс документирования

Документ ВИ.ТП.1 Варианты использования.

ЗЛ	ID			
		<b>Бизнес-прецедент</b>		<b>Конкретизация</b>
		Наименование	ID	
			Описание	
			Потребности*	
			Конкурирующий продукт	
			Отличие и преимущество	
			Внешний вид	
			Атрибуты качества	
			Приоритет	

# Технологический процесс документирования

Документ ВИ.ТП.2. Варианты использования. Требования пользователя. Варианты использования

Пользователь (Заинтересованное лицо)		Цель		Бизнес прецедент			Конкретизация	Примечание (Итерация, версия)
Наименование	Идентификатор (ID)	Наименование	Идентификатор цели (ID)	Наименование бизнес прецедента	Идентификатор бизнес прецедента (ID)			
		<i>основная задача, решаемая задача</i>		<i>Наименование бизнес прецедента (Задача... глагол ... объект ...)</i>				
						<i>Сценарий бизнес прецедента</i>		
						<i>Предусловия</i>		
						<i>Основной поток событий</i>		
						<i>Альтернативный поток событий</i>		
						<i>Постусловия</i>		
						<i>Бизнес-правила</i>		



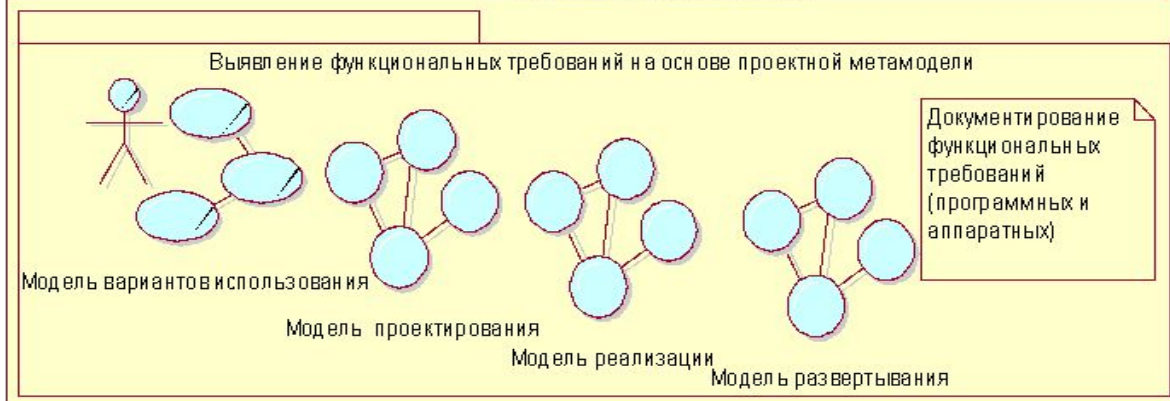
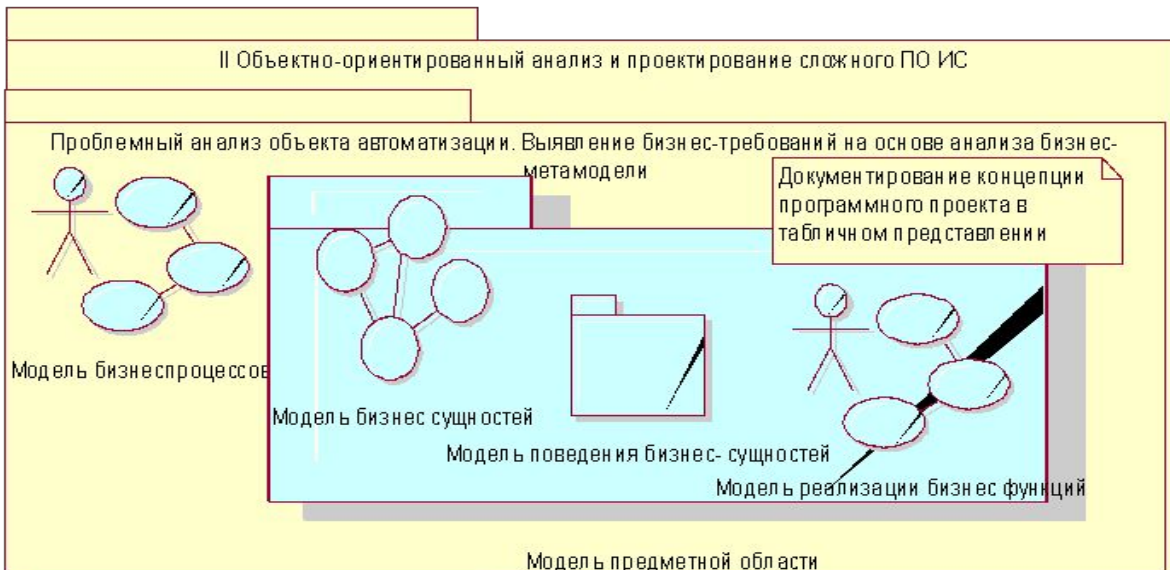
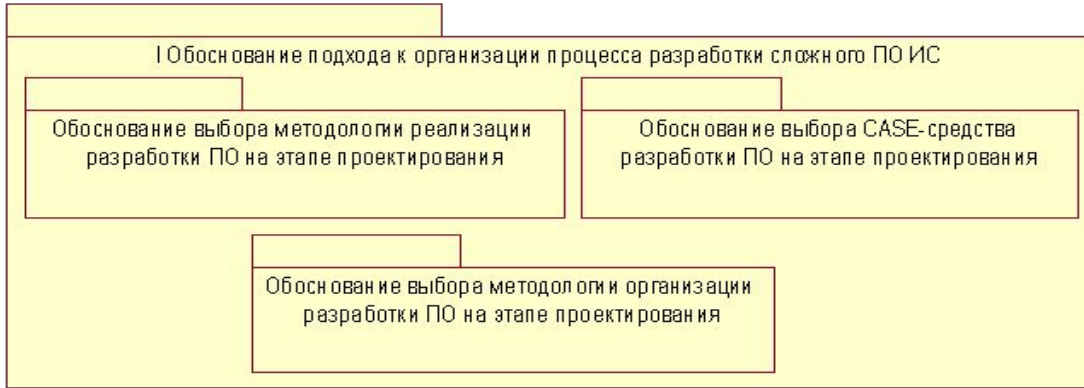
# Технологический процесс документирования

Документ ВИ.ТП.2. Варианты использования. Требования пользователя. Варианты

Использование (Заинтересованное лицо)		Цель		Вариант использования		Конкретизация	Примечание (Итерация, версия)
Наименование	Идентификатор (ID)	Наименование	Идентификатор цели (ID)	Наименование варианта использования	Идентификатор бизнес прецедента (ID)		
		<i>основная задача, решаемая задача</i>		<i>Наименование варианта использования (Задача... глагол ... (Что будет делать Пользователь с ИС?) объект ... (Для чего Пользователю ИС ?)</i>			
						Сценарий варианта использования	
						– Предусловия	
						– Основной поток событий	
						– Альтернативный поток событий	
						– Постусловия	
						– Бизнес-правила	
						Атрибуты качества	
						– Производительность	
						– Быстродействие	
						– другие	

1. *артефакт (artifact)* – диаграмма, документ, модель, и т.д., иначе – нечто, описывающее определенное понятие предметной области. Артефакты используются как исходные данные для последующей деятельности, содержат справочные сведения о проекте или выступают в роли поставляемых по контракту составляющих;
2. *модель (model)* – самый важный вид артефактов в RUP. Имеется девять моделей, которые совместно охватывают все важнейшие решения относительно визуализации, специфицирования, конструирования и документирования программной системы:
  3. модель бизнес-процессов – формализует абстракцию организации;
  4. модель предметной области – формализует контекст системы;
  5. модель прецедентов, иначе вариантов использования – формализует функциональные требования к системе;
  6. проектная модель – формализует словарь предметной области и области решения;
  7. аналитическая модель (необязательная) – формализует идею проекта;
  8. модель процессов (необязательная) – формализует механизмы параллелизма и синхронизации в системе;
  9. модель развертывания – формализует топологию аппаратных средств, на которых выполняется система;
  10. модель реализации – описывает части, из которых собирается система;
  11. модель тестирования – формализует способы проверки и приемки системы.
12. *вид (view)* – одна из проекций модели, а именно: с точки зрения проектирования, процессов, развертывания, реализации и прецедентов.
13. другие артефакты:
  14. группа требований – описывает, что система должна делать;
  15. группа проектирования – описывает, как система должна быть построена;
  16. группа реализации – описывает сборку разработанных программных компонентов.

# Пакетное представление при выборе объектно-ориентированного подхода. Артефакты



# Анализ и проектирование. Процесс «Проектирование»

Процесс определения требований правообладателей

**Цель** состоит в выявлении требований к системе, выполнение которых может обеспечивать предоставление услуг, необходимых пользователям и другим правообладателям в заданной среде применения.

- ✓ Этот процесс позволяет определять правообладателей или классы правообладателей, которые связаны с системой на протяжении всего ее жизненного цикла, а также их потребности и пожелания.
- ✓ В рамках процесса они анализируются и преобразуются в общую совокупность требований правообладателей, которые описывают желаемое поведение системы в процессе взаимодействия со средой применения. Она служит в качестве ссылки, по отношению к которой каждая предоставляемая услуга подвергается валидации для подтверждения того, что система полностью удовлетворяет заявленным требованиям

## Выходы

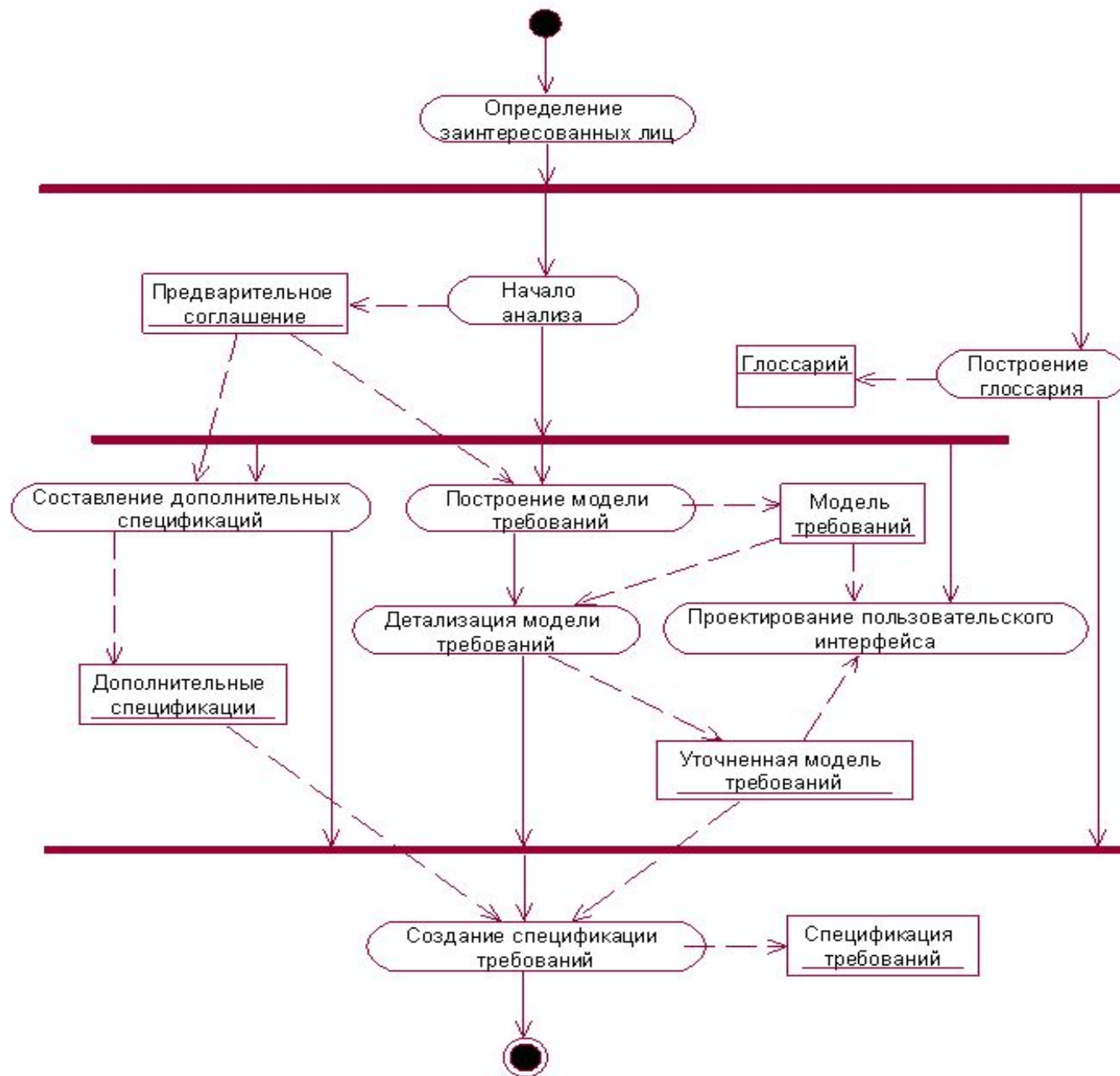
В результате успешного осуществления процесса определения требований правообладателей:

1. задаются требуемые характеристики и условия использования услуг;
2. определяются ограничения для системных решений;
3. достигается возможность прослеживания от требований правообладателей к правообладателям и их потребностям;
4. описывается основа для определения системных требований;
5. определяется основа для валидации соответствия услуг;
6. формируется основа для ведения переговоров и заключения соглашений о поставке услуги или продукции.

# Анализ и проектирование. Модель «Требования»

1. Модель требований служит для достижения соглашения между заказчиком и разработчиками, давая возможность заказчику убедиться в том, что система будет делать то, что они ожидают, а разработчикам создать то, что требуется. Модель требований позволяет, во-первых, установить иерархию требований, что способствует лучшему пониманию человеком, во-вторых, дает наглядное графическое представление требований и зависимостей между ними, в третьих позволяет связать графическую форму представления с текстовой. Модель включает актеров и ВИ, показывает, как система взаимодействует с актерами и что она делает в каждом ВИ.
2. **Модель прецедентов.**
3. **Спецификация требований** (Software Requirements Specification) – основной документ, используемый при проектировании и разработке ПС. Она включает модель требований и дополнительные спецификации, которые представляют собой текстовое описание требований к конечному продукту, но не к процессу его разработки и не содержат деталей реализации требований.
4. **Прототип пользовательского интерфейса** обеспечивает визуальное представление интерфейса пользователя с ПС.
5. **Глоссарий** – текстовый документ, содержащий определения основных понятий и терминов, которые должны одинаково пониматься заказчиком и разработчиком. Источниками данных для создания перечисленных артефактов могут, в частности, служить артефакты, созданные при бизнес-анализе (см. статью «RUP. Обследование организации (бизнес-анализ)»).

# Анализ и проектирование. Модель «Требования»



# Модель прецедентов. Диаграммы прецедентов

- Идея описания функциональных требований в виде вариантов использования (use case) была сформулирована в 1986 г. Иваром Якобсоном. Эта идея была признана конструктивной и получила широкое одобрение. Впоследствии наиболее значительный вклад в решение проблемы определения сущности вариантов использования и способов их описания внес Алистер Коберн.

Прецедент (Вариант использования) представляет собой последовательность действий (транзакций), выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом). Вариант использования описывает, что делает система, но не указывает как.

Прецедент описывает типичное взаимодействие между пользователем и системой и отражает представление о поведении системы с точки зрения пользователя (отражает цель использования системы актером/действующим лицом).

**Цель** построения диаграмм прецедентов – **документирование функциональных требований к системе** в самом общем виде, поэтому они должны быть предельно простыми.

Сами бизнес-процессы выделяются в контексте бизнес-прецедентов, или сценариев реализации бизнес-функций, иначе вариантов использования, которые формализуют функциональные требования к системе.

Модель с точки зрения вариантов использования, иначе прецедентов (Use case view) предназначена для формализации требований к системе, охватывает прецеденты, которые описывают бизнес-процессы системы, и на логическом уровне должна отражать: статический аспект концептуального представления бизнес-процессов; динамический аспект – динамику, то есть связи и взаимосвязи, действия и взаимодействия участников бизнес-процессов (бизнес-актеров) и артефактов (документов или др.).

# Модель прецедентов. Диаграммы прецедентов

- **Диаграммы вариантов использования/прецедентов (Use Case Diagrams)** служат для описания поведения целевой системы с внешней точки зрения, кроме рисования диаграмм, VP-UML позволяет подробно документировать требования по описанию прецедентов. Все эти данные могут быть выведены в HTML // PDF // формат MS Word.

При разработке бизнес-метамоделей следует использовать Rational UML profile (VP-UML), который предоставляет UML-язык для записи бизнес-моделей являясь компонентом Rational Unified Process® (RUP®).

- **Предназначение профиля – разрешить использование UML-средств для разработки бизнес-систем.** Профиль формирует семантику взаимодействия между UML-средствами и другими средствами моделирования бизнес-систем

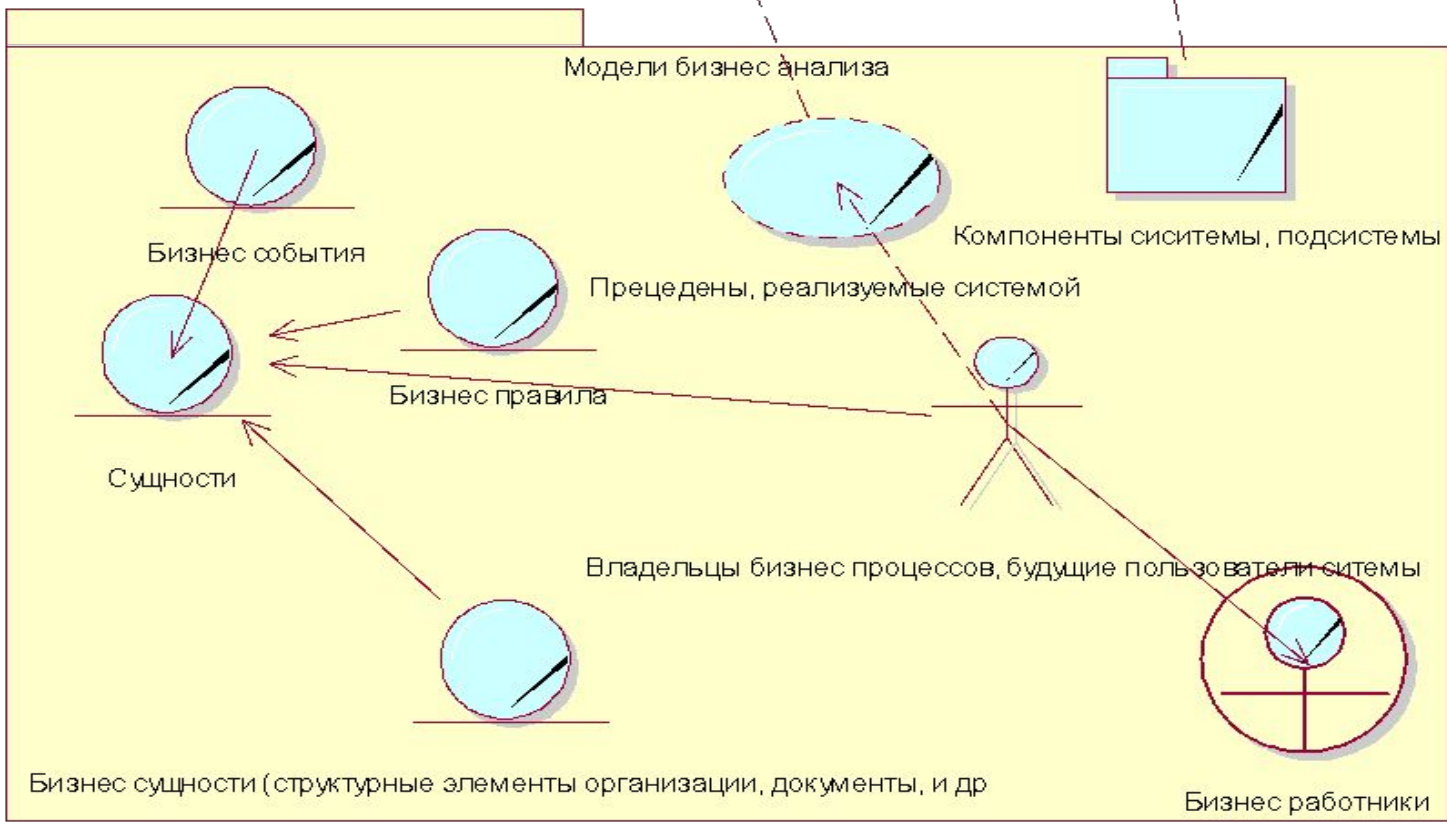
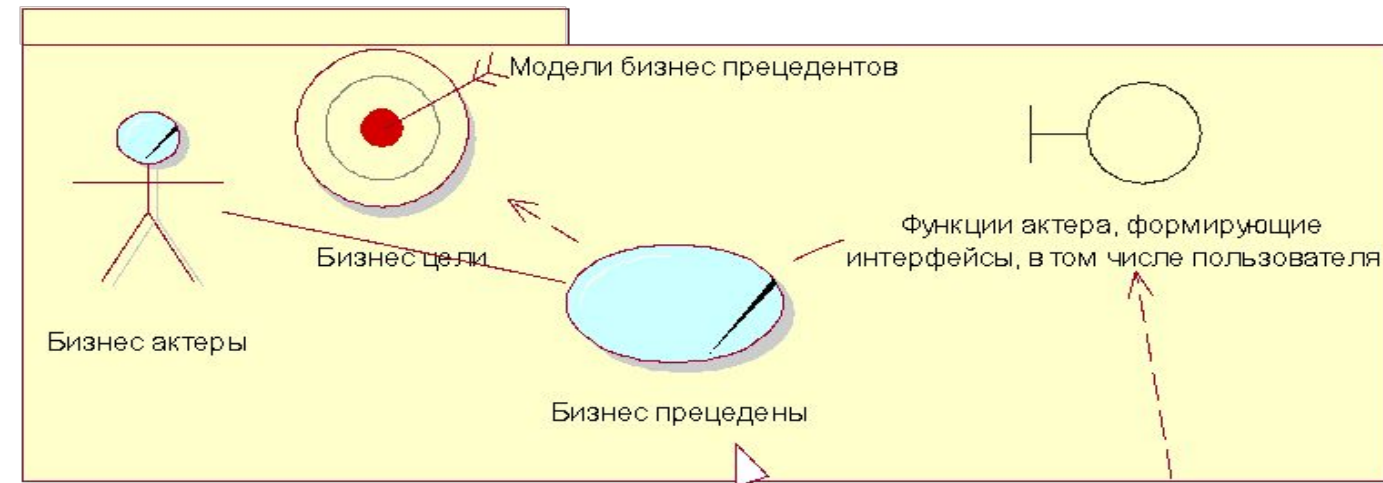


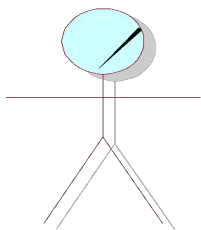
# Модель прецедентов. Диаграммы прецедентов. SWEBOOK®

## Расширения UML для моделирования бизнес-систем

Мета-класс UML	Стереотипы	
Actor	Business Actor	Бизнес-актер
Class	Business Entity, Business Goal, Business Worker, Case Worker	Бизнес-сущность, Бизнес-цель, Бизнес-worker, Ситуативный исполнитель
Collaboration	Business Use Case Realization	Смежные бизнес-процессы (поставщик-клиент)
Constraint	Business Rule	Бизнес-правила (условия, ограничения)
Dependency	owner, supports	Владелец процесса, тот, кто осуществляет только поддержку процесса
Model	Business Use Case Model, Business Analysis Model	Модель бизнес прецедентов, Модель бизнес анализа
Package	Business System	Бизнес система
Signal	Business Event	Бизнес события
Use Case	Business Use Case	Бизнес-прецеденты

# Расширения UML для моделирования бизнес систем – концепции профиля и взаимосвязи между этими концепциями

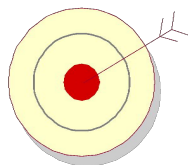




Бизнес актеры

«Стереотип» **Business Actor** расширяет «metaClass» **Actor** – определяет набор экземпляров бизнес-актера, в котором каждый экземпляр бизнес-актера играет одинаковую роль по отношению к бизнесу. Важно, что бизнес-актер представляет некоторого участника вне области действия бизнеса и, следовательно, имеет представление только внешнего видимого поведения бизнес-процесса.

Категория	Имя	Тип	Описание
Attribute	Characteristics	String	используется для более детального описания бизнес-актеров: количество индивидуумов, представленных бизнес-актером, уровень компетенций, уровень компьютерного опыта, другие общие характеристики, такие как пол, возраст, образовательная база и т.д.

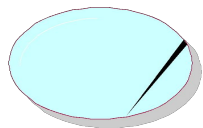


Бизнес цель

«Стереотип» Business Goal расширяет "metaClass" Class – бизнес-цель является, в сущности, требованием, которому должен удовлетворять бизнес. Такие цели направляют бизнес-операции на их удовлетворение, используя несколько механизмов, но главным образом через последовательное улучшение управления бизнес-процессами. Бизнес поддерживает много целей непосредственно, используя бизнес-прецеденты, которые выполняют одну или несколько целей. Назначением бизнес-целей является преобразование бизнес-стратегии в небольшие шаги, которыми бизнес-операции могут следовать в правильном направлении, и, если необходимо, быть улучшенными. Эти измеримые количественно мероприятия позволяют устанавливать реальные ожидания, касающиеся улучшений бизнес-процесса, и обеспечивают объективное измерение прогресса при реализации изменений и улучшений бизнес-процесса. Бизнес-менеджеры и акционеры используют бизнес-цели для трансляции бизнес-стратегии в конкретные измерения. Аналитики бизнес-процессов и бизнес-дизайнеры используют бизнес-

Категория	Цели для проверки соответствия бизнес-процессов бизнес-стратегии.		
Attribute	ChangeValue	String	Скалярная величина, на которую предполагается изменить значение.
Attribute	ChangeKind	{direct, percent}	«direct» указывает на то, что ChangeValue представляет абсолютное значение. «percent» означает относительное значение ChangeValue.
Attribute	ChangeBy	Datetime	Дата и время, когда изменение должно быть реализовано.
Attribute	Measure	String	Описание значения, используемое для проверки того, была ли достигнута цель.
Attribute	Priority	Decimal	Относительный приоритет (определенная пользователем семантика).
Attribute	ChangeValue	String	Скалярная величина, на которую предполагается изменить значение.
Attribute	ChangeKind	{direct, percent}	«direct» указывает на то, что ChangeValue представляет абсолютное значение. «percent» означает относительное значение ChangeValue.

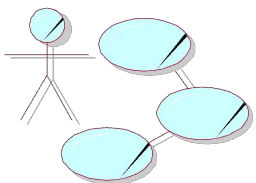
# Диаграммы прецедентов. Нотация



Бизнес прецедент

«Стереотип» Business Use Case расширяет «metaClass» UseCase – бизнес-прецедент определяет множество экземпляров бизнес-прецедентов, в котором каждый экземпляр представляет собой последовательность выполняемых бизнес-действий, которые приводят к наблюдаемому конкретным бизнес-актером результату. Класс бизнес-прецедента содержит все основные, альтернативные потоки работ, относящиеся к производству «наблюдаемого результата». Бизнес-прецедент может описывать бизнес-процесс с внешней, дополнительной точки зрения. Акционеры, аналитики бизнес-процессов и бизнес-дизайнеры используют бизнес-прецеденты для описания бизнес-процессов и понимания эффекта любых предложенных изменений работы бизнес-системы. Бизнес-прецеденты используются также системными аналитиками и проектировщиками ПО для понимания того, как программная система вписывается в организацию. Тестировщики используют бизнес-прецеденты для предоставления контекста для разработки тестовых сценариев программных систем. Менеджеры проекта используют бизнес-прецеденты для планирования содержания итераций моделирования бизнес-системы и слежения за развитием.

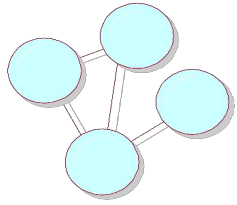
Категория	Имя	Тип	Описание
AssociationEnd	Category	Process Category	Принадлежит ли бизнес-прецедент категории 'core', 'supporting' или 'management'.
Attribute	Possibilities	String	Описание потенциала улучшения бизнес-прецедента.
Attribute	Risks	String	Спецификация рисков выполнения
Attribute	Special Requirements	String	Характеристики бизнес-прецедента, не охваченные потоком работ, который был описан.
Attribute	Workflow	String	Текстовое описание потока работ, который представляет бизнес-прецедент. Поток должен описывать, что делает бизнес-система для доставки ценности бизнес-актеру, а не то, как она решает свои проблемы. Описание должно быть понятным каждому участнику бизнес-системы.



Модель бизнес прецедентов

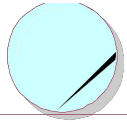
«Стереотип» Business Use Case Model расширяет «metaClass» Model – модель бизнес-прецедентов – это модель бизнес-целей и предполагаемых функций. Она используется как важная входная информация для идентификации ролей и результатов деятельности в организации. Business Use Case Model описывает направление и намерение бизнес-системы. Business Use Case Model используется акционерами, аналитиками бизнес-процессов и бизнес-дизайнерами для осмысления и улучшения способа взаимодействия бизнеса с его средой, а также системными аналитиками и проектировщиками программного обеспечения для предоставления контекста для разработки ПО. Менеджер проекта использует Business Use Case Model для планирования содержимого итераций во время моделирования бизнеса и для отслеживания развития.

Категория	Имя	Тип	Описание
Attribute	SurveyDescription	String	Текстовое описание, содержащее информацию, не отраженную в Business Use Case Model, включая типичную последовательность, в которой бизнес-прецеденты применяются пользователями, и функциональность, не управляемую моделью бизнес-прецедента.



Модель бизнес анализа

**«Стереотип» Business Use Case Model расширяет «metaClas» Model – модель бизнес-прецедентов** – это модель бизнес-целей и предполагаемых функций. Она используется как важная входная информация для идентификации ролей и результатов деятельности в организации. Business Use Case Model описывает направление и намерение бизнес-системы. Business Use Case Model используется акционерами, аналитиками бизнес-процессов и бизнес-дизайнерами для осмысления и улучшения способа взаимодействия бизнеса с его средой, а также системными аналитиками и проектировщиками программного обеспечения для предоставления контекста для разработки ПО. Менеджер проекта использует Business Use Case Model для планирования содержимого итераций во время моделирования бизнеса и для отслеживания развития.

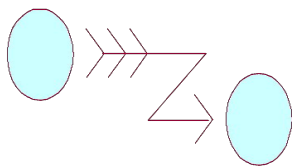


Бизнес-сущность

«Стереотип» **Business Entity** расширяет **Abstract Entity** – **бизнес-сущность** представляет значительную и постоянную часть информации, управляемой бизнес-актерами и исполнителями. Бизнес-сущности пассивны, то есть, они не иницируют взаимодействия сами по себе. Бизнес-сущность может быть использована во множестве различных реализаций бизнес-прецедентов и обычно реагирует на любое единичное взаимодействие. Бизнес-сущности обеспечивают основу для разделения информации (потока документов) среди исполнителей, участвующих в различных реализациях бизнес-прецедентов. Бизнес-сущности представляют абстракцию важной постоянной информации в бизнес-системе. Любая часть информации, являющаяся свойством чего-нибудь еще, вероятно не является бизнес-сущностью в действительности. Информация, которая не сохраняется, но создается или определяется по требованию (когда необходимо), тоже не является бизнес-сущностью. Например, список продуктов – это определенно важная информация, но это – не устойчивая информация. Каждый раз, когда кто-нибудь хочет знать количество экземпляров конкретного штрих-кода, находящееся в настоящее время на полке (или на складе), эта информация будет вычислена и затем сброшена. Акционеры используют бизнес-сущности для проверки того, что информация созданная и требуемая организацией присутствует в Business Analysis Model. Бизнес-дизайнер отвечает за идентификацию и описание бизнес-сущностей, а также за определение влияния организационных изменений на информацию, созданную и требуемую бизнес-системой. Бизнес-сущности также используются системными аналитиками и дизайнерами при описании системных прецедентов и идентификации программных сущностей соответственно.



## Диаграммы прецедентов. Нотация

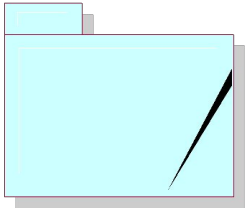


Бизнес событие

«Стереотип» **Business Event** расширяет «metaClass» **Signal** – **бизнес-событие** описывает значимое явление в пространстве и времени, важное для бизнес-системы. Бизнес-события используются для оповещения между бизнес-процессами и обычно связаны с бизнес-сущностями. Как необязательный элемент RUP бизнес-события полезны при синхронизации, взаимодействии или интеграции бизнес-функций, приложений или месторасположений. Бизнес-события не обязательны в том случае, когда бизнес-процессы и бизнес-сущности не моделируются. Бизнес-события используются для явного определения значимых явлений во время повседневных операций бизнес-системы. Акционеры и аналитики бизнес-процессов используют бизнес-события для лучшего понимания и описания бизнес-операций. Бизнес-дизайнеры отвечают за детализацию бизнес-событий и используют их для разъединения пространства и времени внутри бизнес-процессов. Бизнес-события используются также системными аналитиками при идентификации актеров программной системы и прецедентов, а также проектировщиками программного обеспечения для того, чтобы сделать программные системы более гибкими и ремонтпригодными. «Business Event» имеет ассоциацию с «Business Entity».

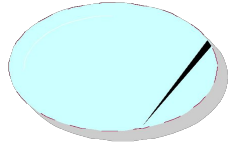
«Стереотип» **Business Rule** расширяет «metaClass» **Constraint** – **бизнес-правило** представляет собой объявление политики или условия, которое должно быть удовлетворено, и выражается в виде ограничения или инварианта в Business Analysis Model, поэтому отсутствует графическая нотация. Бизнес-правила должны использоваться в тех ситуациях, когда существуют многочисленные или сложные условия, управляющие бизнес-операциями. Такие правила могут быть выражены на естественном языке, например, «заказ должен иметь назначенного заказчика» или, возможно на формальном языке, например OCL (Object Constraints Language). Назначением этого артефакта является определение конкретного ограничения или инварианта, которые должны быть удовлетворены бизнес-системой. Акционеры и аналитики бизнес-процессов просматривают бизнес-правила для проверки того, что описания бизнес-системы соответствуют способу ведения бизнеса. Аналитики бизнес-процессов отвечают за Business Analysis Model в целом, а бизнес-дизайнеры отвечают за фиксирование бизнес-правил в модели и проверку завершенности и непротиворечивости бизнес-правил. Бизнес-правила используются также системными аналитиками и проектировщиками при определении и проектировании ПО, поддерживающего бизнес-систему.

Категория	Имя	Тип	Описание
associationEnd	Kind	Business Rule Kind	Определяет категорию бизнес-правила (используя категоризацию Рона Росса (Ron Ross)).



Бизнес-подсистема

«Стереотип» **Business System** расширяет «metaClass» **Package** – бизнес-система инкапсулирует множество ролей и ресурсов, которые все вместе выполняют конкретную задачу, а также определяет множество ответственностей, с которыми эта цель может быть достигнута. Бизнес-системы используются во многом аналогично использованию аппаратных и программных компонентов. Они определяют единицу структуры, инкапсулирующую содержащиеся в них структурные элементы, и характеризуются своими внешними видимыми свойствами. Бизнес-системы используются аналитиками бизнес-процессов для определения того, присутствуют ли внутри организации необходимые возможности, а также для проверки того, что бизнес-модель допускает изменение. Бизнес-дизайнеры используют бизнес-системы для формирования наборов связанных исполнителей и бизнес-сущностей, а также явного определения и управления зависимостями внутри организации. Менеджеры проекта также используют бизнес-системы для параллельного планирования работы.

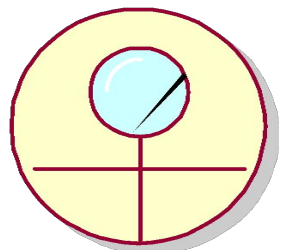


Реализация бизнес прецедента

**«Стереотип» Business Use Case Realization расширяет «metaClass» Collaboration – реализация бизнес-прецедентов** описывает, как исполнители, бизнес-сущности и бизнес-события кооперируются для выполнения конкретного бизнес-прецедента. Там, где бизнес-прецедент документирует внешне видимое поведение бизнес-системы, – предоставляется «what»; говоря о том, какие участники и сущности обеспечивают поведение прецедента, – реализация документирует «how». Тогда как бизнес-прецедент описывает шаги, которые должны быть выполнены для передачи ценности акционеру бизнеса, реализация бизнес-прецедента описывает, как эти шаги выполняются в организации. Бизнес-прецеденты описываются с точки зрения внешней перспективы, тогда как реализация бизнес-прецедента описывается с точки зрения внутренней перспективы. Реализация бизнес-прецедентов будет использоваться акционерами для проверки того, что команда проекта (или другой участник) понимает принцип работы бизнес-системы; она также используется при идентификации и выставлении приоритетов улучшений организации. Аналитики бизнес-процессов и бизнес-дизайнеры используют реализации бизнес-прецедентов для определения ролей, ответственностей и информации, требуемой внутри организации для реализации бизнес-прецедентов. При помощи реализаций бизнес-прецедентов может быть оценена эффективность изменений организации, например, автоматизации бизнес-процессов или аутсорсинга бизнес-процессов. Системные аналитики и проектировщики ПО используют реализации бизнес-прецедентов для понимания того, как программная система вписывается в организацию.

# Диаграммы прецедентов.

## Нотация



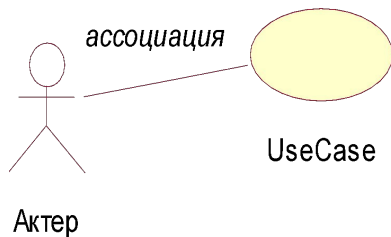
«Стереотип» **Business Worker** расширяет «metaClass» **Class – исполнитель** представляет собой абстракцию человека или программной системы, которая представляет роль, выполняемую внутри реализаций бизнес-прецедентов. Исполнитель кооперируется с другими исполнителями, получает уведомления о бизнес-событиях и управляет бизнес-сущностями для реализации их ответственностей. Исполнитель используется для представления роли, которую человек или программная система будут играть в организации. Эта абстракция позволяет идентифицировать потенциальные улучшения бизнес-процессов и оценить эффективность автоматизации бизнес-процессов или аутсорсинга бизнес-процессов. Акционеры используют исполнителей для подтверждения того, что ответственности и взаимодействия исполнителя корректно отражают способ выполнения работы, а также при оценке эффективности изменений в организации (например, автоматизация бизнес-процессов). Бизнес-дизайнер проверяет, что все потоки работ реализаций бизнес-прецедентов распределены исполнителям. Исполнители также полезны для системных аналитиков при идентификации актеров программной системы и прецедентов, а также при составлении требований к ПО.

Категория	Имя	Тип	Описание
Attribute	Characteristics	String	Содержит специфическую информацию об исполнителе. Например, концентрирует внимание на конкретных требованиях исполнителя.

# Диаграммы прецедентов.

## Нотация

1. **Ассоциация**
2. **Зависимость**
3. **Обобщение** — стрелка с незакрашенным треугольником
4. **Включение прецедента** — пунктирная стрелка со стереотипом «include»,
5. **Расширение прецедента** — пунктирная стрелка со стереотипом «extend» (стрелка входит в расширяемый прецедент, в дополнительном разделе которого может быть указана *точка расширения* и, возможно в виде комментария, условие расширения)
6. **комментарии**



- 1. Ассоциация (association)** – определяет семантические отношения, которые могут возникнуть между типизированными экземплярами. Отношение ассоциации является одним из фундаментальных понятий в языке UML. Отношение устанавливает, какую конкретную роль играет актер при взаимодействии с экземпляром варианта использования.

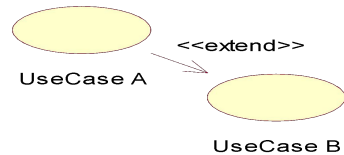
Наименование	Значение
Name	наименование отношение ассоциации
Visibility	видимость определяет, где связь появляется вновь в общей модели, и возможность ее применения
Association End From	определение начала ассоциации
Association End To..	определение конца ассоциации
Documentation	описание
Abstract	аннотированная (объявленная) ассоциация предназначена для использования в других диаграммах
Leaf	ограничение на изменение
Derived Specifies	отношение является производным от других элементов модели, таких как другие ассоциации или ограничения.

1. **Отношение зависимости (dependency)** – означает, что один или набор элементов модели требует других элементов модели для их спецификации и реализации.

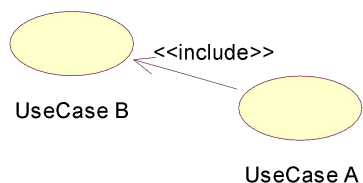
Наименование	Значение
Name	наименование варианта использования
Supplier	определяет элемент-поставщик (от которого зависит), используется например, для уточнения абстракции
Client	определяет элемент-клиент (который зависит)
Visibility	видимость определяет, где данное отношение появляется вновь в общей модели, и возможность его применения
Documentation	описание



- 1. Отношение расширения (extend)** – определяет взаимосвязь экземпляров отдельного прецедента с более общим. В метамодели отношение расширения является направленным и указывает, что применительно к отдельным примерам некоторого прецедента должны быть выполнены конкретные условия, определенные для его расширения. Так, если имеет место отношение расширения от прецедента А к В, то это означает, что свойства экземпляра прецедента В могут быть дополнены благодаря наличию свойств у расширенного варианта использования А.

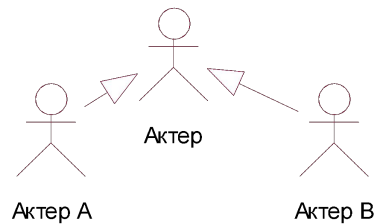


Наименование	Значение
Name	наименование отношения расширения
Extension	ссылка на прецедент, имеющий расширение
Extended Case	ссылки на прецеденты расширения
Visibility	видимость определяет, где отношения расширения появляется вновь в общей модели, и возможность его применения
Extension point	точки расширения
Condition	ссылка на условие, при котором выполняется расширение
Documentation	описание



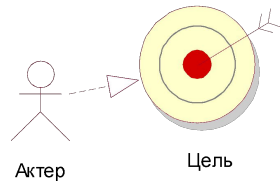
- 1. Отношение включения (include)** – между двумя прецедентами указывает, что некоторое заданное поведение для одного прецедента включается в качестве составного компонента в последовательность поведения. Один прецедент может быть включен в несколько других, а также включать в себя другие варианты. Включаемый прецедент может быть независимым от базового прецедента в том смысле, что он предоставляет последнему некоторое инкапсулированное поведение, детали реализации которого скрыты от последнего и могут быть легко перераспределены между несколькими включаемыми прецедентами. Базовый вариант может зависеть только от результатов выполнения включаемого в него поведения, но не от структуры включаемых в него прецедентов. Отношение включения, направленное от прецедента А к В, указывает, что каждый экземпляр прецедента А включает в себя функциональные свойства, заданные для В.

Наименование	Значение
Name	наименование отношения включения
Including Case	ссылка на прецедент, имеющий расширение
Addition	ссылки на прецеденты, которые должны быть включены
Visibility	видимость определяет, где отношения включение появляется вновь в общей модели, и возможность его применения
Documentation	описание



- 1. Отношение обобщения (generalization)** – служит для указания того факта, что некоторые элементы А и В, в том числе и прецеденты, могут быть обобщены. При этом общий элемент называется предком или родителем по отношению к А и В, А и В – потомками. Потомок наследует все свойства и поведение своего родителя, а также может быть дополнен новыми свойствами и особенностями поведения.

Наименование	Значение
Name	наименование отношения обобщения
General	ссылка на обобщающий элемент
Specific	ссылка на обобщаемые элементы
Visibility	видимость определяет, где отношение обобщения появляется вновь в общей модели, и возможность его применения
Documentation	описание
Substitutable	взаимозаменяемость - указывает, может ли конкретный элемент использоваться везде, где присутствует обобщенный элемент

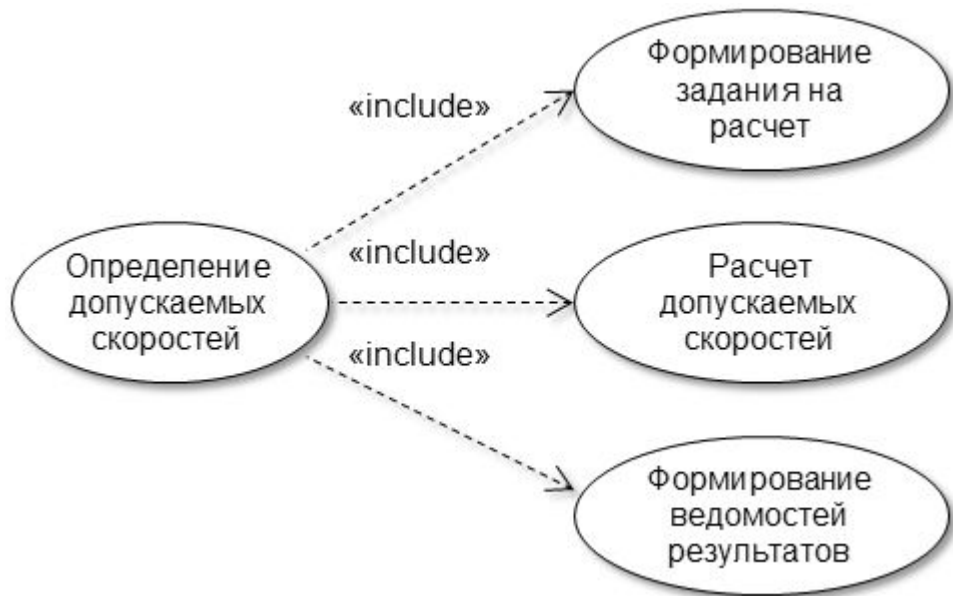


- 1. Реализация (Realization)** – специализированное отношение абстракции между двумя наборами элементов модели, один из которых представляет спецификации (поставщик), а другой представляет собой реализацию последней (клиента). Реализация может быть использована для моделирования пошагового уточнения, оптимизации, преобразования, шаблонов, моделей синтеза и т.д.

Наименование	Значение
Name	наименование отношения реализации
Supplier	определяет элемент-поставщик (который реализует),
Client	определяет элемент-клиент (который реализуется)
Visibility	видимость определяет, где отношение обобщения появляется вновь в общей модели, и возможность его применения
Documentation	описание
Mapping...	наложение условия или ограничения, от которого зависит возможность реализации варианта (не является обязательным для визуализации)

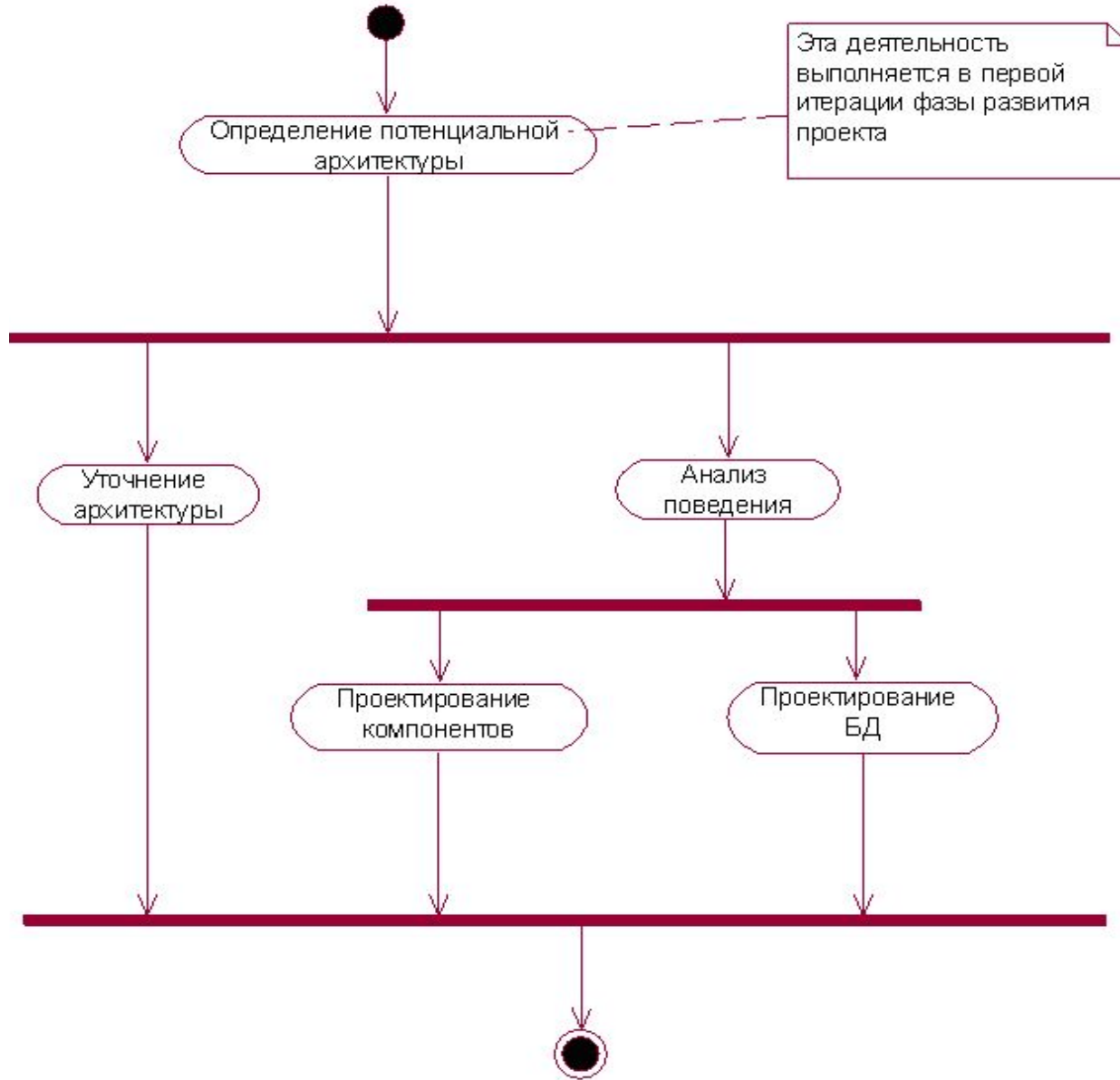
Примечания

**Примечание (комментарий) (note)** – дает возможность подключать различные замечания к элементам. Комментарий не несет семантической силы, но может содержать информацию, полезную для моделирования.



Отношение **включения**, помечаемое стереотипом <>, означает, что для полного осуществления основного (базового) ВИ необходимо выполнение и включаемого варианта. В общем случае выделение включаемых ВИ будет целесообразным в тех случаях, когда такой вариант включается в несколько базовых. Об отношении включения “знает” только базовый вариант использования, но не включаемый. Включение показывается пунктирной стрелкой, направленной от базового ВИ к включаемому

# Диаграммы прецедентов. Нотация



*Разработка Модели с точки зрения проектирования*

- *Текстовый сценарий, уточняет или детализирует последовательность действий, совершаемых системой при выполнении ее вариантов использования*

**Сценарий (scenario) – определенная последовательность действий, которая описывает действия актеров и поведение моделируемой системы в форме текста**

### **Стили описания:**

1. шаблон от Алистера Коберна (наиболее полный формат);
2. шаблон от Карла Вигерса;
3. стиль RUP;
4. стиль ICONIX;
5. стиль OpenUP;
6. свободный формат.



# Диаграммы прецедентов. Сценарий. Шаблон Алистера

**Название** <краткая фраза в виде глагола в неопределенной форме совершенного вида, отражающая цель>

**Контекст использования** <уточнение цели, при необходимости – условия ее нормального завершения>.

**Область действия** <ссылка на рамки проекта> (например – подсистема бухгалтерского учета).

**Уровень** <один из трех: обобщенный, цели пользователя, подфункции>.

**Основное действующее лицо** <имя роли основного актёра или его описание>.

**Участники и интересы** <список других актёров-участников прецедента с указанием их интересов>.

**Предусловие** (pre-condition) <то, что ожидается, уже имеет место>.

**Минимальные гарантии** <что гарантируется актерам-участникам> (Например – в случае неудавшейся транзакции все данные, имевшиеся в системе до ее начала, сохраняются неизменными).

**Гарантии** (guarantee) **успеха** описывает обязательные действия системы по окончании работы шаблона ответа. Успешные гарантии выполняются после успешного сценария; минимальные гарантии выполняются после любого сценария <что получат актёры-участники в случае успешного достижения цели>.

**Триггер** (trigger) определяет событие, инициирующее выполнение прецедента <то, что «запускает» вариант использования, обычно – событие во времени>.

**Основной сценарий** <перечисляются шаги основного сценария, начиная от триггера и вплоть до достижения гарантии успеха>.

**Формат описания:** <Номер шага> <Описание действия>

**Расширения:** <последовательно описываются все альтернативные сценарии>. Каждая из альтернатив привязана к шагу основного сценария.

**Формат описания:** <Номер шага. Номер расширения> <Условие>:<Действие или ссылка на подчиненный вариант использования>.

Любой из шагов основного сценария может иметь одно или более ветвлений. Каждое ветвление оформляется в виде расширения. В блоке «Расширения» все расширения описываются последовательно.

В случае, если альтернативный сценарий не удастся описать одной строкой – применяется следующий формат: начиная со строки, следующей после описания расширения, идет описание его действий в формате основного сценария: <Номер шага. Номер расширения. Номер шага расширения> <Действие>

**Описание расширения** заканчивается описанием выхода из расширения. Основные варианты выхода из расширения: возврат к очередному по номеру шагу основного сценария, окончание прецедента, переход к другому шагу основного сценария.

**Список изменений в технологии и данных** <что гарантируется актерам-участникам>. Например – в случае неудавшейся транзакции все данные, имевшиеся в системе до ее начала, сохраняются неизменными.

**Вспомогательная информация** <дополнительная информация, полезная при описании варианта использования>

Описание варианта использования согласно RUP можно назвать более кратким и демократичным

**Наименования и краткое описание.** В этом разделе указывается: наименование варианта использования, актеры, краткое (в один абзац) описание варианта использования.

**Поток событий.**

**Основной поток событий.** Описание «Основного сценария».

**Альтернативные потоки событий.** Каждый из альтернативных сценариев описывается в отдельном параграфе, в том же стиле, что и основной поток событий. Альтернативные сценарии описывают поведение системы при любых отклонениях от основного сценария, а также поведение в исключительных ситуациях.

**Специальные требования.** Здесь перечисляются нефункциональные требования, имеющие непосредственное отношение именно к этому варианту использования.

**Предусловия.** События, описываемые условиями или условиями, должны быть состояниями, которые пользователь может наблюдать. Предусловие описывает состояние, в котором система должна находиться до начала исполнения прецедента.

**Постусловия.** Постусловие RUP по сути описывает то же, что и минимальная гарантия у Коберна. Корректно сформулированное постусловие должно быть истинным при любом возможном сценарии прецедента, а не описанном в основном потоке.

**Точки расширения.** Данный параграф определяет положение точек, расширяющих поток событий.

Иногда представляется удобным помещать сценарии вариантов использования в таблицу. Информация при этом принимает более структурированный вид.



# Рекомендации для написания сценариев реализации прецедентов/вариантов использования

## **Имя прецедента/варианта использования.**

*Нет стандарта.*

*Рекомендуем начинать с глагола или глагольной группы (отглагольного существительного).  
(Например, Выплатить налог с оборота или Выплата налога с оборота).*

*Не смешивайте оба стиля в одной модели!*

*Является уникальным идентификатором в рамках модели.*

## **ID прецедента/варианта использования.**

*Суррогат идентификатора. Используется для облегчения и краткости при описании ссылок.*

## **Краткое описание – цель.**

*Изложение цели или сути прецедента/варианта использования.*

## **Актеры – действующие лица.**

*Каждый прецедент всегда инициируется одним действующим лицом.*

*В разные моменты времени один и тот же прецедент может быть инициирован разными действующими лицами.*

*Любое действующее лицо, которое может инициировать прецедент, является основным действующим лицом. Все остальные действующие лица – второстепенные.*

# Рекомендации для написания сценариев реализации прецедентов/вариантов использования

## **Предусловия и постусловия – ограничения.**

*Предусловия определяют, в каком состоянии должна находиться система, чтобы запуск прецедента был возможным.*

*Постусловия определяют, какие условия будут истинными после завершения прецедента.*

*Рекомендуется явно указывать на отсутствие пред- и постусловий. Например, словом «Нет».*

## **Основной поток.**

*«Идеальный» ход развития событий – нет ошибок, отклонений, прерываний или ответвлений.*

*Первый шаг рекомендуется записывать, так:*

*1. Прецедент начинается, когда <действующее лицо> <действие>.*

*Далее рекомендуется использовать следующий шаблон записи шага: <номер> <кто-либо> <совершает некоторое действие>*

# Примеры написания сценариев реализации прецедентов/вариантов использования

Тривиальный пример

1

Имя варианта использования:	Оплатить налог с оборота
ID варианта использования (ID):	1
Краткое описание:	Выплата налога с оборота в Налоговое управление по окончании налогового периода.
Основное действующее лицо:	Время (Таймер).
Второстепенные действующие лица:	Налоговое управление.
Предусловия:	1. Конец налогового периода
Основной поток:	1. Прецедент начинается в конце налогового периода 2. Система определяет сумму Налога с оборота, которую необходимо выплатить Налоговому управлению 3. Система посылает электронный платеж в Налоговое управление.
Постусловия:	1. Налоговое управление получает соответствующую сумму Налога с оборота
Альтернативные потоки:	Нет.

# Примеры написания сценариев реализации прецедентов/вариантов использования

Тривиальный пример

2

Имя варианта использования:	Оплатить налог с оборота
ID варианта использования (ID):	2
Краткое описание:	Выплата налога с оборота в Налоговое управление по окончании налогового периода.
Основное действующее лицо:	Бухгалтер.
Второстепенные действующие лица:	Налоговое управление.
Предусловия:	1. Конец налогового периода
Основной поток:	1. Прецедент начинается, когда Бухгалтер выбирает опцию «оплатить налог с оборота». 2. Система определяет сумму Налога с оборота, которую необходимо выплатить Налоговому управлению 3. Система посылает электронный платеж в Налоговое управление.
Постусловия:	1. Налоговое управление получает соответствующую сумму Налога с оборота.
Альтернативные потоки:	Нет.

# Примеры написания сценариев реализации прецедентов/вариантов использования

## Пример с наличием альтернативных потоков

Имя варианта использования:	Управлять торговой корзиной
ID варианта использования (ID):	3
Краткое описание:	Покупатель меняет количество товаров в корзине.
Основное действующее лицо:	Покупатель
Второстепенные действующие лица:	Нет
Предусловия:	Предусловия: 1. Содержимое корзины для покупок является видимым.
Основной поток:	Основной поток: 1. Прецедент начинается, когда Покупатель выбирает товарную позицию в корзине. 2. Если Покупатель выбирает «удалить позицию». 2.1. Система удаляет позицию из корзины.
Постусловия:	Нет.
Альтернативные потоки:	3. Если Покупатель вводит новое количество. 3.1. Система обновляет количество товаров в корзине.

# Примеры написания сценариев реализации прецедентов/вариантов использования

Пример при наличии предусловия и повторения в потоке

Имя варианта использования:	Найти продукт.
ID варианта использования (ID):	4
Краткое описание:	Система ищет некоторые продукты на основании критерия поиска, заданного Покупателем, и выводит их на экран для Покупателя.
Основное действующее лицо:	Покупатель
Второстепенные действующие лица:	Нет
Предусловия:	Предусловия: 1. Содержимое корзины для покупок является видимым.
Основной поток:	Основной поток: 1. Прецедент начинается, когда Покупатель выбирает опцию «найти продукт». 2. Система запрашивает у Покупателя критерий поиска. 3. Покупатель вводит запрашиваемый критерий. 4. Система ищет продукты, соответствующие критерию Покупателя. 5. Если система находит соответствующие продукты, тогда: 5.1. Для каждого найденного продукта 5.1.1. система выводит на экран миниатюрное представление продукта 5.1.2. система выводит на экран краткое описание продукта 5.1.3. система выводит на экран цену продукта 6. Иначе: 6.1. Система сообщает Покупателю о том, что соответствующие продукты не найдены.
Постусловия:	Нет.
Альтернативные потоки:	Нет.



# Примеры написания сценариев реализации прецедентов/вариантов использования

Пример описания сценария при наличии постусловий и повторения в потоке

Имя варианта использования:	Показать данные о компании
ID варианта использования (ID):	5
Краткое описание:	Система выводит данные о компании для Покупателя.
Основное действующее лицо:	Покупатель
Второстепенные действующие лица:	Нет
Предусловия:	Нет
Основной поток:	Основной поток: 1. Прецедент начинается, когда Покупатель выбирает опцию «показать данные о компании». 2. Система выводит на экран веб-страницу с данными о компании. 3. Пока Покупатель просматривает данные о компании. 3.1. Система воспроизводит некоторую фоновую мелодию. 3.2. Система отображает специальные предложения в баннере.
Постусловия:	1. Система показала данные о компании. 2. Система воспроизвела фоновую мелодию. 3. Система показала специальные предложения.
Альтернативные потоки:	Нет.

# Примеры написания сценариев реализации прецедентов/вариантов использования

Первый пример с наличием альтернативных потоков

Альтернативные потоки могут быть инициированы тремя разными способами:

- вместо основного потока. Такая ситуация часто возникает в вариантах использования типа CRUD (Create, Read, Update, Delete);
- после определенного этапа основного потока;

в любой момент в ходе выполнения основного потока

Имя варианта использования:	Создать новую учетную запись Покупателя.
ID варианта использования (ID):	5
Краткое описание:	Система создает новую учетную запись для Покупателя.
Основное действующее лицо:	Покупатель
Второстепенные действующие лица:	Нет.
Предусловия:	Нет.
Основной поток:	<p>Основной поток:</p> <ol style="list-style-type: none"> <li>1. Прецедент начинается, когда Покупатель выбирает опцию «создать новую учетную запись Покупателя».</li> <li>2. Пока данные Покупателя недействительны               <ol style="list-style-type: none"> <li>2.1. Система просит Покупателя ввести его данные, включая адрес электронной почты, пароль и еще раз пароль для его подтверждения</li> <li>2.2. Система проверяет действительность данных Покупателя</li> </ol> </li> <li>3. Система создает новую учетную запись для Покупателя.</li> </ol>
Постусловия:	Новая учетная запись создана для Покупателя
Альтернативные потоки:	<p>Неверный e-mail адрес.</p> <p>Неверный пароль.</p> <p>Отмена.</p>

# Примеры написания сценариев реализации прецедентов/вариантов использования

## Описание альтернативного потока «Неверный e-mail адрес»

Имя варианта использования:	Создать новую учетную запись Покупателя.
Альтернативный поток:	Неверный e-mail адрес
ID варианта использования (ID):	5.1
Краткое описание:	Система сообщает Покупателю, что он ввел недействительный адрес электронной почты.
Основное действующее лицо:	Покупатель
Второстепенные действующие лица:	Нет
Предусловия:	1. Покупатель ввел недействительный адрес электронной почты.
Альтернативные потоки:	1. Альтернативный поток начинается после шага 2.2 основного потока 2. Система сообщает Покупателю, что он ввел недействительный адрес электронной почты.
Постусловия:	Нет.
Альтернативные потоки:	Нет.

# Примеры написания сценариев реализации прецедентов/вариантов использования

## Описание альтернативного потока «Отмена»

Имя варианта использования:	Создать новую учетную запись Покупателя.
Альтернативный поток:	Отмена.
ID варианта использования (ID):	5.3
Краткое описание:	Покупатель отменяет процесс создания учетной записи
Основное действующее лицо:	Покупатель
Второстепенные действующие лица:	
Предусловия:	Нет.
Альтернативные потоки:	1. Альтернативный поток начинается в любой момент времени. 2. Покупатель отменяет создание учетной записи.
Постусловия:	1. Новая учетная запись не была создана для Покупателя.

На основании детального описания сценариев, можно перейти к созданию эскизов непосредственных интерфейсов взаимодействия с системой: для GUI (графический пользовательский интерфейс) – это макеты пользовательского интерфейса; для CLI (Command Line Interface) и API – спецификация интерфейса.

Проектирование пользовательского интерфейса выполняется для того чтобы заказчик мог более точно представить себе работу и возможности будущей ИС и дать свои замечания и уточнения требований. В зависимости от сложности проекта и уровня подготовленности заказчика результаты этих работ могут быть представлены в различных формах:

- программная реализация, воспроизводящая точный вид экранных окон;
- альбом экранных форм;
- модель навигации экранов в виде диаграмм классов с указанием атрибутов – полей и операций – кнопок.