

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Пермская государственная сельскохозяйственная академия
имени академика Д.Н. Прянишникова»

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ
направление 230700 «Прикладная информатика»

ЛАБОРАТОРНОЕ ЗАНЯТИЕ № 4

Тема: **Модель прецедентов: описание прецедентов**

Учебные вопросы:

1. Задачи и описание.
2. Типы и форматы прецедентов.
3. Задачи и рамки прецедента.

Вопрос 1. Задачи и описания

В контексте UP *модель прецедентов* (Use-Case Model) относится к дисциплине "Требования".

Требования – это весь набор прецедентов, т.е. модель функционирования системы и ее окружения.

Введем некоторые неформальные определения.

У **потребителей и конечных пользователей** есть свои задачи (которые в контексте UP называют **потребностями**), решение которых должна обеспечить компьютерная система.

Прецедент – это набор сценариев использования, в котором каждый экземпляр сценария представляет собой последовательность действий, выполняемых системой для достижения осязаемого для конкретного исполнителя результата.

Прецеденты – это механизм упрощения этапа формулировки требований для всех заинтересованных лиц. По существу это рассказы об использовании системы в процессе решения поставленных задач.

Основная идея состоит в исследовании и формулировке функциональных требований путем написания историй "из жизни системы". Эти истории помогают сформулировать различные задачи и представляют собой сценарии использования системы.¹ **Сила механизма прецедентов состоит в возможности масштабировать уровень сложности и формальности описания в зависимости от реальных потребностей.**

Сценарий (scenario) – это специальная последовательность действий или взаимодействий между исполнителями и системой. Его иногда также называют *экземпляром прецедента* (use case instance). Это один конкретный сценарий использования системы либо один проход прецедента, например, сценарий успешной покупки товаров за наличный расчет, либо сценарий неудачного завершения покупки из-за прерванной транзакции по обработке данных кредитной карточки.

Основное внимание при описании прецедента нужно сконцентрировать на вопросе: "**Как использование системы обеспечивает осязаемый для пользователя результат или решает его задачу?**", а не на обдумывании системных требований в терминах свойств или функций. Прецеденты определяют пожелания или соглашения относительно поведения системы.

Описания прецедентов – это текстовые документы, а не диаграммы. Моделирование прецедентов – это процесс написания текста, а не рисования. Однако для иллюстрации имен прецедентов и исполнителей, а также их взаимоотношений в UML определены обозначения для диаграммы прецедентов.

Вопрос 2. Типы и форматы прецедентов

Прецеденты типа "черный ящик" и системные обязанности

Прецеденты типа "черный ящик" (black-box use cases) – это самый типичный и рекомендуемый тип прецедентов. Они не описывают внутреннюю работу системы, ее компоненты или дизайн. Наоборот, системе вменяются некоторые **обязанности** (responsibilities). Этот метафорический термин широко применяется в объектно-ориентированном проектировании: программные элементы имеют обязанности и взаимодействуют с другими элементами со своими обязанностями.

Определяя обязанности системы через прецеденты типа "черный ящик", можно указать, **что** должна делать система (функциональные требования), не расписывая, **как** это делать (не выполняя проектирование). Позднее, на этапе проектирования, создается решение, удовлетворяющее разработанной спецификации.

Стиль черного ящика	Другой стиль (белый ящик)
Система регистрирует покупку	Система записывает сведения о покупке в базу данных. Или, еще хуже: система генерирует оператор SQL INSERT для данной продажи...

¹ Термин "прецедент" (use case) появился в Швеции и в переводе означал "случай использования" (usage case).

Прецеденты описываются в различных форматах, в зависимости от потребностей, т.е. выделяют несколько степеней формализации описания прецедентов.

- **Сжатый** – аннотация в виде одного абзаца. Обычно она описывает только главный успешный сценарий. Пример такого описания приведен выше для прецедента Обработка продажи (Process Sale).

Сжатый формат описания прецедента **Обработка продажи (process sale)**.

Покупатель подходит к кассе с выбранными товарами. Кассир с помощью POS-системы регистрирует каждый товар. Система отображает информацию о каждом наименовании товара и вычисляет общую сумму. Покупатель вводит требуемую информацию; система ее верифицирует и регистрирует. Система выполняет инвентаризацию. Покупатель получает товарный чек и покидает магазин с покупками.

- **Свободный** – неформальный стиль описания. Описание прецедента занимает несколько абзацев и охватывает различные сценарии. Примером такого описания является рассмотренный выше прецедент Возврат товара.

Свободный формат прецедента **Возврат товара (Handle Returns)**, включающего некоторые альтернативные сценарии.

Основной успешный сценарий.

Покупатель подходит к кассе с товарами, подлежащими возврату. Кассир использует POS-систему для регистрации каждого возвращаемого товара...

Альтернативные сценарии.

- Если в авторизации кредитной карточки отказано, кассир информирует об этом покупателя и предлагает ему другой способ оплаты покупки.
- Если идентификатор товара в системе не обнаружен, система уведомляет об этом кассира и предлагает ему вручную ввести идентификационный код (возможно, штрих-код поврежден и его сложно считать).
- Если у системы возникают сложности при коммуникации с внешней системой вычисления налога.

- **Развернутый** – наиболее подробный стиль описания. При таком подходе детально описываются все шаги и варианты развития сценария, а также предусловия и результаты.

Развернутые описания прецедентов структурированы и содержат большое количество деталей. Их полезно использовать для углубления понимания целей, задач и требований. Такие описания можно обсуждать на семинарах по определению требований на начальной стадии проекта вместе с системным аналитиком, экспертами предметной области и разработчиками.

Для развернутого описания прецедентов существуют различные шаблоны форматирования. Однако чаще всего используется шаблон, приведенный на Web-узле www.use-cases.org. Рассмотрим пример развернутого описания прецедента *Оформление продажи* для POS-системы «ТТ».

Прецедент П 1. Оформление продажи

Основной исполнитель. Кассир.

Заинтересованные лица и их требования:

- **Кассир.** Хочет точно и быстро ввести данные, не допуская ошибок в платеже, поскольку недостача вычитается из его зарплаты.
- **Продавец.** Хочет получить свои комиссионные от продажи
- **Покупатель.** Хочет купить товары и быстро оформить покупку с минимальными усилиями. Хочет получить подтверждение факта покупки для возможного возврата товара.
- **Компания.** Хочет аккуратно записать транзакцию и удовлетворить интересы покупателя. Хочет удостовериться, что служба авторизации платежей зафиксировала все данные о платеже. Заинтересована в обеспечении устойчивости к сбоям; хочет продолжать регистрировать продажи, даже если серверные компоненты (например, служба удаленной проверки кредитоспособности) недоступны. Хочет автоматически обновлять бухгалтерскую документацию и вести складской учет.

- **Государственные налоговые службы.** Хотят получать налог от каждой продажи. Таких служб может быть несколько, в том числе национальная и местная.

Предусловия. Кассир идентифицирован и аутентифицирован.

Результаты (Постусловия). Данные о продаже сохранены. Налоги корректно вычислены, Бухгалтерские и складские данные обновлены. Комиссионные начислены. Чек сгенерирован. Авторизация платежа выполнена.

Основной успешный сценарий (или основной процесс)

1. Покупатель подходит к кассовому аппарату POS-системы с выбранными товарами.
2. Кассир открывает новую продажу.
3. Кассир вводит идентификатор товара.
4. Система записывает наименование товара и выдает его описание, цену и общую "стоимость. Цена вычисляется на основе набора правил.

Кассир повторяет действия, описанные в п.п. 3-4, для каждого наименования товара.

5. Система вычисляет общую стоимость покупки с налогом.
6. Кассир сообщает покупателю общую стоимость и предлагает оплатить покупку.
7. Покупатель оплачивает покупку, система обрабатывает платеж.
8. Система регистрирует продажу и отправляет информацию о ней внешней бухгалтерской системе (для обновления бухгалтерских документов и начисления комиссионных) и системе складского учета (для обновления данных).
9. Система выдает товарный чек.
10. Покупатель покидает магазин с чеком и товарами (если он что-то купил).

Расширения (или альтернативные потоки)

***а.** При каждом выходе системы из строя.

Для ввода системы в строй и корректной обработки платежа нужно обеспечить восстановление всех транзакций и событий с любого шага сценария:

1. Кассир перезапускает систему, регистрируется и предлагает восстановить предыдущее состояние.
 2. Система восстанавливает предыдущее состояние.
- 2а.** Система определяет аномалию, повлекшую сбой.
1. Система уведомляет об ошибке кассира, регистрирует ошибку и переходит в начальное состояние.
 2. Кассир начинает новую продажу.
- 3а.** Неправильный идентификатор.
1. Система уведомляет об ошибке и отменяет ввод данного наименования товара.
- 3б.** В рамках одной категории существует несколько различных наименований товара, и идентифицировать конкретное наименование не нужно (например, 5 пакетов леденцов).
1. Кассир может ввести идентификатор категории товара и количество единиц.
- 3в.** Покупатель просит кассира отменить покупку одного из товаров.
1. Кассир вводит идентификатор товара для удаления из продажи.
 2. Система отображает обновленную промежуточную стоимость.
- 3г.** Покупатель просит кассира отменить продажу.
1. Кассир отменяет продажу.
- 3д.** Кассир приостанавливает продажу.
1. Система записывает сведения о продаже таким образом, чтобы они были доступны с любого терминала POS-системы.
- 4.** Сгенерированная системой цена товара не устраивает покупателя (например, у него есть дисконтная карта и он рассчитывает на более низкую цену товара).
1. Кассир вводит команду об изменении цены.
 2. Система вычисляет новую цену.
- 5а.** Система выявляет сбой при коммуникации с внешней службой вычисления налога.
1. Система перезапускает службу с данного узла POS-системы и продолжает работу.
 - 1а. Система определяет, что служба не перезапускается.
 1. Система сигнализирует об ошибке.
 2. Кассир может вручную вычислить и ввести сумму налога либо отменить продажу.
- 5б.** Покупатель сообщает о положенной ему скидке (например, для сотрудников данного

предприятия или постоянных покупателей).

1. Кассир отправляет запрос на скидку.
2. Кассир вводит идентификационные данные покупателя.
3. Система представляет сумму скидки, вычисленную на основе дисконтных правил.

5в. Покупатель сообщает об открытом в данном магазине кредите на фиксированную сумму и просит оформить продажу с его помощью.

1. Кассир отправляет запрос на оформление платежа с использованием открытого кредита,
2. Кассир вводит идентификационную информацию о покупателе.
3. Система снижает стоимость покупки (вплоть до 0) и уменьшает оставшуюся сумму кредита.

6. Покупатель сообщает, что хочет оплатить покупку наличными, но у него недостаточно денег.

1 а. Покупатель использует альтернативный способ платежа.

16. Покупатель просит кассира отменить продажу. Кассир отменяет продажу в системе.

7а. Оплата наличными.

1. Кассир вводит предложенную покупателем сумму.
2. Система вычисляет положенную сдачу и открывает кассу с наличностью.
3. Кассир складывает поученные деньги и выдает сдачу покупателю.
4. Система регистрирует платеж наличными.

7б. Оплата по кредитной карточке.

1. Покупатель вводит информацию о своей кредитной карточке.
 2. Система отправляет запрос на авторизацию платежа внешней системе службы авторизации платежей и запрашивает подтверждение платежа.
- 2а. Система определяет сбой при взаимодействии с внешней системой.
1. Система сигнализирует об ошибке кассиру.
 2. Кассир просит покупателя изменить тип платежа.
3. Система получает информацию о подтверждении платежа и сообщает об этом кассиру.
- 3а. Система получает информацию об отказе в выполнении платежа,
1. Система сообщает об отказе кассиру.
 2. Кассир просит покупателя изменить тип платежа.
4. Система регистрирует платеж по кредитной карточке, включая информацию о подтверждении платежа.
 5. Система предоставляет механизм ввода подписи для платежа по кредитной карточке.
 6. Кассир просит покупателя подписать чек на оплату по кредитной карточке, Покупатель вводит подпись.

7в. Оплата чеком.

7г. Оплата по дебитной карточке.

7д. Покупатель предоставляет купоны.

1. До обработки платежа кассир вводит информацию о каждом купоне, в соответствии с которой система снижает стоимость покупки. Система регистрирует предъявленный купон для нужд бухгалтерии.
- 1а. Купон действует не для всех выбранных товаров,
1. Система сигнализирует об ошибке кассиру.

9а. Генерация чека.

1. Система выводит формы и чеки для каждого товара.

9б. Покупатель просит выдать ему подарочный чек (без указания цены).

1. Кассир вводит запрос на подарочный чек, и система выдает его.

Специальные требования

- Сенсорный экран с интерфейсом пользователя для большого плоского монитора. Текст должен быть виден с расстояния один метр.
- Отклик службы авторизации в 90% случаев приходит в течение 30 секунд.
- Каким-то образом нужно обеспечить робастное восстановление информации в случае сбоя при доступе к удаленным службам, таким как система складского учета.
- Возможность локализации (представления на различных языках) отображаемого текста.
- Возможность добавления новых бизнес-правил на шагах 3 и 7 в процессе функционирования системы.

Список технологий и типов данных

- 3а. Идентификатор товара считывается со штрих-кода (при наличии последнего) лазерным сканером или вводится с клавиатуры.
- 3б. Идентификатор товара может определяться по схемам кодирования UPC, EAN, JAN или SKU.
- 7а. Информация об открытом кредите вводится с помощью считывающего устройства или с клавиатуры.
- 7б. Подпись при оплате чеком ставится на бумажном документе. Однако ожидается, что в течение двух лет большинство покупателей будут требовать цифровые устройства считывания подписи.

Частота использования: почти постоянно.

Открытые вопросы

- Изучить законодательство по налогообложению.
- Исследовать вопрос восстановления удаленных служб.
- Какая настройка потребуется для различных типов магазинов.
- Должен ли кассир снимать кассу при выходе из системы.
- Может ли пользователь сам использовать устройство считывания данных с карточки или это должен делать кассир.

Этот прецедент скорее является иллюстративным, чем исчерпывающим (хотя и основывается на реальных требованиях к POS-системе). Тем не менее, он описан достаточно подробно и позволяет составить реалистичное представление о развернутом формате описания прецедентов и их сложности.

Этот пример может служить **моделью** для многих других прецедентов.

Пояснения к примеру

Вводные элементы

В описание прецедента можно добавлять различные вводные элементы. Те из них, которые важны для изложения последующего сценария, следует помещать в начало описания. Второстепенный материал можно располагать в конце сценария. Например, в описание можно добавить следующий вводный элемент:

Главный исполнитель. Основной исполнитель, вызывающий системные службы для достижения цели.

Заинтересованные лица и их потребности

Этот список играет важную роль. С его помощью можно понять, что должна делать система. Приведем цитату: "Система реализует соглашение между заинтересованными лицами. Поведение системы описывается с помощью прецедентов... Прецедент, как соглашение о поведении, включает все возможные аспекты поведения, связанные с удовлетворением запросов заинтересованных лиц".

Эта цитата дает ответ на вопрос, что нужно описывать в прецеденте. Там нужно описывать все, что служит удовлетворению запросов заинтересованных лиц. Кроме того, начиная описание прецедента с перечня заинтересованных лиц и их интересов, можно более точно определить функции системы.

Предусловия и постусловия

Предусловия (*preconditions*) – это перечень предпосылок, которые **всегда должны** выполняться до начала сценария прецедента. Предусловия **не** проверяются при реализации прецедента. То есть это условия, которые считаются истинными. Обычно в качестве предусловия выступает успешный результат выполнения другого сценария, например, загрузки или авторизации. В качестве предусловий перечисляются не все возможные истинные условия. Например, никто не упоминает в предусловиях наличие напряжения в электросети. Предусловия – это те предпосылки, на выполнение которых разработчик прецедента хочет обратить особое внимание.

Результаты или постусловия (*postconditions*) – описывают, какие условия обязательно должны выполняться в случае успешного завершения сценария. Эти результаты должны удовлетворять интересам всех заинтересованных лиц.

Основной успешный сценарий (или основной процесс)

В нем описывается типичная последовательность действий, приводящая к успешному

завершению сценария и удовлетворяющая потребности всех заинтересованных лиц.

Чаще всего в этом разделе *нет* никаких условий или ветвей. И хотя вводить какие-либо условия не запрещается, их обычно выносят в раздел расширений.

В разделе основного сценария описываются три вида действий:

1. Взаимодействие между исполнителями.²
2. Верификация (обычно со стороны системы).
3. Изменение состояния системы (например, запись или модификация некоторых сущностей).

Первый шаг прецедента не всегда подпадает под эту классификацию. Он служит **триггером** события начала сценария.

Имена исполнителей принято начинать с заглавной буквы для облегчения их идентификации. Повторяющиеся действия выделяются курсивом.

Расширения (или альтернативные потоки)

В этом разделе указываются все остальные сценарии или ветви, приводящие к успешному или неудачному завершению прецедента.

При описании прецедента основной успешный сценарий и его расширения должны охватывать *почти все интересы заинтересованных лиц*. Некоторые интересы лучше выразить в виде нефункциональных требований в дополнительной спецификации, а не в описании прецедента. **Расширения** – это ответвления от основного сценария.

Специальные требования

В этот раздел заносятся нефункциональные требования, атрибуты качества или ограничения, связанные с данным прецедентом. Сюда относятся характеристики производительности, надежности, удобства использования и конструктивные ограничения (например, на устройства ввода-вывода).

Запись этих условий при описании прецедента – классический совет разработчиков унифицированного процесса. Однако на практике многие специалисты помещают эти требования в едином общем документе, например в дополнительной спецификации. Тогда их удобнее читать и осмысливать, поскольку обычно они рассматриваются в общем контексте в процессе анализа архитектуры.

Список технологий и типов данных

Зачастую при реализации проекта важно не *что* сделать, а *как*. Перечень используемых технологий тоже приводится в описании прецедента. Типичным примером такой ситуации являются технические ограничения, выдвигаемые заинтересованными лицами для технологий ввода и вывода.

Например, заказчик может потребовать, чтобы POS-система поддерживала ввод данных кредитной карточки с клавиатуры и с помощью считывающего устройства.

В этом разделе приводятся лишь примеры проектных решений и ограничений, появляющихся на ранней стадии реализации проекта. В целом, такой конкретизации следует избегать, однако иногда она бывает неизбежна, особенно в отношении технологий ввода/вывода. Нужно также указать возможные варианты типов данных, например, различные кодировки идентификаторов товаров.

Вопрос 3. Задачи и рамки прецедента

Как выделить прецедент?

Зачастую определить правильный (а точнее, **полезный**) прецедент очень сложно. Каждую задачу можно рассматривать на разных уровнях детализации, начиная от конкретных простых действий и заканчивая деятельностью на уровне предприятия.

На каком же уровне детализации следует формулировать прецеденты?

В процессе анализа требований к компьютерному приложению следует сосредоточить внимание на уровне *элементарных бизнес-процессов* (EBP – Elementary Business Processes).

Термин EBP заимствован из области исследования бизнес-процессов³ и определяется

² Заметим, что описываемая система сама является исполнителем, если рассматривается аспект ее взаимодействия с другими системами.

³ Термин EBP аналогичен термину *пользовательская задача* (user task), применяемому в области изучения

следующим образом.

"Элементарный бизнес-процесс – это задача, выполняемая одним человеком в одном месте в одно время в ответ на некоторое бизнес-событие, добавляющая измеряемое бизнес-значение и переводящая данные в некоторое устойчивое состояние, например, подтверждение платежа по кредитной карточке или распоряжение брокеру при изменении цен" (источник неизвестен).

Прецедент – это не один маленький шаг, такой, например, как удаление товара или печать документа. Основной сценарий прецедента обычно включает пять-десять шагов. Описываемый сценарий выполняется не в течение нескольких дней или сеансов, как, например, переговоры с поставщиками. Это задача, выполняемая в течение одного сеанса. Реализация прецедента может длиться от нескольких минут до нескольких дней. Как и при определении унифицированного процесса, основное внимание уделяется получению ощутимого (измеримого) результата и переходу системы и данных в **устойчивое состояние**.

Обоснованные отклонения от элементарного бизнес-процесса

Несмотря на то, что основные прецеденты приложения должны соответствовать элементарным бизнес-процессам, зачастую полезно создавать отдельные прецеденты более низкого уровня, представляющие подзадачи или подпоследовательности действий в рамках основного прецедента. То есть могут существовать прецеденты, не соответствующие элементарным бизнес-процессам, а функционирующие на более низком уровне. Общая рекомендация позволяет выявить только основные прецеденты в процессе анализа требований к приложению и сконцентрировать внимание на их написании.

Например, подзадача или расширение "Оплата по кредитной карточке" может быть представлена в нескольких основных прецедентах. Поэтому ее желательно выделить в отдельный прецедент (не удовлетворяющий условиям элементарного бизнес-процесса) и связать с несколькими основными прецедентами, чтобы избежать дублирования информации.

Прецеденты и задачи

Исполнители имеют свои задачи (или потребности), для решения которых они используют систему. Поэтому прецеденты уровня EBF еще называют прецедентами уровня *задач пользователя* (user goal). Это делается для того, чтобы обратить внимание на реализацию потребностей пользователей системы или основного исполнителя.

Отсюда следует алгоритм выделения прецедентов:

1. Выделить задачи (цели) пользователей.
2. Определить для каждой из них отдельный прецедент.

При таком подходе несколько смещаются акценты аналитиков. Вместо вопроса "Каковы прецеденты для данной системы?" возникает вопрос: "Каковы задачи исполнителей?". Чтобы отобразить эту взаимосвязь, **имя прецедента должно соответствовать названию задачи**. Например, задаче электронного оформления продажи должен соответствовать прецедент Оформление продажи.

Такая симметрия **позволяет оценить адекватность** уровня выделения прецедентов и задач пользователя в соответствии с правилом выделения элементарных бизнес-процессов.

Таким образом, ключевая идея состоит в том, чтобы для выделения прецедентов исследовать задачи исполнителей.

Пример: использование рекомендаций в соответствии с EBF

Допустим, вы являетесь системным аналитиком и отвечаете за формулировку требований к системе ТТ. Для этого вам нужно исследовать задачи пользователей. На семинаре по формулировке требований может состояться такой диалог.

Системный аналитик: "Каковы ваши задачи в контексте использования POS-системы?"

Кассир: "Во-первых, быстро зарегистрироваться. Во-вторых, оформлять продажи".

Системный аналитик: "Какая задача более высокого уровня приводит, на ваш взгляд, к необходимости выделения отдельной задачи регистрации?"

Кассир: "Мне необходимо "представиться" системе, а она должна проверить, имею ли я право ею пользоваться".

Системный аналитик: "Какова еще более глобальная задача?"

Кассир: "Предотвратить утечку или повреждение данных".

удобства использования. Однако в той предметной области его значение является менее жестким.

Обратите внимание на стратегию аналитика выстроить иерархию целей и выявить таким образом нужный уровень для элементарного бизнес-процесса. Это позволяет лучше понять мотивацию действий исполнителей и их задачи.

Предотвращение утечки данных – это цель более высокого уровня, чем задача пользователя. Поэтому пока мы ее рассматривать не будем, хотя она является чрезвычайно важной для данной системы. (Самым кардинальным решением этой задачи является полный отказ от POS-системы и услуг кассира.)

Цель следующего уровня иерархии ("представиться" системе и выполнить аутентификацию) несколько ближе к задачам пользователя. Но относится ли она к уровню элементарных бизнес-процессов? Ее решение не добавляет осязаемого результата или измеримого бизнес-значения. Если на вопрос о том, чем кассир занимался сегодня на работе, прозвучит ответ: "Я 20 раз зарегистрировался!", то вряд ли такой ответ устроит начальника. Значит, это второстепенная задача, которая служит достижению важной цели, но не относится к уровню ЕВР. Уровню ЕВР точнее всего соответствует задача оформления продажи.

В качестве другого примера можно рассмотреть процесс регистрации выручки, когда кассир задвигает ящик с наличностью и закрывает его в системе, регистрируется и вводит в систему сумму выручки. Регистрация выручки – это прецедент уровня ЕВР (или уровня задач пользователя), а регистрация в системе – это один из его шагов, выполняющий вспомогательную задачу, а не отдельный прецедент.

Вспомогательные задачи и прецеденты

Несмотря на то, что задача "представиться" системе и выполнить аутентификацию (или задача регистрации) не была отнесена к уровню задач пользователя, она все же является целью, но на более низком уровне.

Такие задачи называют *вспомогательными* (subfunctional goal), поскольку они призваны обеспечивать выполнение задач пользователя. Для вспомогательных задач отдельные прецеденты создаются очень редко, хотя специалисты по написанию прецедентов зачастую рекомендуют улучшать (обычно упрощать) набор прецедентов.

Для вспомогательных задач писать прецеденты не запрещается, однако это не всегда нужно, поскольку при этом усложняется модель прецедентов. Количество вспомогательных задач для системы может исчисляться сотнями, но стоит ли создавать так много прецедентов?

Важно понимать, что с увеличением числа прецедентов возрастает сложность задачи формулировки и управления требованиями, а значит, увеличивается также время решения этой задачи.

Основным мотивом написания прецедента для вспомогательной задачи должна служить повторяемость этой задачи или ее важное значение как предусловия для множества других прецедентов. Этому принципу удовлетворяет задача авторизации, которая обеспечивает предусловие для большинства, если не всех остальных прецедентов уровня задач пользователя.

Таким образом, вполне логично создать прецедент Аутентификация пользователя.

Составные задачи и прецеденты

Задачи могут принадлежать к различным уровням сложности и являться составными, начиная от уровня предприятия ("быть прибыльным"), до задач среднего уровня ("регистрация торговых операций") и вспомогательных задач в рамках приложения ("проверка правильности ввода").

Аналогично, и прецеденты могут относиться к различным уровням сложности и состоять из прецедентов более низкого уровня.

Наличие нескольких уровней сложности задач и прецедентов может ввести в заблуждение при определении соответствующего уровня для основных прецедентов. Для отсеивания слишком детальной информации следует использовать принцип нахождения элементарных бизнес-процессов.

Задание на самостоятельную работу (для выбранной темы индивидуального проекта):

1. Описать все прецеденты в сжатом формате.

2. Выбрать один прецедент (на примере которого будет выполнено объектно-ориентированное проектирование) и составить для него развернутое описание.