



UML для IT бизнес аналитиков: Бизнес Анализ. Этап начала проекта (Initiation phase)

Luxoft

Обо мне



Alexander Belin

abelin@luxoft.com

Senior Business Analyst

План семинара



- **Вступление**
 - Зачем ВА объектно ориентированный подход?
 - Для кого этот курс?
 - Что вы будете уметь прослушав данный курс
- **Кто такой Бизнес Аналитик?**
- **Проекция Бизнес Анализа на Объектно Ориентированный подход**
 - Понимание, как ОО подход влияет на работу Бизнес Аналитика в IT проекте
 - Краткий обзор основных ОО понятий
- **Шаги Бизнес Объектно Ориентированного Моделирования (Business Object Oriented Modeling (B.O.O.M.))**
- **Первые шаги процесса Бизнес Анализа. Этап начала проекта (Initiation phase)**
- **Определение бизнес Workflow с использованием Диаграммы Деятельности (Activity Diagram)**

Вступление

- Зачем ВА объектно ориентированный подход?
- Для кого этот курс?
- Что вы будете уметь прослушав данный курс?
- Практический пример

Зачем БА объектно ориентированный подход



Благодаря в значительной степени развитию клиент-серверных приложений и Интернета, объектно-ориентированные (ОО) языки, таких как C++ и Java приняты всем разработческим миром. Это уже изменило подходы, с помощью которых технические члены ИТ-команды (системные аналитики, программисты, и так далее) работают. Но бизнес - аналитики, люди, которые доносят требования разработчикам - прежнему работают, по большей части, как будто ОО не существует. В результате написание документации бизнес аналитика, документации, которая первая должна была быть переведена в термины Объектно Ориентированного подхода и стандартов разработчиков - неэффективный и подверженный ошибкам этап. Поэтому очевидна необходимость в пошаговой программе обучения, которая поможет БА работать более эффективно в проектах используя Unified Modeling Language (UML), наиболее распространенный стандарт для моделирования бизнес-и ИТ-систем в проектах ОО (хотя и не ограничивается ОО проектами, поскольку стандарт также имеет не-ОО элементы моделирования)

Для кого этот курс?



- Эта курс предназначен для тех, кто исполняет роль аналитика IT бизнес аналитика (IT-BA) – объединяющую роль команды по разработке программного обеспечения, роль, в рамках которой сотрудник ответственной за разработку и поддержку бизнес и пользовательских требований к программному обеспечению. Если вы новичок в IT Бизнес Анализе, или вы являетесь опытным БА кто только начинает работать со стандартом UML, то этот курс для вас. Это не означает, однако, что курс рассчитан на слушателей, которые непременно имеют должность с названием "IT-бизнес-аналитик". Другие названия, которые могут перекрываться с IT ролью, включая роль «бизнес-аналитик», «аналитик бизнес-процессов», инженер требований, аналитик бизнес - систем, системный аналитик, специалист по анализу данных, функциональный архитектор, product manager и аналитик пользовательского опыта. Если вы занимаете любую из этих позиций и в ваши обязанности входит сбор требований и моделирование бизнес области в рамках IT проекта, то этот курс для вас.

Что вы будете уметь прослушав данный курс?



- Создавать документ бизнес требований (business requirements document BRD), который соответствует стандарту UML 2.2 и который включает в себя, **use cases, диаграммы классов и другие методы объектно-ориентированного анализа (OOA)**
- **следовать пошаговому процессу OOA проведения интервью, выполнения исследований, и документирование требований**
- **Использовать следующие артефакты: Диаграммы бизнес Use Cases (Business use-case diagrams)**
 - **Диаграммы системных Use Cases и шаблонов Use Cases (System use-case diagrams and use-case templates)**
 - **Диаграммы пакетов (Package diagrams)**
 - **Диаграммы классов (Class diagrams)**
 - **Диаграммы внутренней структуры (Composite structure diagrams)**
 - **Диаграммы объектов (Object diagrams)**
 - **Диаграммы деятельности с разбиением и без (Activity diagrams with and without partitions)**
 - **Диаграмма автомата (State-machine diagrams)**

Все это лишь игра!

- **В процессе работы вам приходится тратить много времени, анализируя сложные взаимосвязи в большой системе. В этой ситуации легко начать сомневаться в правильности выбранных решений. Ну, вот хорошая новость: не существует «единственного» или правильного решения, поскольку существует больше одного способа создания модели реального мира. Так что лучший подход - это просто играть с идеей и посмотреть, куда вас это приведет. Как определить, насколько хорошо ваше решение? Вы можете быть уверены, что вы идете в правильном направлении, если ваша модель имеет следующие качества:**
 - Элегантность: простое решение сложной проблемы**
 - Адаптивна: может быть легко изменена, чтобы отразить изменения в требованиях**
 - Неизбыточна: не повторяет сама себя, каждый факт описан только в одном месте**



Кто такой Бизнес Аналитик?

Содержание раздела



В рамках данного раздела мы разберем следующее:

- **Роль IT Бизнес Аналитика в течение всего жизненного цикла проекта**
- **Что означают следующие термины: бизнес модель, модель процесса, модель данных, и объектная модель**

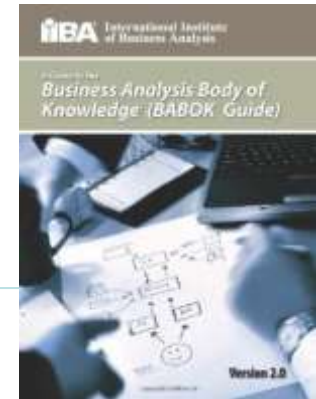
- **Есть два типа Бизнес Аналитиков:**
 - IT Бизнес Аналитик – аналитик, работающий в области информационных технологий (IT BA). Данный аналитик работает в рамках IT – проектов по продаже, покупке, или модификации программного обеспечения. Этот человек поддерживает связь с бизнес и техническими заказчиками (**stakeholders**) и отвечает за сбор требований, которые формулируются на основе информации, исходящей от бизнеса;
 - не- IT Бизнес Аналитик - это тот, кто работает в рамках какого-то бизнеса. Этот человек вовлечен в процесс улучшения бизнеса, сокращения расходов, и так далее.
 - **«Бизнес консультанты» - аналитики, занимающиеся анализом и оптимизацией бизнеса компаний;**
 - **«Бизнес наблюдатели» - аналитики, следящие за тенденциями развития рынков, отраслей бизнеса. Аналитики, способные дать прогноз, предсказать направление дальнейшего развития.**

- Guide to the Business Analysis Body of Knowledge®
(BABOK® Guide) Version 2.0
<http://www.theiiba.org/am/>
- Software Requirements,
Second Edition by Karl E. Wiegers
<http://www.karlwiegers.com/>

Guide to the Business Analysis Body of Knowledge® (BABOK® Guide) Version 2.0



“A business analyst is any person who performs business analysis activities, no matter what their job title or organizational role may be. Business analysis practitioners include not only people with the job title of business analyst, but may also include business systems analysts, systems analysts, requirements engineers, process analysts, product managers, product owners, enterprise analysts, business architects, management consultants, or any other person who performs the tasks described in the BABOK® Guide, including those who also perform related disciplines such as project management, software development, quality assurance, and interaction design.”

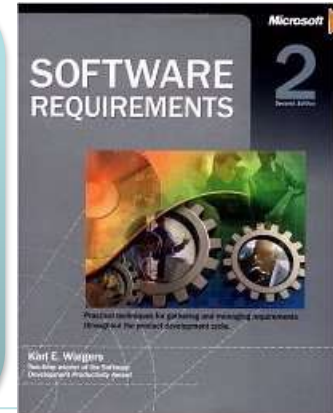


- Бизнес аналитик есть человек, который выполняет действия по бизнес анализу, не зависимо от названия занимаемой должности или роли в компании. Специалисты в Бизнес Анализе включают в себя не только людей занимающие позиции Бизнес Аналитика, но могут включать также людей с позициями Бизнес Системный Аналитик, Системный Аналитик, Инженер Требований, Процессный Аналитик, product managers, product owners, enterprise analysts, business architects, management consultants, или каких-либо других специалистов, которые выполняют задачи, описанные в *BABOK® Guide*, включая тех, кто работает в рамках других дисциплин, таких как *project management, software development, quality assurance, и interaction design*

Software Requirements, Second Edition by Karl E. Wiegers



Explicitly or implicitly, someone performs the role of requirements analyst on every software project. Corporate IS organizations identify specialists called *business analysts* to perform this function. Synonyms for *requirements analyst* include *systems analyst*, *requirements engineer*, *requirements manager*, and simply *analyst*. In a product-development organization, the job is often the product manager's or marketing staff's responsibility. The analyst is a translator of others' perspectives into a requirements specification and a reflector of information back to other stakeholders. The analyst helps stakeholders find the difference between what they say they want and what they really need.



- Явно или неявно в каждом проекте кто-то исполняет роль Аналитика Требований. Организации выделяют специалистов, которых называют «Бизнес Аналитик» для выполнения этих функций. Список синонимов для термина «Аналитик Требований» включает «Системный Аналитик», «Инженер Требований», «*requirements manager*» и просто Аналитик. В организациях, производящих продукт, данные обязанности зачастую называются “product manager's” или “marketing staff's” обязанности. Аналитик – это переводчик видения заказчика в описания требований (requirements specification) и человек, который возвращает информацию обратно заказчику. Аналитик помогает заказчику понять разницу между тем, что он (заказчик) говорит и тем, что он реально хочет иметь.

Роль IT Бизнес Аналитика

Guide to the Business Analysis Body of Knowledge® (BABOK® Guide) Version 2.0



Business analysts must analyze and synthesize information provided by a large number of people who interact with the business, such as customers, staff, IT professionals, and executives. The business analyst is responsible for eliciting the actual needs of stakeholders, not simply their expressed desires. In many cases, the business analyst will also work to facilitate communication between organizational units. In particular, business analysts often play a central role in aligning the needs of business units with the capabilities delivered by information technology, and may serve as a “translator” between those groups.

- Бизнес Аналитик должен анализировать и синтезировать информацию, предоставляемую большим количеством людей, которые взаимодействуют с бизнесом, такие как «заказчики», персонал компаний, IT специалисты и исполнители. Бизнес Аналитик ответственен за выявление фактических требований заказчика, а не только тех пожеланий, которые он (заказчик) высказывает. Во многих случаях, Бизнес Аналитик содействует общению, обмену информации между различными организационными единицами (планирование, разработка, тестирование). В особенности, Бизнес Аналитик зачастую играет центральную роль в увязывании потребностей бизнес подразделений с возможностями, предоставляемыми в области информационных технологий, и может работать как своеобразное «передаточное звено» между этими группами

Роль IT Бизнес Аналитика

Software Requirements,
Second Edition by Karl E. Wieggers



The requirements analyst is the individual who has the primary responsibility to gather, analyze, document, and validate the needs of the project stakeholders. The analyst serves as the principal conduit through which requirements flow between the customer community and the software development team, as shown in Figure (see next slide). Many other communication pathways also are used, so the analyst isn't solely responsible for information exchange on the project. The analyst plays a central role in collecting and disseminating product information, whereas the project manager takes the lead in communicating project information

- Аналитик Требований есть человек, первичной обязанностью которого является сбор, анализ, документирование, проверка и согласование нужд заказчиков проекта. Аналитик работает как основное средство коммуникации, по которому требования перемещаются между группой заказчиков и командой программистов, как это показано на рисунке. Также используется множество других каналов связи, таким образом Аналитик не отвечает единолично за информационный обмен в рамках проекта. Аналитик играет центральную роль в сборе и распространении информации, касающейся продукта, в то время, как руководитель проекта руководит процессом обмена информации в рамках проекта

Итог:

В этом разделе мы рассмотрели:

- **Роль IT БА – это представить пожеланий стороны бизнеса команде разработчиков.**
- **Основные обязанности IT БА – выявить и донести требования до разработчиков.**



Объектно Ориентированный подход с точки зрения Бизнес Аналитика

Содержание раздела

В данном разделе мы рассмотрим:

- как Объектно Ориентированный подход (ОО) влияет на роль БА IT проектов;
- Рассмотрим такие основные концепции ОО как:
 - **Объекты**
 - **Операции и атрибуты**
 - **Инкапсуляция**
 - **Классы**
 - **Классы сущностей**
 - **Отношения:**
 - Обобщение
 - Ассоциации
 - Агрегирование
 - Композитный агрегации
 - Полиморфизм
 - Системные Use Cases
 - Бизнес Use Cases
 - Unified Modeling Language (UML)

Как Объектно Ориентированный подход влияет на роль БА IT проектов



Что такое OO?

OO это акроним понятия "объектно-ориентированный подход". Аналитик, использующий OO видит систему как набор объектов, которые взаимодействуют, посылая запросы друг другу (другой вариант этой фразы: объекты посылают сообщения друг другу) OO – это законченная, концептуальная система взглядов, которая охватывает весь жизненный цикл IT проектов.

- OO влияет на то, как БА анализирует и моделирует требования**
- OO влияет на то, как инженер-программист (технический аналитик систем) разрабатывает спецификацию дизайна системы**
- OO влияет на то, как структурирован собственно программный код. Объектно-ориентированные языки программирования, такие как C++ и .NET поддерживают OO концепции и структуры**
- Все они основаны на одних и тех же понятиях, который мы рассмотрим в этой главе.**

ОО начинается с наблюдения, что мы воспринимаем мир не просто в виде неопределенных ощущений. Скорее, мы различаем отдельные объекты, и у нас есть их внутренние образы, которые вы можете увидеть глазами нашего разума. Взятые вместе, эти внутренние объекты моделируют элементы реального мира.

Взгляд БА

Мы начинаем анализировать бизнес, прося опрашиваемых описать бизнес объекты, которые существуют в рамках данного бизнеса. Бизнес объекты - это сущности или понятия, которые бизнес (и IT система должна автоматизировать это) должен контролировать или которые участвуют в бизнес-процессах. Примерами таких объектов могут включать счета, представителей службы поддержки клиентов, или, собственно, вызов.

Атрибуты и операции



- Мы продолжаем исследовать объекты. Что мы можем о них еще сказать. Это то, что внутренне представление объектов должно содержать в себе информацию о его свойствах.
- Например, цвет объекта «рубашка» синий, его размер «L». В ОО, цвет и размер, называются атрибутами, «синий» и «L» - это значения атрибутов. Каждый объект имеет свой набор значений атрибутов.
Что мы еще знаем об объектах? Объекты выполняют какие-то функции. Например, когда мы в первый раз увидели карандаш, нам понадобилось некоторое время, чтобы узнать, что он может быть использована для рисования каракулей на стенах. К сожалению, для ваших родителей, в следующий раз, когда вы видели, что карандаш, вы точно знали, что с ним делать. Почему? Потому что ты вспомнил, что каракули – это было то, что вы могли бы сделать с этим объектом. В ОО, рисование каракулей называется «операция».
Чтобы подытожить то, что мы успели выяснить: Две вещи мы помним о объектах
 - значения их атрибутов
 - операции, которые мы можем сделать с ними
 -
 -

Взгляд БА

- Следующий шаг в анализе бизнес систем – это выяснить, какие атрибуты и бизнес операции присущи каждому объекту. Например, два атрибута, которые присущи к объекту «счет» - это баланс и дата последней проводки, две операции, которые относятся к объекту «внести средства» и «снять средства». Операций объекта, как правило, изменяют или запрашивают значения его атрибутов. Например, операция «снять средства» изменяет значение атрибут

Операции и методы



Двигаемся далее, мы не только знаем, что вы можем сделать с объектом, мы знаем, как мы это делаем. Например, мы не только знаем, что мы можем позвонить с мобильного телефона, но мы также знаем, как сделать это. Для того, чтобы позвонить мы должны следовать определенной процедуры: во-первых мы вводим номер телефона и затем нажимаем кнопку «Звонить». С точки зрения ОО, «сделать звонок» - это операция; процедура, используемая для его реализации называется методом.

Взгляд БА

Далее, мы рассматриваем каждую операцию и спрашиваем заинтересованных лиц какова процедура (последовательность действий) реализации этой операции. Мы документируем описанную процедуру как метод. Например, вы просите заинтересованных лиц рассказать, какова процедура снятия средств со счета. Они говорят, что в первую очередь они проверяют, не заблокирован ли счет и достаточно ли средств на счете. Если все в порядке, они уменьшают баланс и создают запись аудита данной транзакции. Мы записываем эту процедуру, как метод, используемый для проведения операции

Каждый день вы используете объекты, не зная, как они работают и какова их внутренняя структура. Это очень важный аспект того, как мы человеческие объекты взаимодействуем с другими объектами. Это хранит нас от необходимости знать слишком много. Это также означает, что мы можем легко использовать другой объект с совершенно другой внутренней структурой в случае, если внешне он ведет себя так же.

Это ОО принцип инкапсуляции: Только операции объекта видны другим объектам. Атрибуты и методы остаются скрытыми от глаз.

Взгляд БА

Когда вы описываете метод объекта, не упоминайте атрибуты другого объекта или не описывайте предположения о том, как другой объект осуществляет свои операции. Преимущество такого подхода в том, что если методы или атрибуты, связанные с бизнес-объект будут когда-либо изменены, вам придется внести исправления в только одной части модели. Остальные объекты это изменение не затронет

ОО Концепция: Классы



Определение:

"Класс описывает набор объектов, которые объединяют одинаковые описания свойств, ограничений и семантики." [2] (UML)

Что это означает:

Класс – это категория. Все объекты, которые принадлежат этой категории имеют одинаковые атрибуты и операции (но значения атрибутов может меняться от объекта к объекту).

[2] *UML Superstructure Specification*, v2.2, OMG, page 49, 2009

ОО Концепция: Классы #2



- Это означает, значение атрибутов может меняться от объекта к объекту в рамках класса, например, цвет одной ручки может быть синим, а цвет другой зеленым. Все объекты класса имеют одинаковые операции и методы. Другими словами, все объекты класса могут делать одно и то же, и одинаковым способом. Вы узнаете позже, что отношения (например, отношения между клиентом и счет - фактурой) также может быть указано на уровне класса.
- **Взгляд БА**
Несмотря на название «Объектно Ориентированный Анализ», вы будете тратить большую часть своего времени на определение классов, а не объектов. Классов, которыми вы будете интересоваться, это классы, которые связаны с бизнесом. Их называют классы-сущности (entity classes). Например, в банковской системе, нужно определить характеристики класса - сущности «Счет» и класса - сущности «Клиент» организации.

ОО Концепция: Отношения



Мы часто определяем один класс в терминах другого класса. Например, «автомобиль» является видом «средства передвижения». «Автомобиль» и «средство передвижения» являются классами. Фраза " является видом" описывает отношения между двумя классами.

UML определяет множество типов отношений, которые полезны для ВА: обобщение, ассоциации, агрегация композитная агрегация

ОО Концепция: Обобщение



Определение:

"Обобщение - это отношение между более обобщенным классификатором и более конкретным классификатором. Каждый экземпляр конкретного классификатора также косвенный экземпляр общего классификатора. Таким образом, конкретный классификатор наследует классификатора особенности более общего классификатора. "[3] (UML)

Что это означает:

Когда объект принадлежит специализированный класс (например, iPhone 3G), это автоматически означает, что он принадлежит обобщению этого класса (например, мобильный телефон). Любой атрибут или операция, которая применима к обобщенному классу также относится к специализированным классом. Кроме того, любые отношения, указанные для обобщенного класса также распространяется на специализированные классы. (Например, отношение "Клиент владеет [одним или несколькими] счетами", определено для класса «Счет», распространяется также на «Расчетный счет», «Сберегательный счет», и все другие типы счетов [специализированные классы счета].)

Взгляд БА

Мы рассматриваем классы бизнес объектов, которые являются дочерними по отношению к более общему типу. Например, «Расчетный счет» и «Сберегательный счет» - это два вида (специализированные классы) счетов. Затем, мы документируем, какие атрибуты и операции применимы ко всем счетам, какие применимы только к «Расчетным счетам», и какие только к «Сберегательным счетам». Описывая требования таким образом, вам необходимо описать правила, применимые ко всем видам счетов только один раз. Это облегчает работу с документом, если эти бизнес-правила когда-либо будут изменены. Это также дает Вам возможность сформулировать правила работы со счетами, которые применены ко всем будущим типам счетов, даже к тем типам счетов, о которых вы еще не знаете.

ОО Концепция: Ассоциация



Определение:

«Ассоциация указывает семантические отношения, которые могут возникнуть между экземплярами. Экземпляр ассоциации называется ссылкой.» [7] (UML 2)

- «Ассоциация – это наиболее часто используемый тип отношения между сущностями. Отношение ассоциации имеет место, если одна сущность непосредственно связана с другой (или с другими – ассоциация может быть не только бинарной)» (Book)

Что это означает:

Ассоциация между двумя классами указывает, что объект (экземпляр) одного класса может быть связан (ссылаться) на объект другого класса. Ассоциация указывается на как на уровне класса, так и на уровне объектов.

Взгляд БА

Вы анализируете, как по бизнесу связаны объекты одного класса с объектами другого класса (или, иногда, с другими объектами того же класса). Как БА, вы документируете такого рода правила, как ассоциации. Это важная часть вашей работы. Игнорировать такие ассоциаций неправильно. В результате вы можете разработать программное обеспечение, которое не реализует бизнес-правила.

ОО Концептуия: Агрегация



- **Определение:**

«Агрегация: специальная форма ассоциации, которая определяет отношений часть-целое между совокупностью (целое) и составной частью» (UML)

- «Агрегация – это ассоциация между классом А (часть) и классом В (целое), которая означает, что экземпляры (один или несколько) класса А входят в состав экземпляра класса В» (Book)

-

Что это означает:

- Формально, в UML, агрегация считается особым типом ассоциации, где класс на одном конце ассоциации представляет «целое», и класс на другом конце ассоциации представляет «часть». Агрегирование может быть использовано БА в качестве альтернативы моделирования ассоциации с названием "является частью".
- В агрегации, «часть» может принадлежать более чем одному «целому». Например, объект каталог является коллекцией (агрегацией), который состоит из многих объектов товаров. Однако какой-либо конкретный объект товара может появиться в более чем одном объекта каталога.

-

Взгляд БА

Вы отыскиваете бизнес объекты, которые составляются (состоят) из других бизнес объектов. Вы моделируете эти отношений, как «агрегация». Затем вы концентрируетесь на том, какие правила (атрибуты, операции и отношения) применяются ко «целому» и какие применимы к «частям». Например, вы моделируете карты В-банка, как некоторую агрегацию, одной из частей которого является PIN-код. Вы определяете атрибуты и операции PIN-кода. Позже вы может повторно использовать PIN требования для другой системы кредитных карт, которая также использует PIN-коды.

ОО Концепция: Полиморфизм



- **Полиморфизм означает способность принимать различные формы. Термин применяется как к объектам и к операциям.**

Мы рассмотрим:

Полиморфные объекты

Полиморфные операции

Взгляд БА

Полиморфные объекты

- Предположим, финансовая компания управляет различными подтипами фондов, такие как Азиатский Фонд, Внутренний Фонд, и так далее, каждый со своими особенностями. БА моделирует эту ситуацию используя обобщенный (generalized) класс - Фонд и специализированный класс для каждого подтипа фонда. Далее, БА работает над описанием правил «поступление средств» для класса «поступление». В кооперации с заинтересованными лицами, БА выясняет, что одна из операций класса «поступление» выполняется для всех подтипов фондов одинаково. БА в этой ситуации должен убедиться, что операция «поступление средств» обращается напрямую к обобщенному классу, а не к какому-то специализированному классу. Когда операция фактически выполняется, объект фонд примет одну из своих форм, например, Азиатский Фонд или Внутренний Фонд. Другими словами, объект фонд является полиморфным.

Полиморфные операции



- Продолжая тот же пример, так как все подтипов Фонда должны быть в состоянии принимать вклады, БА определяет операцию для класса Фонд «принимать вклады». Эта операция наследуется всеми специализациями класса. БА может также указать метод для этой операции, которая также будет унаследована специализациями. Но что, если одно или несколько специализаций, например, Азиатский Фонд использует другую процедуру для приема вкладов? В этом случае, БА можете добавить в документации описание для класса Азиатский Фонд. Это описание определяют метод, который перекрывает один из унаследованных от обобщенного класса. Например, этот описанный метод, может включать дополнительные расходы.

ОО Концепция: Полиморфизм



Взгляд БА

При определении операций для обобщенного класса, вы выбираете те, которые должны быть в состоянии поддерживать все специализации. Если вы можете, вы определяете способ (метод), который описывает, как операция обычно проводится. Если какой-либо из специализированных классов имеет отличный способ выполнения этой операции, вы определяете новый метод на уровне специализированного класса. Это упрощает документации. Вам не нужно писать: "Если тип X, значит выполнять один метод; если это Y, то выполнять другой" Вместо этого, какой метод (способ выполнения операции) должен быть использован определяется местом, где этот метод вызван.

Один интерфейс, множество реализаций

Полиморфизм означает "один интерфейс, множество возможных реализаций." Автомобили – простой пример проявления полиморфизма. Все они используют одинаковый интерфейс-педаль акселератора для изменения скорости, хотя внутренний метод может отличаться от модели к модели. Автомобильная промышленность разрабатывает автомобили таким образом, чтобы водителям не пришлось привыкать к новому интерфейсу каждый раз используя новую модель автомобиля.

Use Cases и Сценарии



Определение:

" Use case A - это спецификация набора действий, выполняемых системой, дающих наблюдаемый результат, который, как правило, важен для одного или нескольких Actors или других заинтересованных лиц системы. "(UML 2)

Что это означает:

Use case – это использование системы, которое обеспечивает видимый и (обычно) значимый результат. Документация по Use case (схемы и / или текст) должен описывать последовательность шагов, которые выполняются во время взаимодействия с системой и включают в себя различные способы (пути) этого взаимодействия.

Use Cases и Сценарии

Определение:

"Сценарий: определенная последовательность действий, которая иллюстрирует поведение. Сценарий может использоваться для иллюстрации взаимодействия или выполнения экземпляра *Use case*." (UML 2)

Что это означает:

Сценарий один путь использования *Use case*, точнее - один из возможных путей прохождения *Use case*.

Например, сделать оплату счетов *Use case* может быть выполнен одним из следующих способов:

Сценарий 1: Попытка сделать платеж со счета. Попытка заканчивается успешно

Сценарий 2: Попытка сделать платеж со счета. Попытка заканчивается неуспешно потому, что на счете недостаточно средств.

Взгляд БА

Во время анализа, вы выявляете и документируете сценарии использования системы, т.е. то, что пользователи хотят делать с системой. Вы можете сделать это путем выявления и описания его сценариев - всех способов взаимодействия с системой. Эти *Use case* и сценарии являются пользовательскими требованиями.

Бизнес и Системные Use Cases



Со временем, аналитики начали различать два вида **Use Cases**: бизнес use cases и системные use cases. Это различие не является частью основного стандарта UML, но это корректно и реализовано посредством UML расширений.

- **Use case** (общий) описывает взаимодействие с некоторым типом системы. Вопрос в том, какой тип системы имеется в виду?
- **Бизнес Use case** описывает взаимодействие с бизнес системой. Например, процесс выставления претензии является бизнес *Use case*, описывающим взаимодействие со страховой компанией.
- **Системный Use case** описывает взаимодействие с ИТ системой. Например, системные *Use cases*, которые описывают реализацию вышеупомянутого бизнес *Use case* - это «Зарегистрировать претензию», «Проверить случай», «Назначить ответственного» и так далее. Каждый из них описывает взаимодействие пользователя и компьютерной системы. Системные *Use case* обычно включает в себя одного активного (primary) пользователя (actor) и, как правило, выполняется в течение одной сессии. В конце исполнения системного *Use case*, пользователь должен чувствовать, что он или она достигли какой-то значимой, полезной с точки зрения бизнеса, цели.

Шаги Бизнес Объектно Ориентированного Моделирования (БООМ)

В данном разделе рассматриваются шаги Бизнес Объектно Ориентированного моделирования (БООМ), процедуры выявления, анализа, документирования, и тестирования требований с использованием объектно-ориентированных и дополнительных методов.

БООМ и SDLC



Многие крупные компании используют понятие «жизненного цикла разработки системы» (systems development lifecycle - SDLC) для управления IT проектами. SDLC определяет конкретные этапы и действия в рамках проекта. В разных реализациях SDLC названия фаз отличаются, но большинство SDLC используют нечто близкое к следующим этапам:

- **Initiation:** Разработка бизнес Use Cases. Работа начинается с изучения пользовательского опыта и создания обоснования черновика (предварительного варианта) архитектурной концепции. Любые работы по прототипированию на этапе инициирования должны проводиться с учетом рисков и только в объеме, достаточном чтобы убедиться, что такое решение возможно
- **Discovery:** проведение исследований, с целью понимания желаемое поведение системы. Во время этой фазы, архитектурные решения продолжают разрабатываться
- **Construction:** Полный анализ и дизайн, кодирование, интеграция и тестирование программного обеспечения
- **Final Verification and Validation (V&V):** Выполнение окончательного тестирования, прежде чем продукт или сервис будет передан в эксплуатацию. Closeout: Управление и координация развертывания в на эксплуатационном окружении и закрытие IT проекта.

БООМ шаги

Шаг 1: Initiation



Цели фазы Initiation - разработка Бизнес Use Cases для проекта, открытие проекта и определения объема (scope) проекта, а также поиск решений, в том числе предварительные архитектурные решения. БА помогает менеджеру проекта путем выявления заинтересованных лиц, бизнес сервисов и процессов, и IT сервисов, которые будут так или иначе влиять на проект. К концу этой фазы, ключевая функциональность должна быть определена, такие как ключевые системные Use Cases (пользовательских задач) и IT сервисы, с которыми проект будет интегрироваться или использовать.

БООМ шаги

Шаг 1: Initiation



UML методы

Бизнес Use Cases: инструмент для выявления и описания бизнес-процессов, значимых для проекта

Диаграммы активности (Activity diagrams): Используется, чтобы помочь Вам и заинтересованным лицам сформировать общее понимание workflow каждого бизнес Use Cases

Actors: Описывают пользователей и внешние системы, которые будут взаимодействовать с будущей IT системой

Системные Use Cases: Используется, чтобы помочь заинтересованным лицам, как Бизнес процессы разбиваются на взаимодействия с IT системой
Главным результатом этого этапа является предварительный вариант документа бизнес требований (**business requirements document BRD**)

БООМ шаги

Шаг 1: Initiation



Ниже приведен список шагов, которые вы будете проводить на этом этапе.

1, а) моделирование бизнес use cases

- Определите бизнес use cases (диаграммы бизнес use cases)
- Содержание бизнес use cases (Диаграмма деятельности activity diagram) 1, b) моделирование системных use cases Определить actors (карту ролей role map)
- Определение пакетов системных use cases (диаграмма системных use cases)
- Определение системных use cases (диаграмма системных use cases)

1, с) Начало создания структурной модели (диаграммы классов для ключевых бизнес классов)

1, d) Определение baseline (BRD / Initiation)

БООМ шаги

Шаг 2: Discovery



Основная цель фазы **Discovery** - понять желаемое поведение будущей системы и принять основные базовые архитектурные решения. Эта и предыдущая фаза - основные этапы для БА. Процесс анализ требований достигает пика во время этой фазы. Некоторые системные use cases выбираются на этом этапе, с тем чтобы продемонстрировать работоспособность принятых архитектурных решений (прототипирование).

БА на этом этапе должен сосредоточиться на выявлении подробных требования в процессе интервьюирования заинтересованных лиц, анализе и документировании их для предоставления на проверку заинтересованным лицам. Вы будете использовать большое количество UML и дополнительных методов для в процессе выявления требований, анализе и документировании на этом этапе. Некоторые из основных методов, которые Вы будете использовать, включают следующее:

- **Описание системных use-case** (спецификаций), пошаговом описании взаимодействия между пользователями и предлагаемой IT системой
- **Диаграмма состояний (State-machine diagrams)**, описывающая жизненный цикл ключевых бизнес объектов
- **Диаграмм классов (Class diagrams)**, описывающая ключевые бизнес концепции и бизнес-правила, которые применяются к бизнес объектам, таким как счета, инвестиции, жалобы, претензии и т.д.

BOOM шаги

Шаг 2: Discovery



- **Ниже приведены шаги, которые вы будете проводить на этом этапе**
 - 2а) Анализ поведения**
 - Описать системные **use cases** (шаблон описания **use cases**)
 - Описать поведение состояний (Диаграмма состояний **state-machine diagram**)
 - Определить состояния критически важных объектов
 - Определить переходы состояний - определить условия переходов
 - Определить конкурирующие состояния

БООМ шаги

Шаг 2: Discovery



2, b) Структурный анализ (модель объектов/данных) (диаграммы классов)

- Определение классов сущностей
- Моделирование обобщений
- Моделирование отношений «часть-целое»
- Анализ связей
- Анализ множественности
- Отображение набора системных use cases на в структурную модель
- Добавление атрибутов
- Описание операций
- Анализ и переработка структуры классов

2, в) Разработка артефактов тестирования (тест план / таблица решения)

- Определение белого white-box test cases
- Определение black-box test cases
- Определение системных тестов

2, d) Определение плана реализации (implementation plan)

2,е) Установить baseline для разработки (BRD / Discovery)

Шаг 3: Construction



Бизнес-анализ во время этой фазы зависит от используемого жизненного цикла. На водопадных проектах, в которых анализ и выявление требований на этом этапе, как правило, уже закончен. Однако, БА участвует в поддержке процесса тестирования и проверки того, что технический дизайн отвечает требованиям (проведения ревью тест планов и спецификаций дизайна). В итерационных проектах (Agile), анализ требований и выработка решений выполняются в течение нескольких итераций, шаги, описанные для стадии Discovery (шаги 2A-2e) проводится в течении каждой итерации стадии Construction.

Шаг 4: Final Verification and Validation (V&V)



Бизнес-аналитик осуществляет поддержку на этапе окончательного тестирования, до момента окончательное решение будет развернуто, проводя анализ планов тестирования и результатов тестирования и обеспечение того, что все требования протестированы

Шаг 5: Closeout



Бизнес-аналитик поддерживает процесс развертывания, проводит ревью планов развертывания и участвует в post-implementation review (PIR), чтобы оценить успешность разработанных решений



Шаг 1: The Initiation Phase

Фаза Initiation



Первый этап проекта - этап **Initiation**. Различные подходы к управлению ИТ проектами имеют свои собственные термины для этой фазы и конкретные наборы активностей, которые выполняются в рамках данного этапа. Возможные названия для этой фазы могут быть следующие:

- Envisioning (Microsoft Solutions Framework, MSF): В этой главе рассматриваются следующие цели MSF относительно фазы выработки концепции: " High-level view целей проекта", "Бизнес требования должны быть определены и проанализированы", - <http://technet.microsoft.com/en-us/library/bb497060.aspx>
- Inception (RUP) <http://www-01.ibm.com/software/awdtools/rup/> #
- Initiate (Project Management Institute - PMI) <http://www.pmi.org/>

Что происходит на этапе Initiation?



На этапе инициации, проект вырастает из идеи, в предварительный вариант предложения (proposal), в котором излагаются основные аспекты проекта и описываются основные причины его создания. На этом этапе ваша работа в качестве бизнес-аналитика заключается в выявлении и анализе бизнес-требований для проекта. Вы определяете высокоуровневые бизнес-цели и бизнес use cases. Вы будете работать с заинтересованными лицами для анализа их деятельности. Будете описывать это с помощью диаграмм бизнес use-case. И вы будете общаться с заинтересованными сторонами с целью описания workflow каждого бизнес use-case с использованием диаграмм деятельности (activity diagrams)

Шаг Initiation: BRD

Некоторые другие виды документации, разрабатываемые на этапе **Initiation** включают следующее:

- **Opportunity evaluation:** Документы, описывающие преимущества, которые будут получены в случае реализации проекта
- **Project vision and scope:** Описание того, что планируется достигнуть в рамках данного проекта
- **Product vision and scope:** описание целей программного продукта

Ключевые компоненты **BRD** разрабатываемые на этапе **Initiation:**

- Описание **Бизнес Use Cases** (описанные в **RUP** как спецификации), в том числе **Диаграммы Бизнес Use Cases**
- **Карта ролей (Role map)**
- **Диаграммы Системных Use Cases**
- **Первоначальная диаграмма классов**, описывающая ключевые бизнес классы

Шаг 1а: Модель Бизнес Use Cases

"A business use case defines what should happen in the business when it is performed; it describes the performance of a sequence of actions that produces a valuable result to a particular business actor [someone external to the business]." (*Source: Rational Rose*)

- Бизнес **use case** определяет, что должно произойти в бизнесе, когда **use case** выполняется. Он описывает последовательность выполнения действий, которые производят к ценному с точки зрения бизнеса, результату для конкретного бизнеса **actor** [некто внешний по отношению к бизнесу]

Диаграмма Бизнес Use-Cases

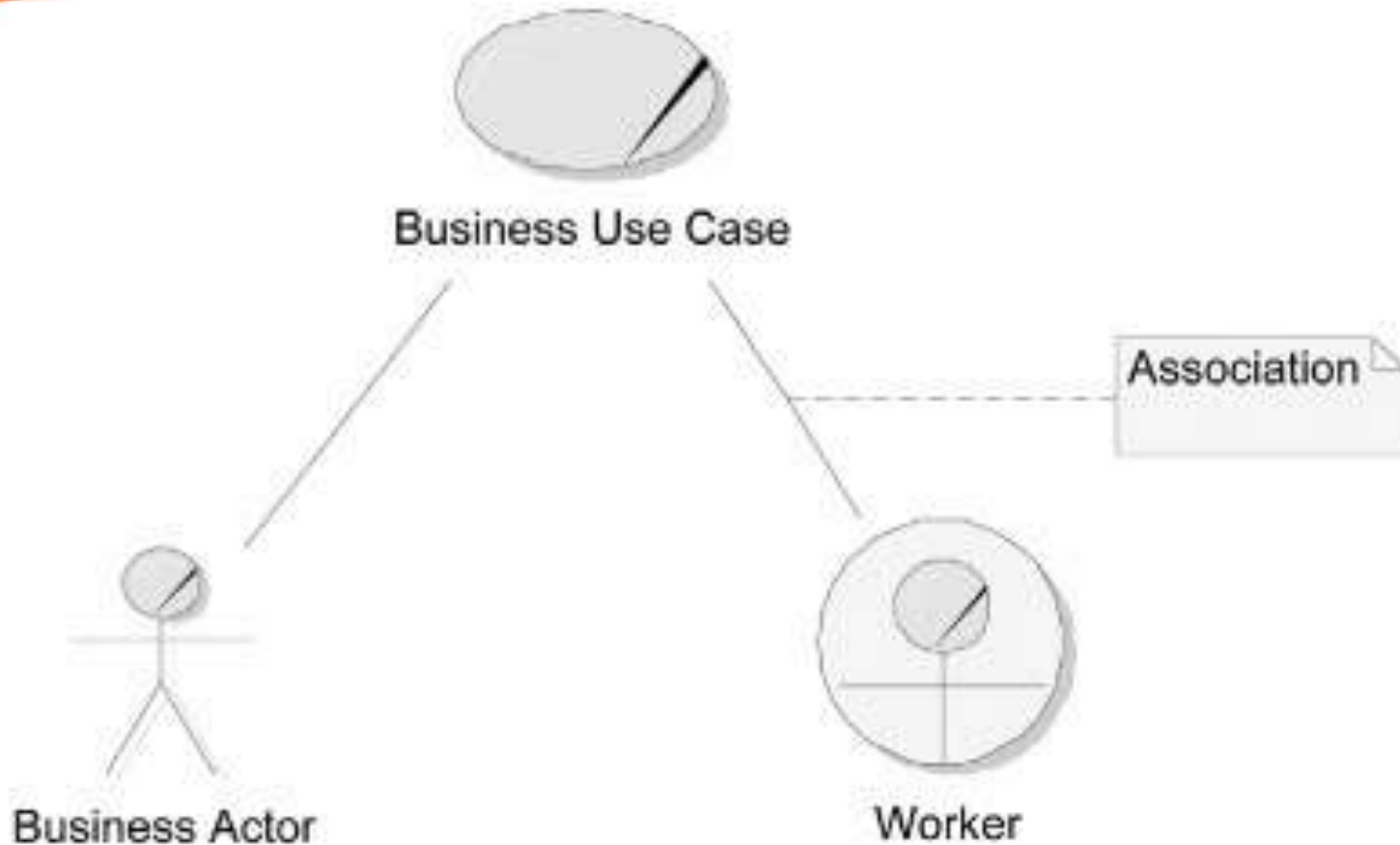


Figure 4.1: Business use-case diagram symbols.

Элементы Диаграммы Бизнес Use-Cases



Элементы моделей:

- **Business actor:** Кто-то внешним по отношению к бизнесу, например, клиент или поставщик
Worker: Тот, кто работает в бизнесе, например, сотрудник
- **Association:** связь между actor и бизнес use case. Показывает, что actor взаимодействует с бизнесом в рамках **бизнес use case**, например, запускает **use case** к действию или исполняет шаги **use case**

Другие элементы модели

в бизнес моделировании иногда используются другие типы **actor**. *The UML Extension for Business Modeling, version 1.1*, например, позволяет разделять **worker** на *case workers* и *internal workers*

- **Case worker:** *работник, который непосредственно взаимодействует с actors из вне системы;*
- **Internal worker:** *работник, который взаимодействует с другими работниками и организациями внутри системы*

Шаг 1aii: Содержание Business Use Cases (Activity Diagram)



Now that you have a business use-case diagram that matches up stakeholders with business processes, you can begin to plan the next stage of the **interviews**. Each interview should focus on a subset of the business use cases. Be sure to invite all stakeholders associated with the use case (as shown on the diagram) as well as off-stage stakeholders—those who do not directly interact with the process but still have a stake in it, such as regulators and high-level management.

The purpose of these interviews is to **analyze** the **workflow** of each **business use case**. Workflow means the sequencing of activities and (optionally) a clear designation of who carries out each activity. Workflow can be documented in text and/or through the use of a workflow diagram.

Diagrams for Depicting Workflow

Diagram	Description	Advantages	Disadvantages
System flowchart	Earliest form for depicting sequencing of activities.	<ul style="list-style-type: none">▪ Intuitive. Each type of input and output is clearly marked with its own symbol.▪ Includes logic symbols.	<ul style="list-style-type: none">▪ Not compliant with UML.▪ Can be hard to learn (many symbols).
Swimlane workflow diagram	<p>Tool used for describing process logic.</p> <p>UML equivalent is an activity diagram with partitions (swimlanes).</p>	<ul style="list-style-type: none">▪ Intuitive.▪ Can handle many situations in one diagram▪ Shows who is responsible for which action (using swimlanes).	<ul style="list-style-type: none">▪ Not compliant with UML.

Diagrams for Depicting Workflow

Diagram	Description	Advantages	Disadvantages
Sequence diagram	UML tool used to describe one path (scenario) through a use case.	<ul style="list-style-type: none">▪ Part of UML standard.▪ Encourages thinking in objects. Clearly specifies who does what.▪ Simplifies logic: Only one situation dealt with in each diagram.▪ Sometimes recommended for business modeling.	<ul style="list-style-type: none">▪ Diagramming style is often non-intuitive for business analysts and users.▪ Requires analyst to determine not only who carries out each activity, but who requests it.

Diagrams for Depicting Workflow

Diagram	Description	Advantages	Disadvantages
Activity diagram	<p>UML tool for describing logic. Used to describe entire system, a use case, or an activity within a use case. Has two versions:</p> <ul style="list-style-type: none">•Activity diagram without partitions (swimlanes):Does not show who does what.•Activity diagram with partitions: shows who does what.	<ul style="list-style-type: none">▪ Part of UML standard.▪ Can handle many situations in one diagram.▪ Simple diagramming conventions.▪ Encourages thinking about opportunities for parallel activities (more than one activity going on at the same time).	<p>Ability to handle many situations can lead to a diagram that is too complex to follow.</p>

Diagrams for Depicting Workflow

Diagram	Description	Advantages	Disadvantages
Business process diagram (BPD)	Business process modeling notation (BPMN) tool for describing workflow	<ul style="list-style-type: none">▪ Part of BMN standard, managed by the OMG▪ Rich symbol set can model complex and subtle workflow requirements better than activity diagrams.	<ul style="list-style-type: none">▪ Not UML-compliant▪ Difficult to understand without prior training

Activity Diagrams for Describing Business Use Cases



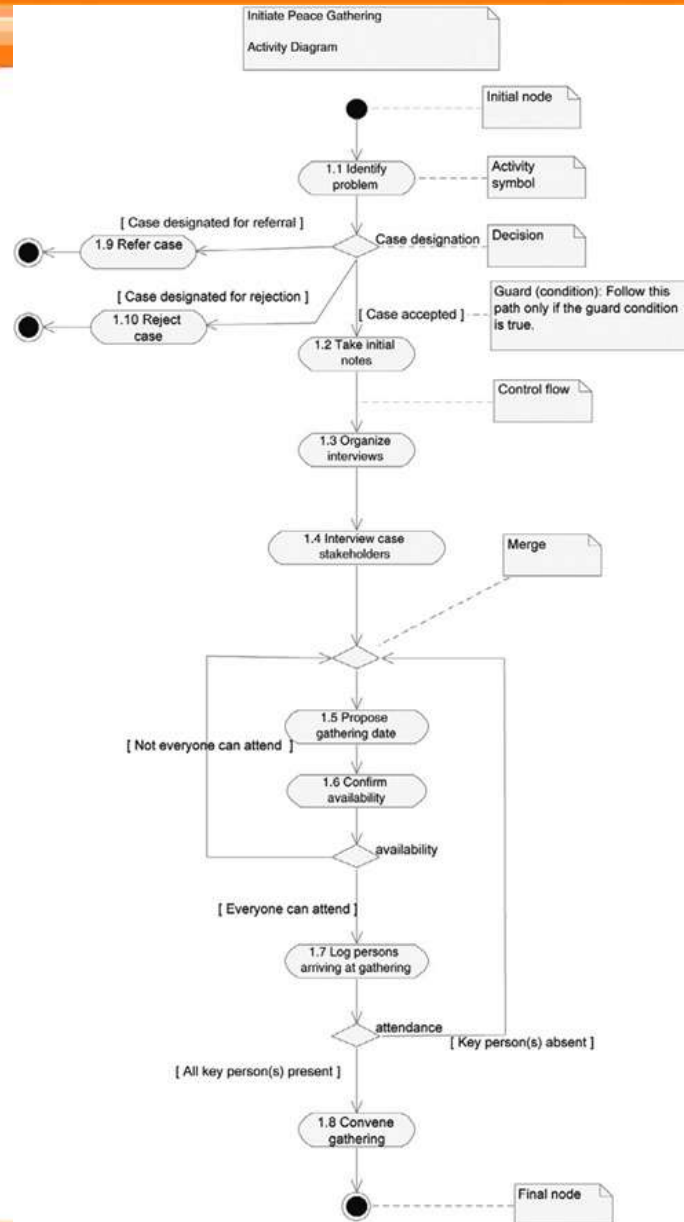
- ◆ *The activity diagram is the one most useful to the IT BA for depicting workflow. It is simple to understand—both for BAs and end-users.*

Activity Diagram Elements

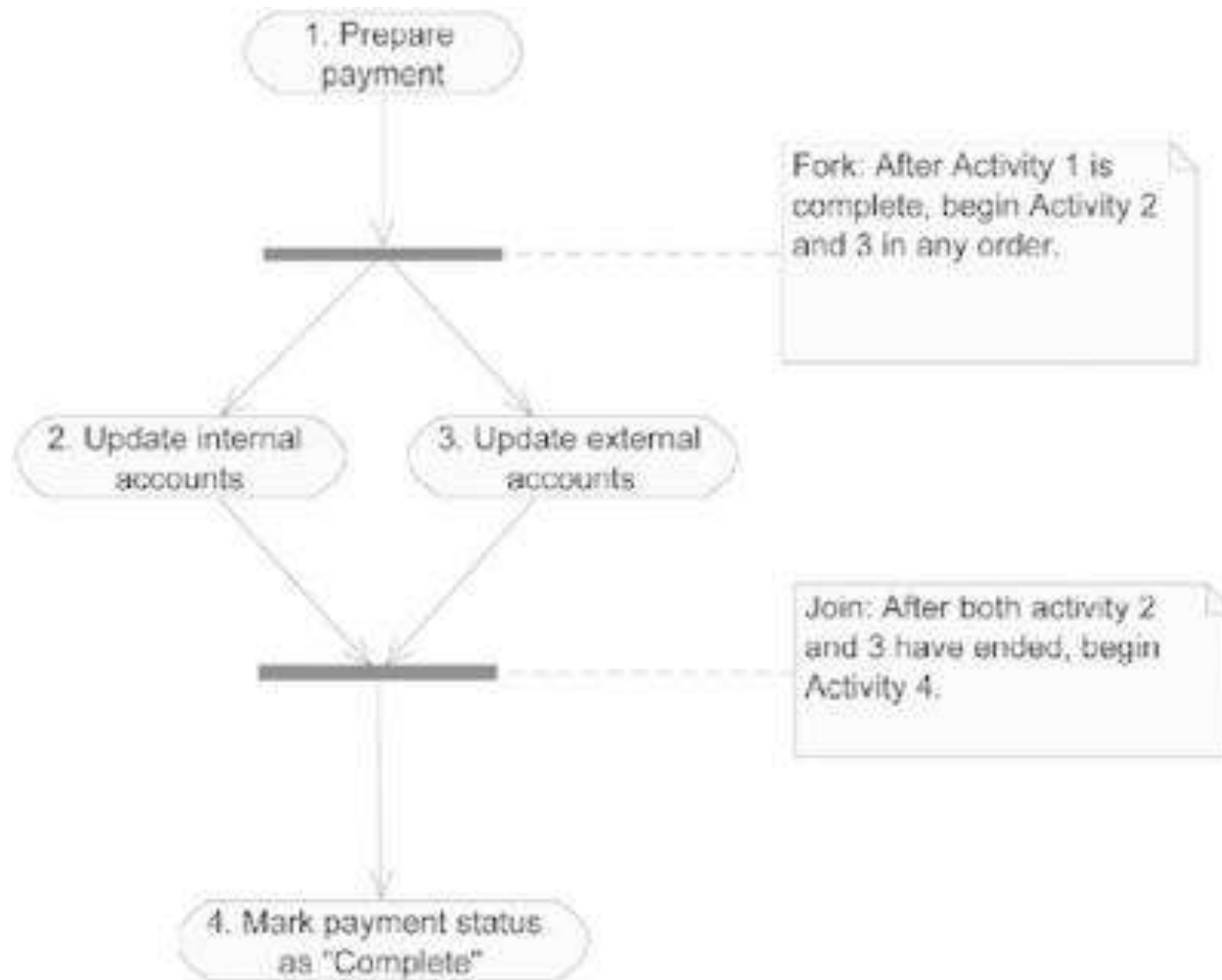
Activity diagrams may include the following elements:

- **Initial node:** Indicates where the workflow begins.
- **Control flow:** An arrow showing the direction of the workflow.
- **Activity:** Indicates a step in the process.
- **Decision:** A diamond symbol, indicating a choice. Workflow will proceed along one of a number of possible paths, according to the guard conditions.
- **Merge:** Use this symbol if you wish to adhere to strictly to the UML standard when modeling a number of alternative flows that lead to the same activity.
- **Guard condition:** A condition attached to a control flow. When the guard condition is true, workflow may flow along the control flow. Guard conditions are usually attached to control flows that come out of a decision symbol. A guard is shown within square brackets.
- **Event:** A trigger attached to a control flow. The event must occur for the flow to move along the control flow. Declaring something as an event has a stronger implication than calling it a guard. An event is indicated without the use of square brackets.
- **Fork and join:** Bars used to document parallel activities. In the UML, parallel activities are those that may begin in any sequence either at the same time or one before the other. A fork indicates the point after which a number of activities may begin in any order. A join indicates that workflow may commence only once the parallel activities that flow into it have all been completed.
- **Final node:** Indicates the end of the process.

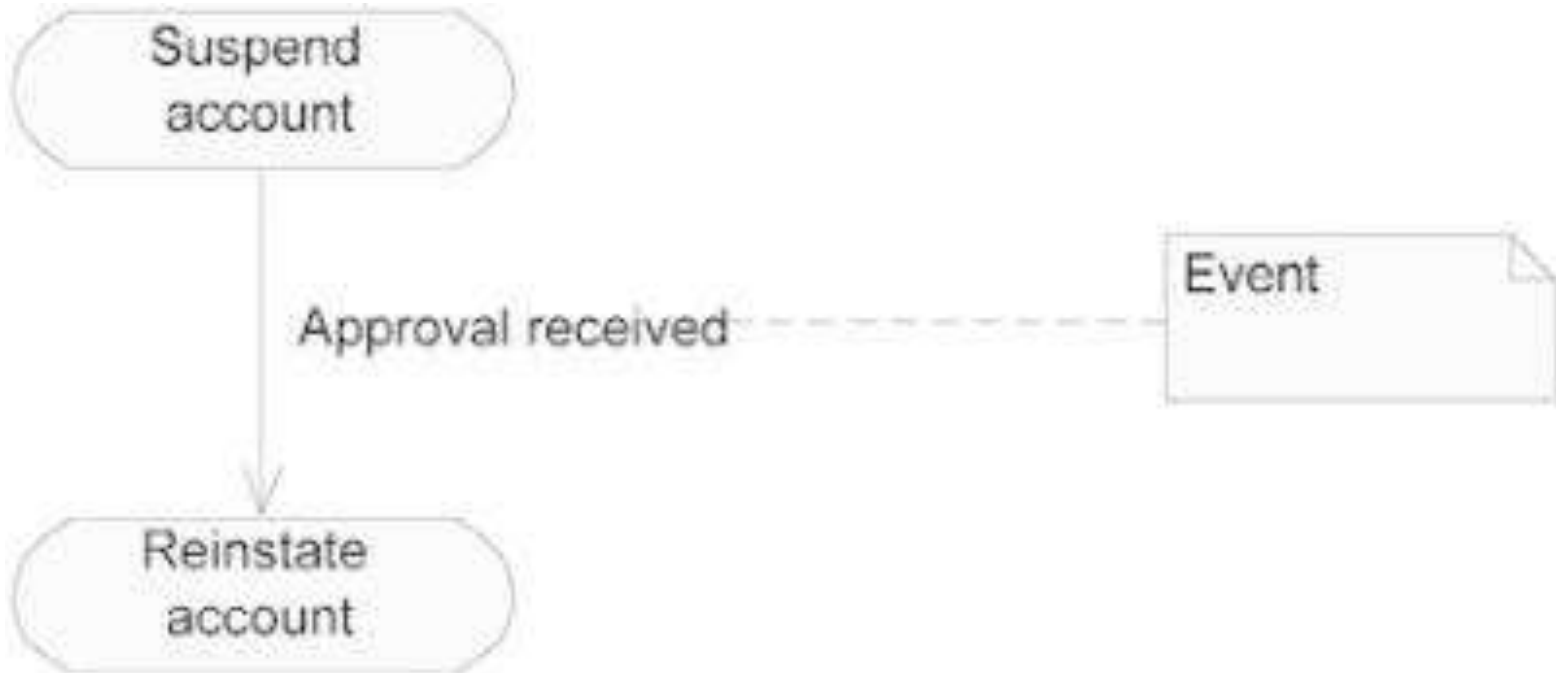
Activity Diagram



The use of fork and join



shows a control flow labeled with an event



Nested Activities

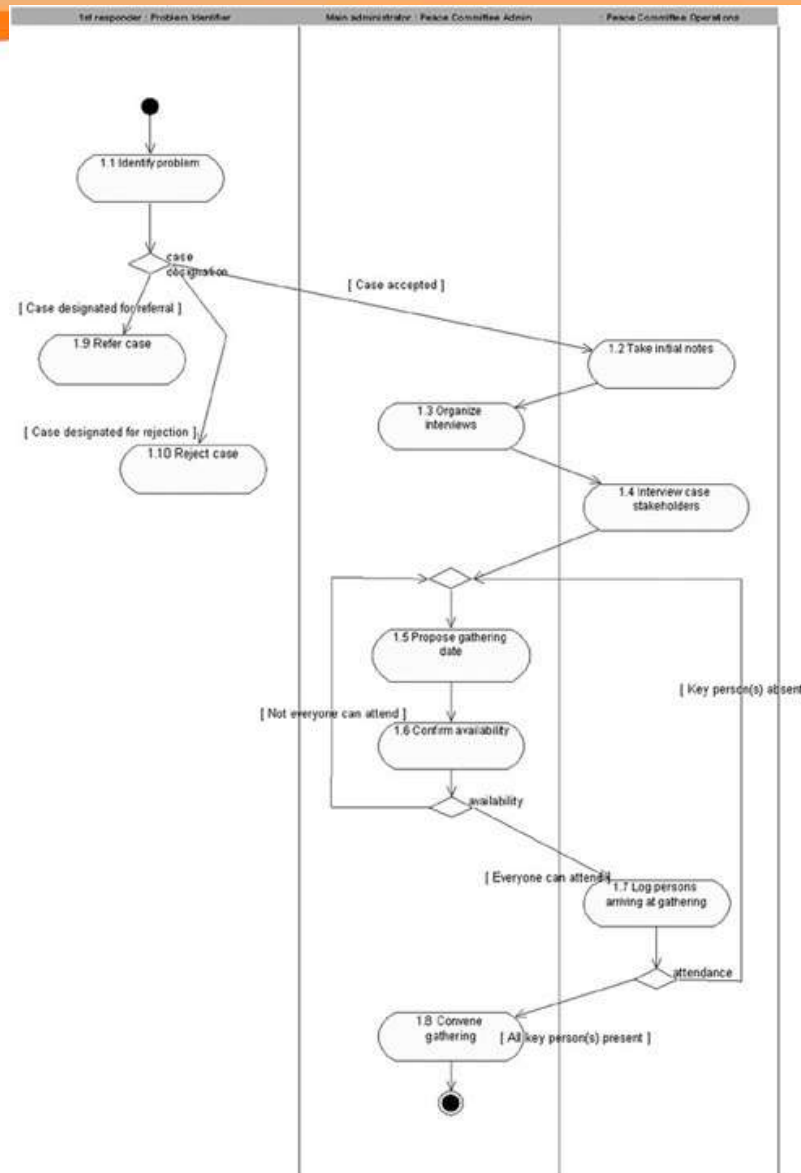


Activity Diagram with Partitions (Swimlanes)

To indicate who performs each activity, you add partitions (commonly referred to as swim-lanes) to the activity diagram. A *partition* is depicted as a column (or row) on an activity diagram. Allocate one partition for each object that takes an active part in the process flow. Each partition represents a stakeholder (business actor or worker) who carries out some activity. Although you shouldn't spend too much time focusing on technology at this time, you may also show a computer system as a partition.

Position every activity in the partition of the object that performs it. Name each partition at the top of the column, according to the participating object

Activity Diagram with Partitions (Swimlanes)



Chapter Summary



In this chapter, you learned the following concepts:

- A **business use case** is an interaction between a stakeholder and the business, yielding a valuable result for the stakeholder; a business process.
- A **business actor** is a stakeholder outside the business that interacts with it, such as a customer or supplier.
- A **worker** is a stakeholder who works within the business, such as a customer-service representative.
- A **business use-case diagram** is a diagram depicting business use cases and their associations with actors.
- An **activity diagram** is a diagram that depicts the sequencing of activities.
- An **activity diagram with partitions (swimlanes)** is a diagram that depicts the sequencing of activities and the object that performs each activity.
 - A **guard** is a condition that restricts flow along a transition.
 - A **control flow** shows the direction of the workflow.
 - An **event** is a trigger that forces the end of an activity and flow to continue along the control flow that it labels.
 - A **decision** is a diamond symbol that marks a point at which flows diverge based upon some condition.
 - A **merge** is a diamond symbol that marks a point at which flows merge. If any of the activities leading into a merge have completed, flow will continue beyond the merge. Use the merge to avoid having more than one incoming flow for an activity.
 - A **fork** marks a point after which parallel activities begin. Activities that are parallel may occur simultaneously or in any sequence.
 - A **join** marks the end of parallel activities. All parallel activities must complete before a flow moves beyond a join

Questions and comments



Questions ?