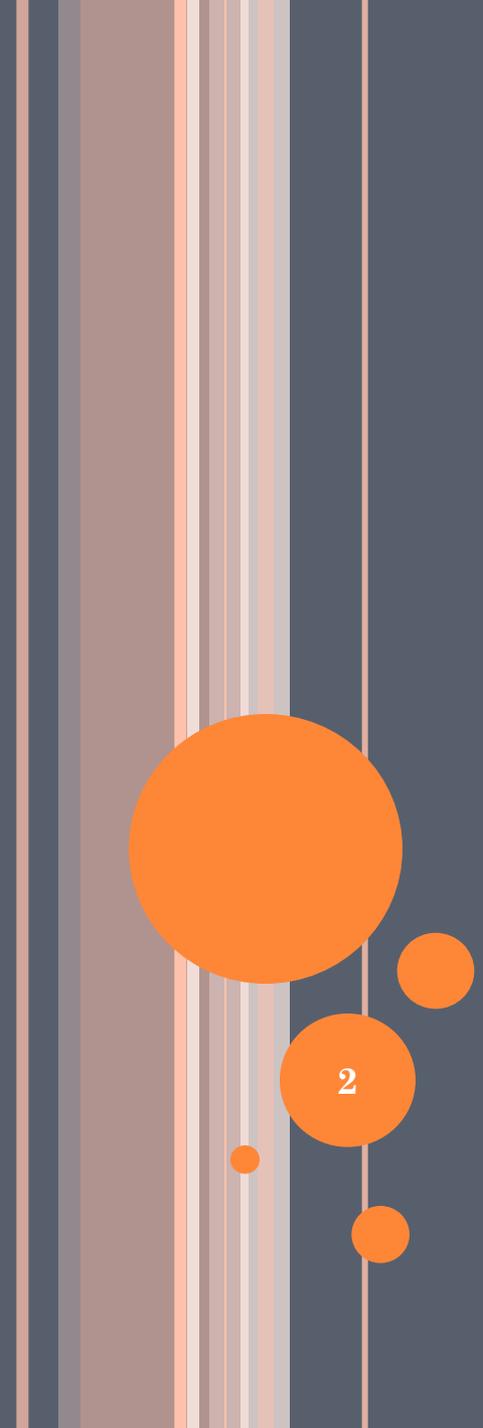




**ИНТЕЛЛЕКТУАЛИЗАЦИЯ ПРОЦЕССОВ  
ОБРАБОТКИ ПОТОКОВ ДАННЫХ, ЛЕКЦИЯ  
6**

**БГУ  
РФиЭ**

**Кафедра Интеллектуальных Систем  
Адуцкевич Иван Анатольевич**



# ПОВТОРЯЮЩИЕСЯ ПРОЦЕССЫ

2

# ПОВТОРЯЮЩИЕСЯ ПРОЦЕССЫ

- Повторяющийся процесс может быть запущен из другого процесса
- Повторяющийся процесс может быть запущен внешней системой используя WSDL
- Повторяющийся процесс может быть внедрен в качестве сервиса или службы, например:
  - Служба форматирования даты/времени
  - Служба проверки бизнес-правил
- Повторяющиеся процессы и под-процессы:
  - Используйте повторяющиеся процессы в случае разработки сервиса; Используйте подпроцесс для абстрагирования от технических деталей.
  - Под-процесс упрощает общий вид процесса и делает логику процесса более наглядной с точки зрения бизнеса
  - Повторяющиеся процессы это процессы, протекающие на сервере отдельно от процессов, вызвавших их.

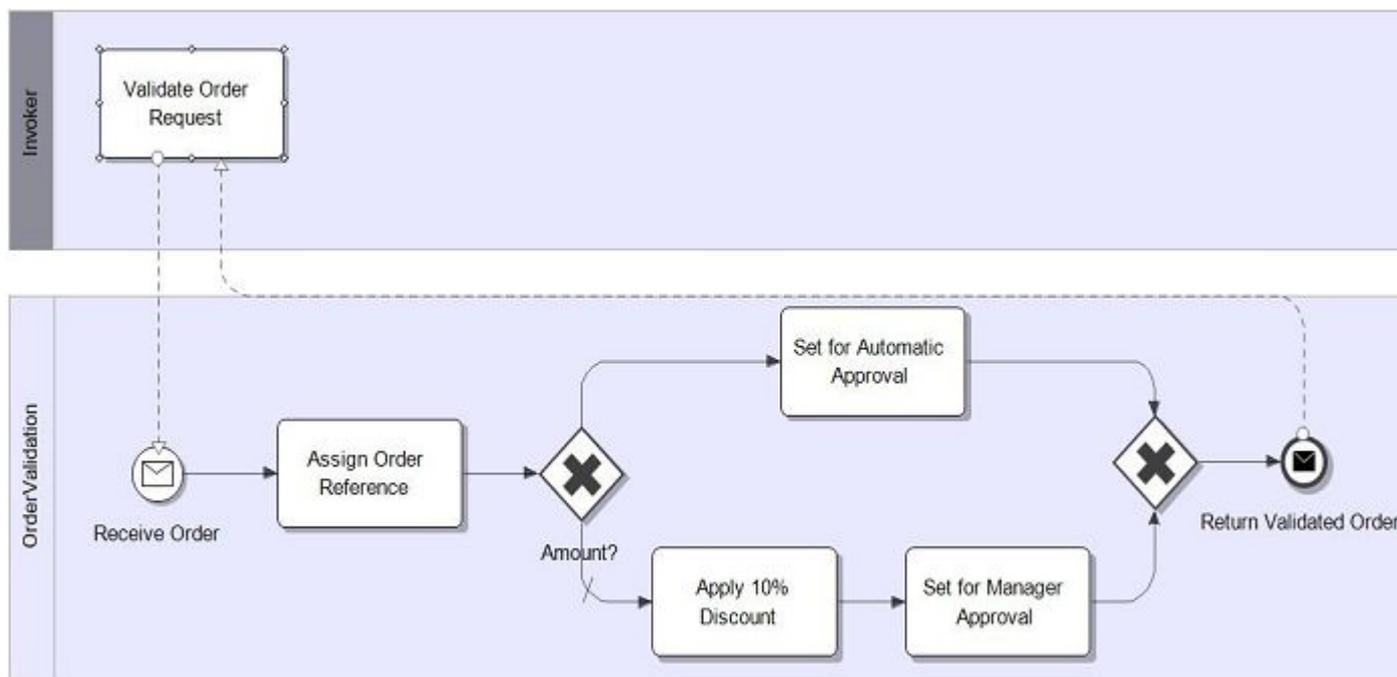


## МОДЕЛИРОВАНИЕ ПРОЦЕССА ПРОВЕРКИ ЗАКАЗА



Спроектируем модель процесса проверки заказа так, чтобы она запускалась при получении заказа. Тогда если цена заказа меньше 5000\$ - заказу не требуется подтверждение менеджера; иначе учесть 10%-ю скидку на товар и отправить заказ на подтверждение менеджеру.

В конечном счете, заказ вернется к тому, кто его совершил.



## СЛУЖБЫ (SERVICES)

- Службы – это действия, совершаемые другим участником, которые определяются:
  - Именем (названием);
  - Сообщением или сообщениями;
  - Конечной точкой – сущностью, процессом или ресурсом, которому можно адресовать сообщение
- Служба не всегда является веб-сервисом
- Веб-сервис это служба, поддерживающая протокол SOAP/HTTP
- Любая служба (не только веб-сервис) может быть описана на языке WSDL



# WSDL

- WSDL – язык описания веб-сервисов
- Стандарт языка предоставлен W3C
- WSDL это, по сути, XML, который описывает веб-сервис: его действия и сообщения
- WSDL используется для определения всех действий, доступных в BPEL
- ESB отображает все сервисы как WSDL



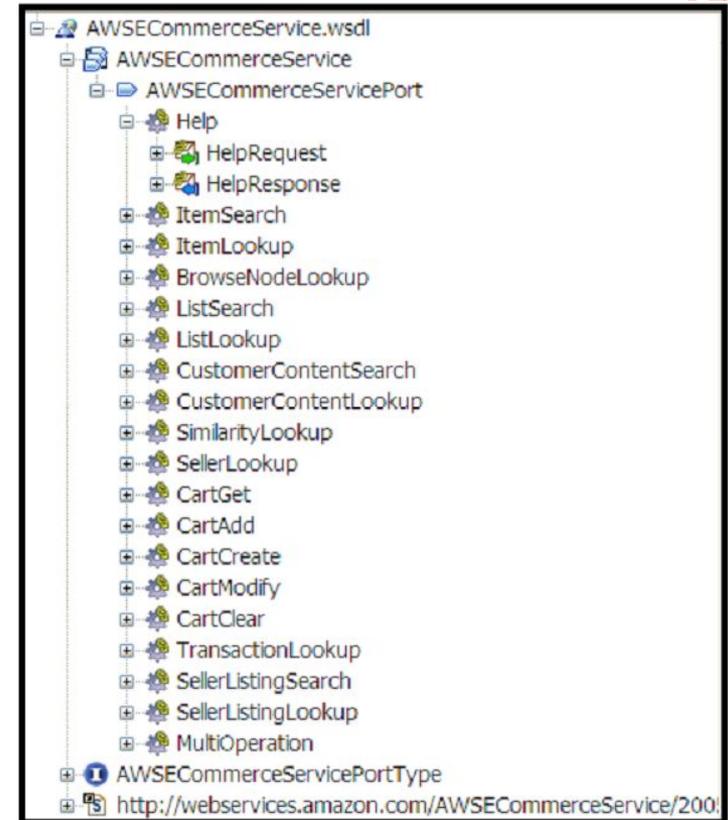
# УПРАВЛЕНИЕ СЛУЖБАМИ

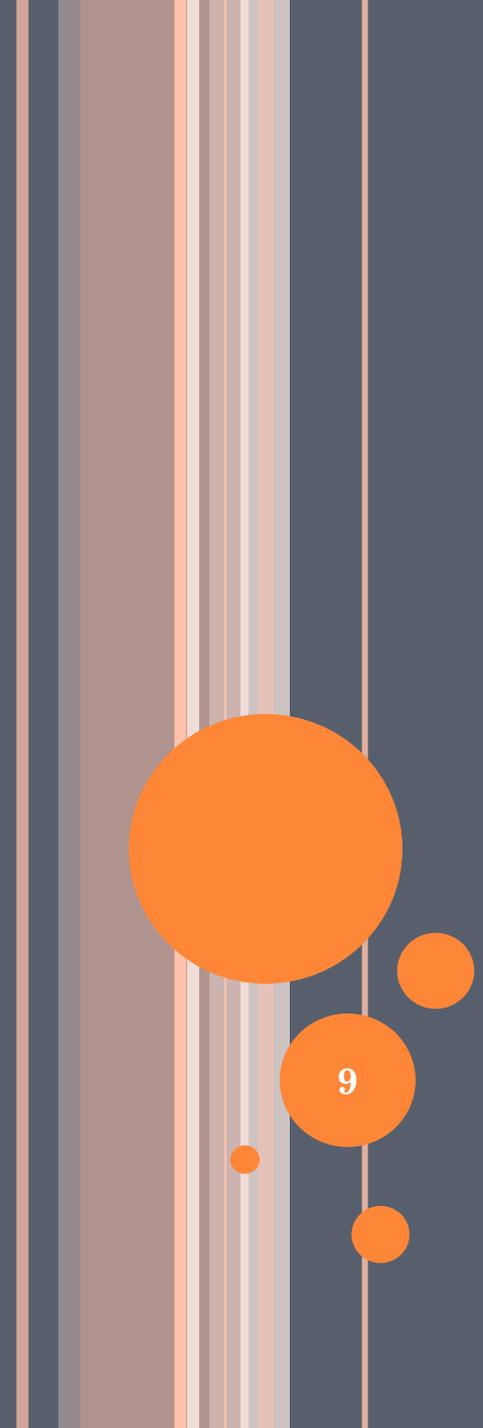
- Службы могут быть предоставлены:
  - Веб сервисами
  - Приложениями
  - Процессами
  
- В любом случае, службы определяются в WSDL
  - Веб-сервисы: предоставляются реестром
  - Приложения: предоставляются специальными связывающими приложениями (созданными конструктором соединений)
  - Процессы: создаются проектировщиком проекта



## УПРАВЛЕНИЕ СЛУЖБАМИ: ВРЕМЯ РАЗРАБОТКИ

- Внешние службы описаны в WSDL как процесс
- WSDL определяет интерфейс для бизнес-приложения по типу сообщения (структуре, имени, типу операции...)
- Intalio | Designer предоставляет WSDL менеджер соединений
- Операции, структуры сообщений, типы сообщений графически отображены в обозревателе проекта (process explorer).
- Задача также может предоставлять или вызывать службу





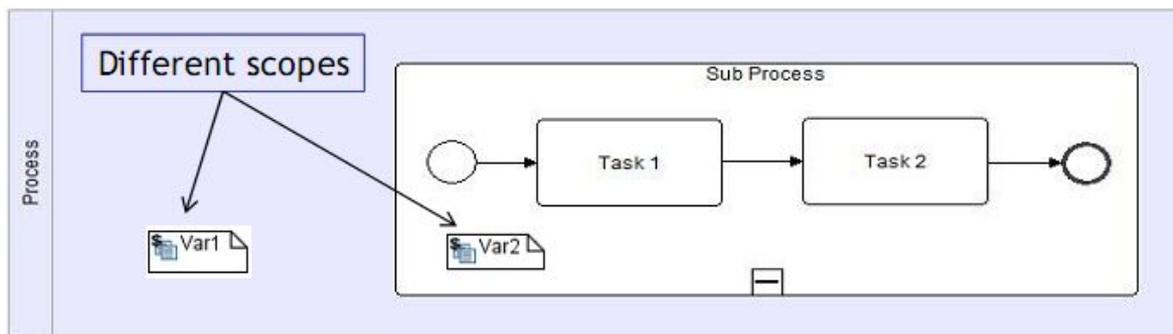
# УПРАВЛЕНИЕ ПЕРЕМЕННЫМИ

9

# УПРАВЛЕНИЕ ПЕРЕМЕННЫМИ

- ❑ Переменные используются для временного хранения значений
- ❑ Создать переменную очень просто – достаточно объявит ее структуру в виде схемы XML-файла и перетянуть элемент в исполняемый блок или под-процесс. Или используя всплывающую панель, которая появляется при выборе какого-либо блока.
- ❑ Каждая переменная имеет свою область видимости. Только исполняемый блок и под-процессы могут получать доступ к переменным
- ❑ Если все под-процессы находятся внутри исполняемого блока, переменные, объявленные внутри исполняемого блока могут считаться **глобальными** переменными
- ❑ Графически переменные отображаются так же как и BPMN данные

10



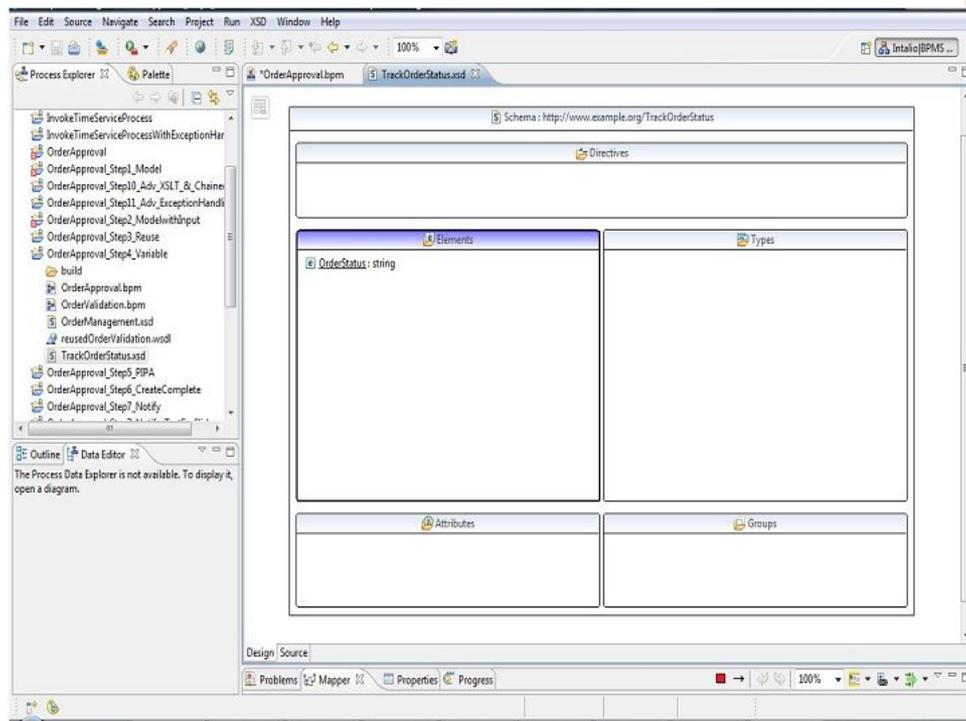
# УПРАВЛЕНИЕ ПЕРЕМЕННЫМИ

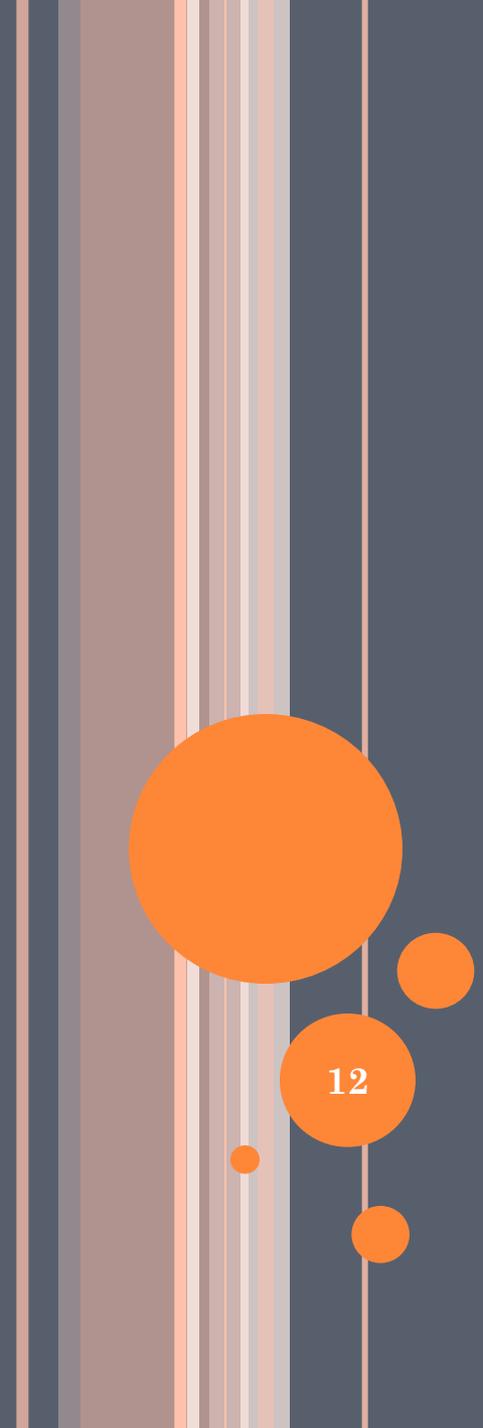


- Создадим переменную, которая будет отображать статус заказа в течение процесса

Создадим схему XML-файла:

- Меню **File** -> **New** -> **Other** -> **XML Schema** .  
Назовем ее **TrackOrderStatus**
- Откроется редактор схемы xml-файла
- Щелкнем правой кнопкой мыши на области элементов -> выберем **добавить элемент (Add element)**
- Появится новый элемент. Изменим его имя на **OrderStatus**
- Оставим ему тип «строка» («**String**»)

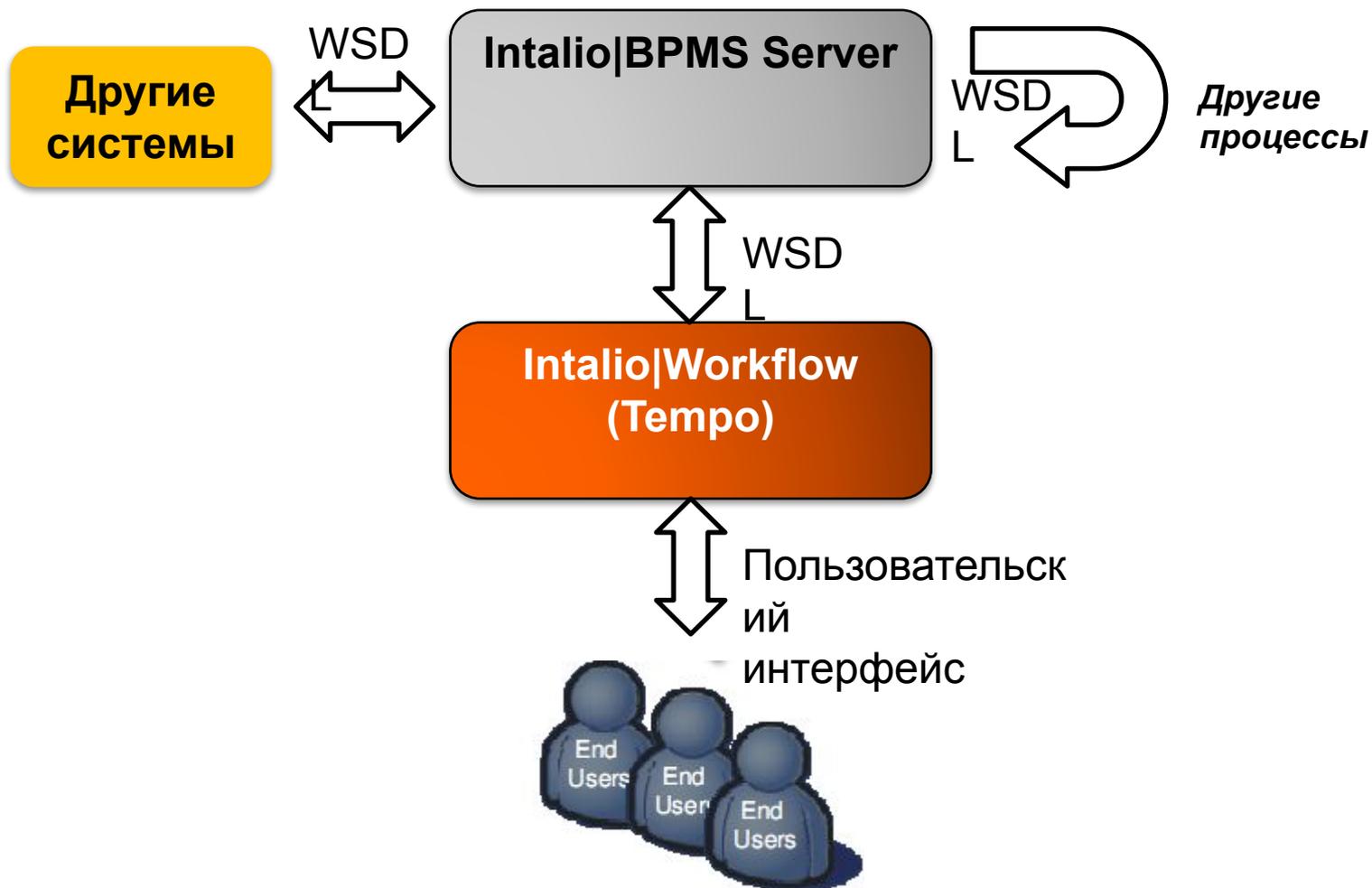




# БИЗНЕС ПРОЦЕСС С УЧАСТИЕМ ЧЕЛОВЕКА

12

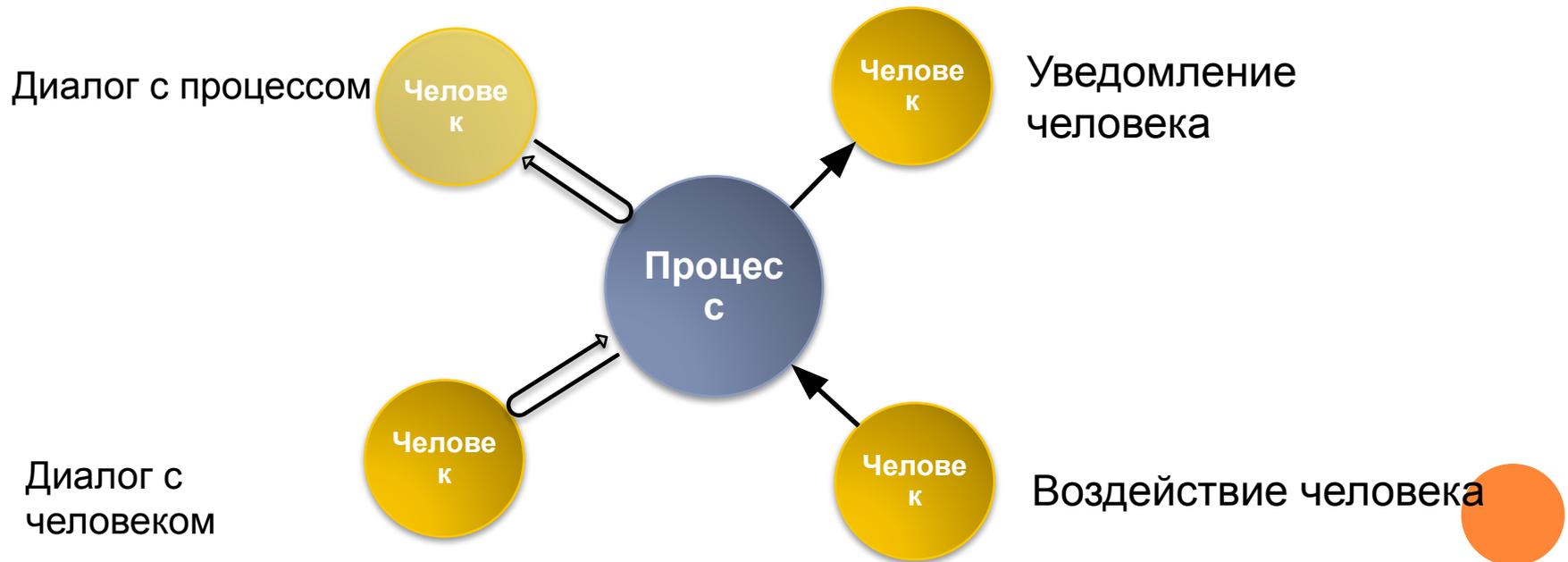
# КАК ЧЕЛОВЕК ВЗАИМОДЕЙСТВУЕТ С ПРОЦЕССОМ



# ВЗАИМОДЕЙСТВИЕ С ЧЕЛОВЕКОМ

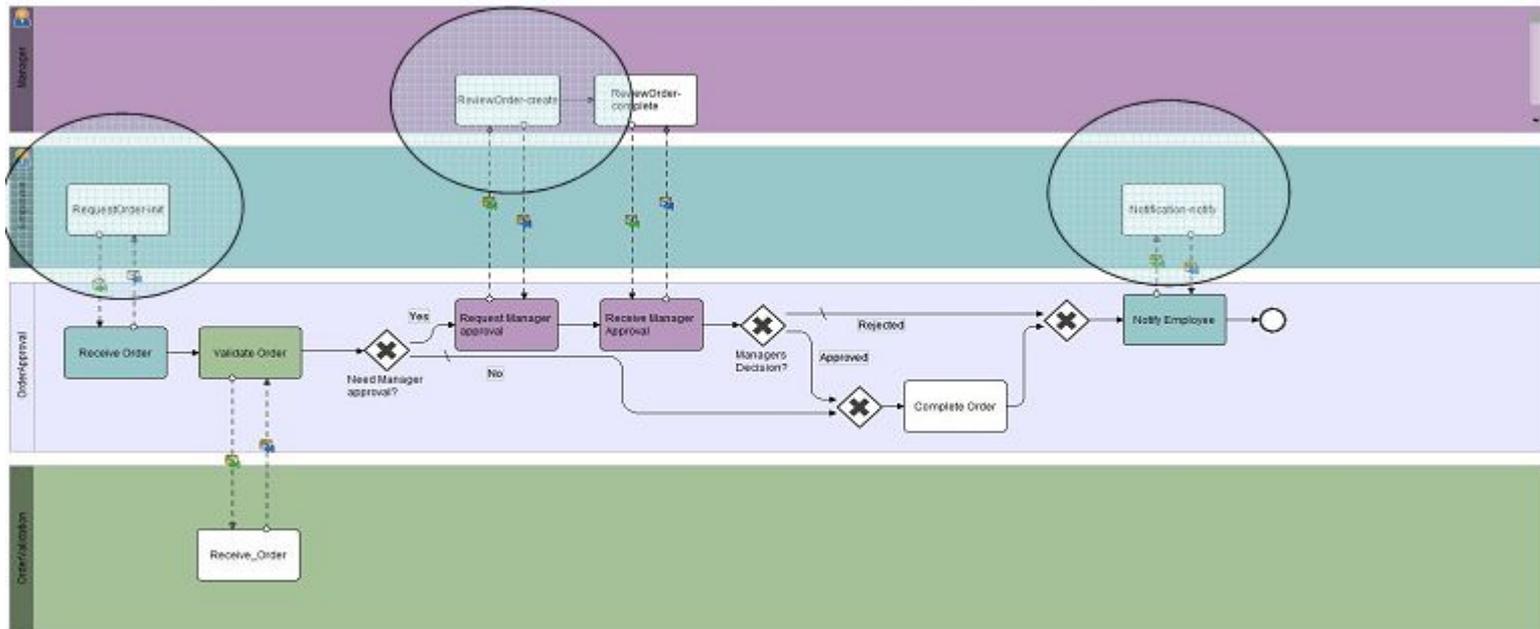
□ Наиболее распространенные способы взаимодействия людей с процессом:

- Уведомление: процесс «уведомляет» человека, но не ожидает какой-либо ответной реакции
- Диалог с человеком: процесс будет ожидать ответа от человека
- Воздействие человека: человек запускает процесс и подает входные данные
- Диалог с процессом: человек обращается к процессу и ожидает ответа



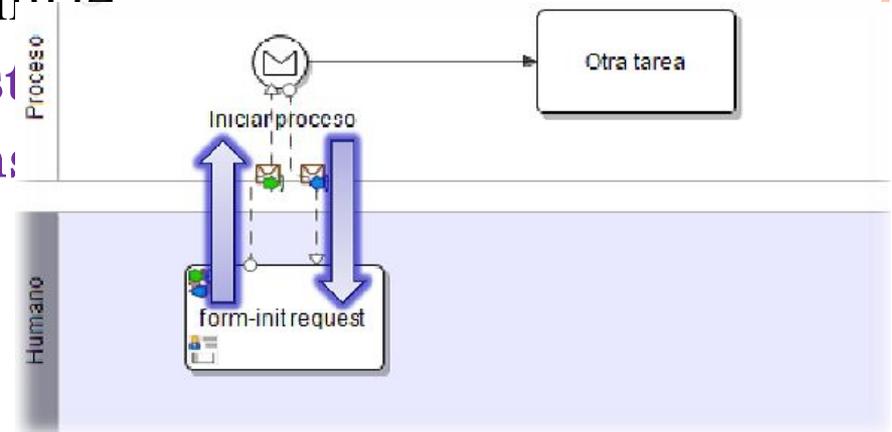
# ФОРМЫ

- Для взаимодействия системы с человеком используются формы
- Форма – это набор управляющих элементов, которые использует человек для ввода информации



# PIPA – СИСТЕМА РУЧНОГО ЗАПУСКА ПРОЦЕССОВ

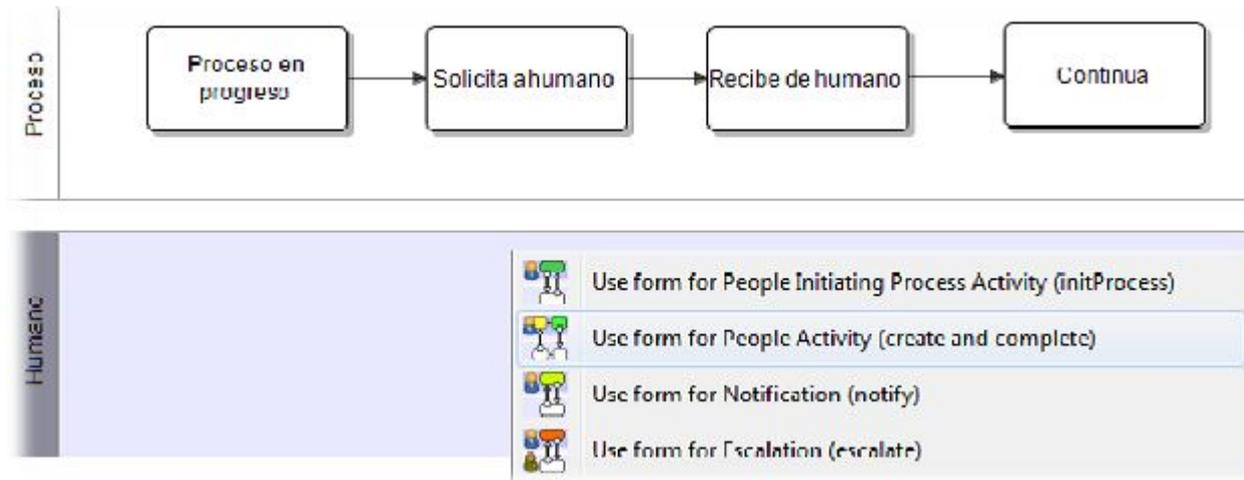
- Для запуска процесса для форм доступна для размещения кнопка «запуск процесса»
- Отправка какой-либо информации из пользовательского интерфейса (формы) запускает новый процесс
- Процесс отправляет пользователю подтверждение того, что он был создан
- При этом создаются переменные:
  - `form1InitProcessRequest`
  - `form1InitProcessResponse`



## ДЕЙСТВИЯ ЛЮДЕЙ

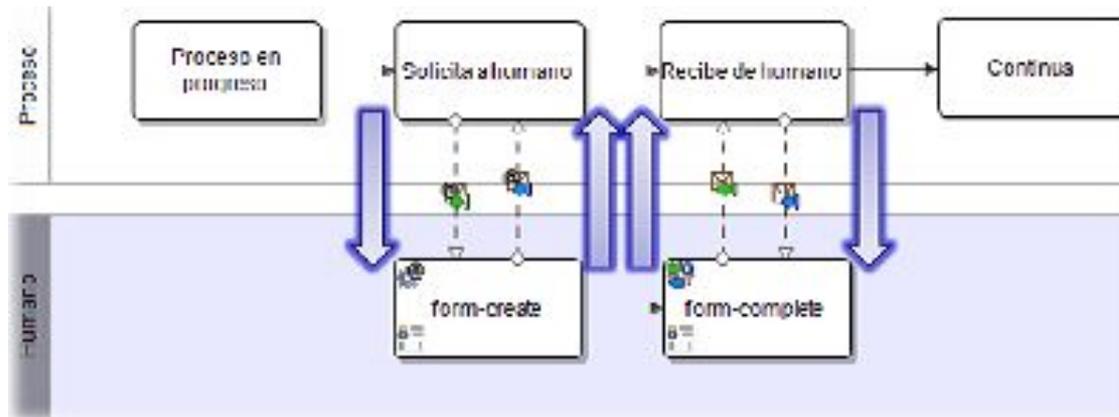
- Процесс создает экземпляр задачи
- Двойной обмен: отправка задачи пользователю и получение ответа
- Человек взаимодействует с пользовательским интерфейсом, в то время как процесс обрабатывает его сообщение

17



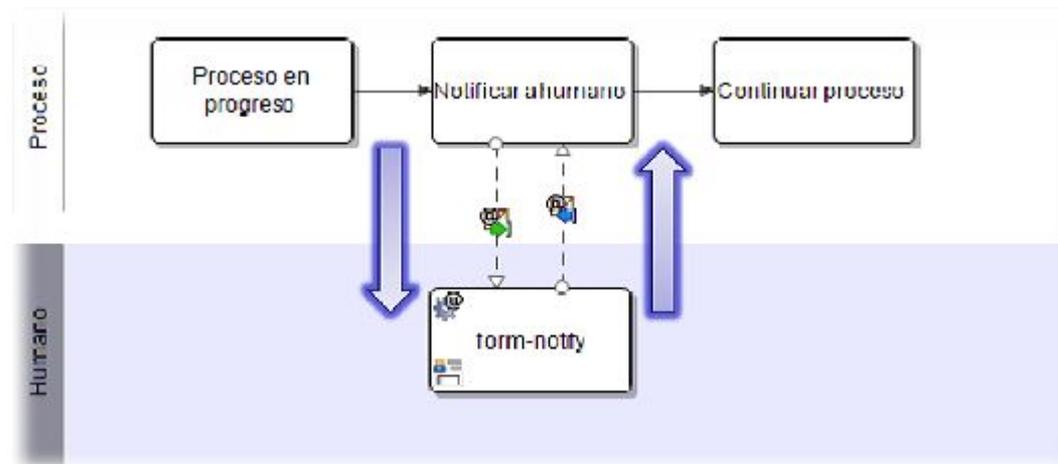
# ДЕЙСТВИЯ ЛЮДЕЙ

- Процесс создает экземпляр задачи
- Двойной обмен: отправка задачи пользователю и получение ответа
- Человек взаимодействует с пользовательским интерфейсом, в то время как процесс обрабатывает его сообщение
- Создаются переменные:
  - `form1CreateTaskRequestMsg`
  - `form1CreateTaskresponseMsg`
  - `form1NotifyTaskCompletionRequestMsg`
  - `form1NotifyTaskCompletionResponseMsg`



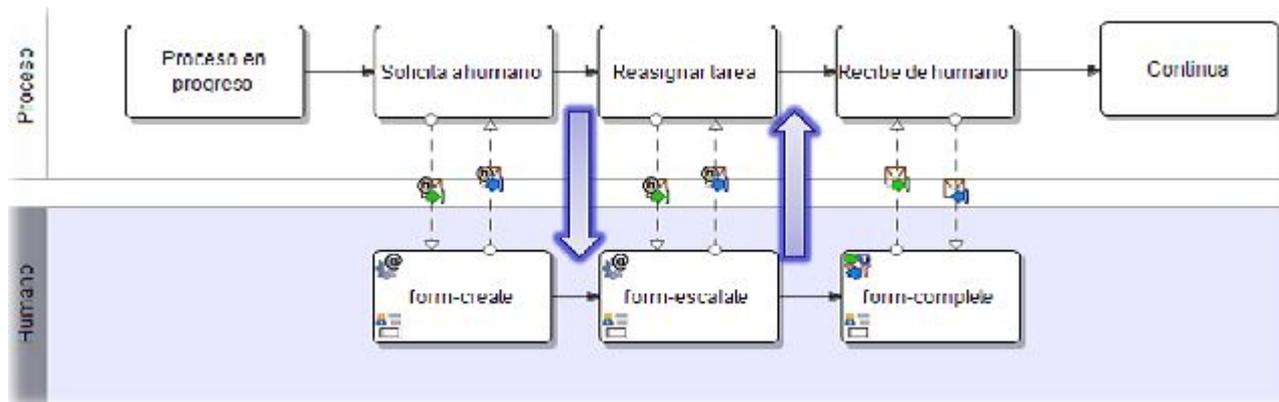
# УВЕДОМЛЕНИЯ

- Используются для отправки уведомляющего сообщения пользователю.
- Процесс не будет ожидать какого-либо действия человека и просто продолжит свое выполнение
- Создаются переменные:
  - **form1NotifyRequestMsg**
  - **form1NotifyResponseMsg**



# ЭСКАЛАЦИЯ

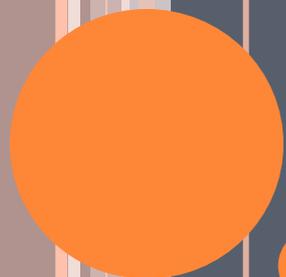
- Перенаправляет задачу другому обработчику или участнику
- Используются в тех случаях, когда необходимо контролировать выполнение задач
- Создаются переменные:
  - **form1EscalateRequestMsg**
  - **form1EscalateResponseMsg**



# INTALIO | WORKFLOW

- Редактор форм: компонент редактора форм **Intalio | Workflow** встроен в пакет **Intalio | Designer**. Он позволяет визуально создавать формы и автоматически генерировать для них код в **Xforms** и **Intalio | AJAX**
- Менеджер связей: это тоже компонент **Intalio | Workflow**, который встроен в **Intalio | Designer**. Позволяет отслеживать создание и изменение файлов **\*.xform** и **\*.gi** и автоматически создает **WSDL** и **XSD** файлы, которые используются для присоединения форм к бизнес-процессу
- **Framework** для интерфейса пользователя: конечный пользователь может взаимодействовать с бизнес процессом используя этот **Framework**





# XML-ФОРМЫ (XFORMS)



22



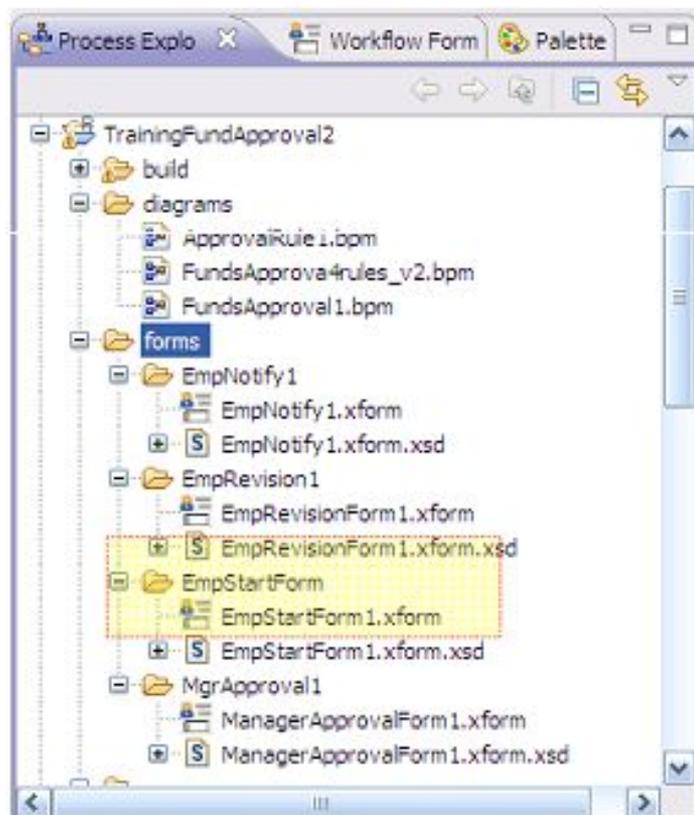
# XFORM'Ы

- Спецификация предоставлена **W3C**
- Новое поколение веб-форм, основанных на языке XML
- Структура формы представлена в виде XML схемы
- XForm'ы определяют механизм связывания между моделью и ее представлением
- Intalio | BPMS Designer включает в себя графический редактор XForm
- Intalio | BPMS Workflow также поддерживают XForm'ы



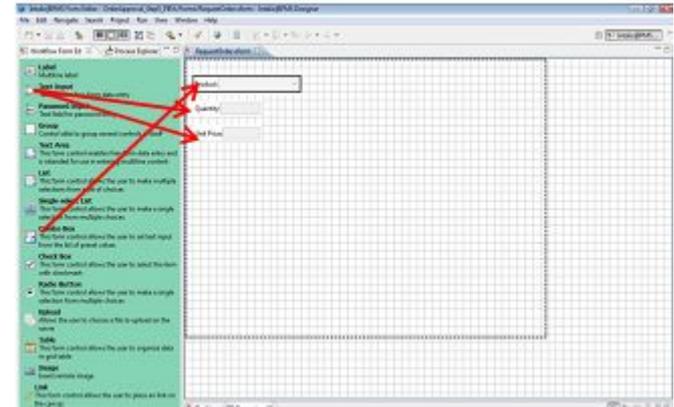
## ФОРМЫ: КОНЦЕПЦИЯ ПАПКОК

- Формы лучше всего создавать в папке
- Лучше всего создавать каждую форму в отдельной папке

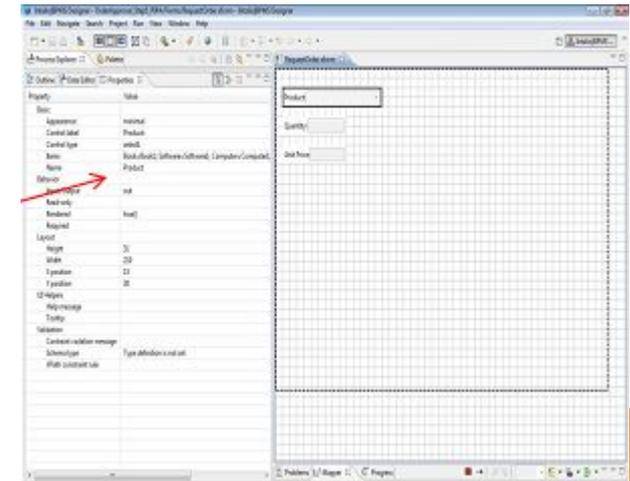


# СОЗДАНИЕ XML-ФОРМ В INTALIO

- Используйте редактор XML-форм (**XFORM Editor**), находящийся на подвижной панели под кнопкой перспективы редактор форм Intalio | BPMS

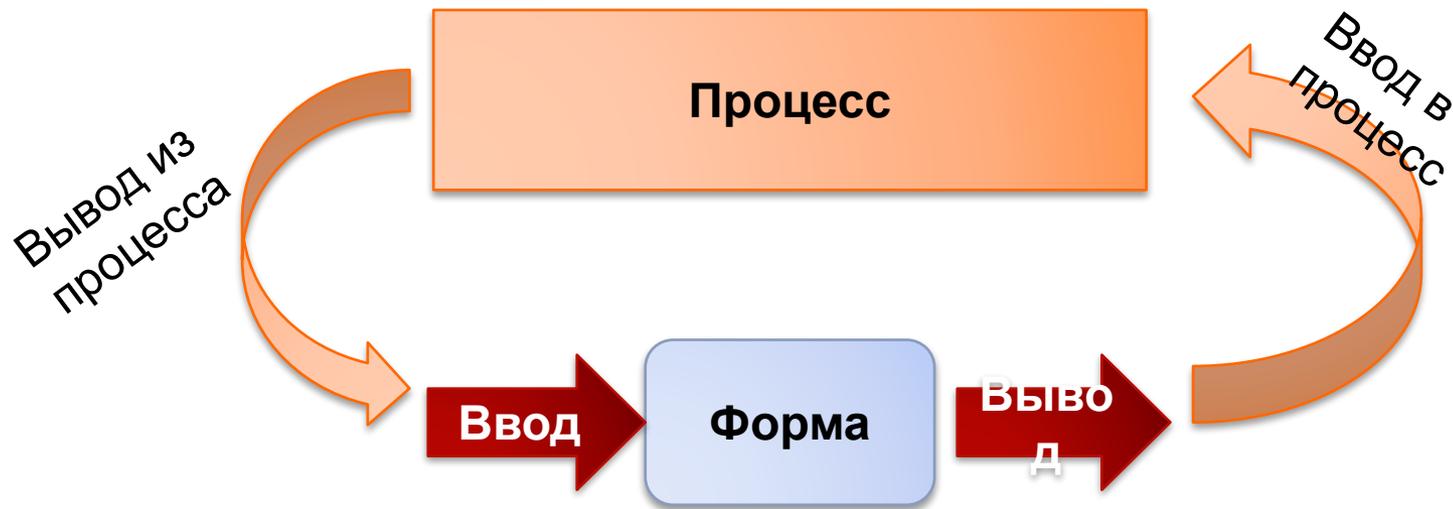


- Каждый элемент формы имеет свои свойства:
  - Вы можете устанавливать метку, имя, параметры ввода/вывода и др. для каждого элемента формы



## Ввод/вывод из форм

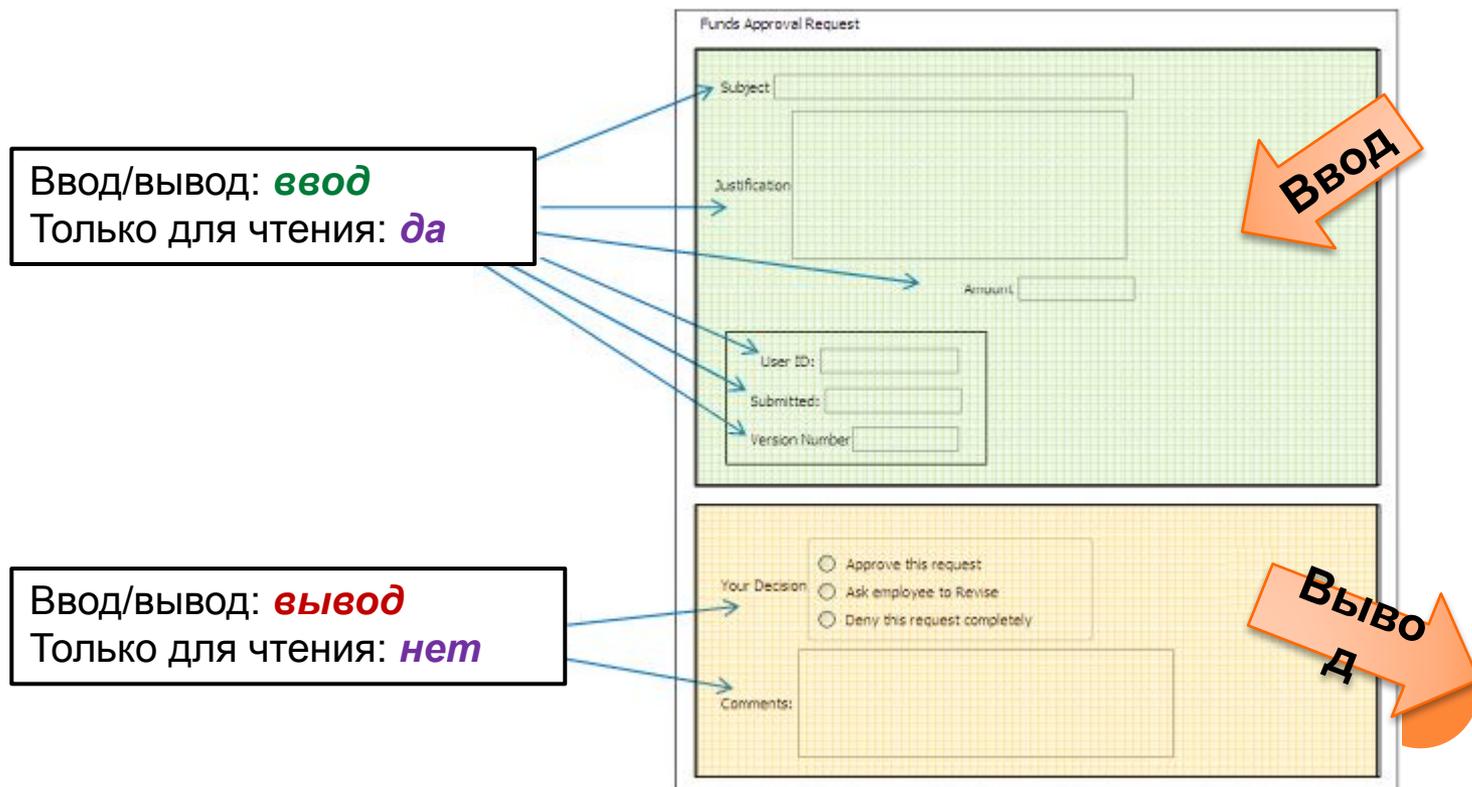
- Ввод в форму используется для заполнения значений полей на форме
- Вывод из формы – для отправки данных процессу

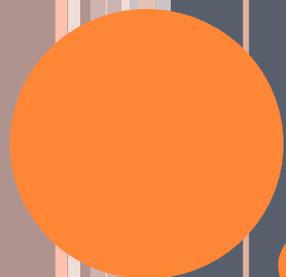


## ПРИМЕР ФОРМЫ для МЕНЕДЖЕРА ПОДТВЕРЖДЕНИЯ ЗАКАЗОВ

□ Некоторым формам необходим ввод, а некоторым – вывод данных

- Ввод данных позволяет просмотреть данные
- Вывод данных позволяет переслать данные от пользователя



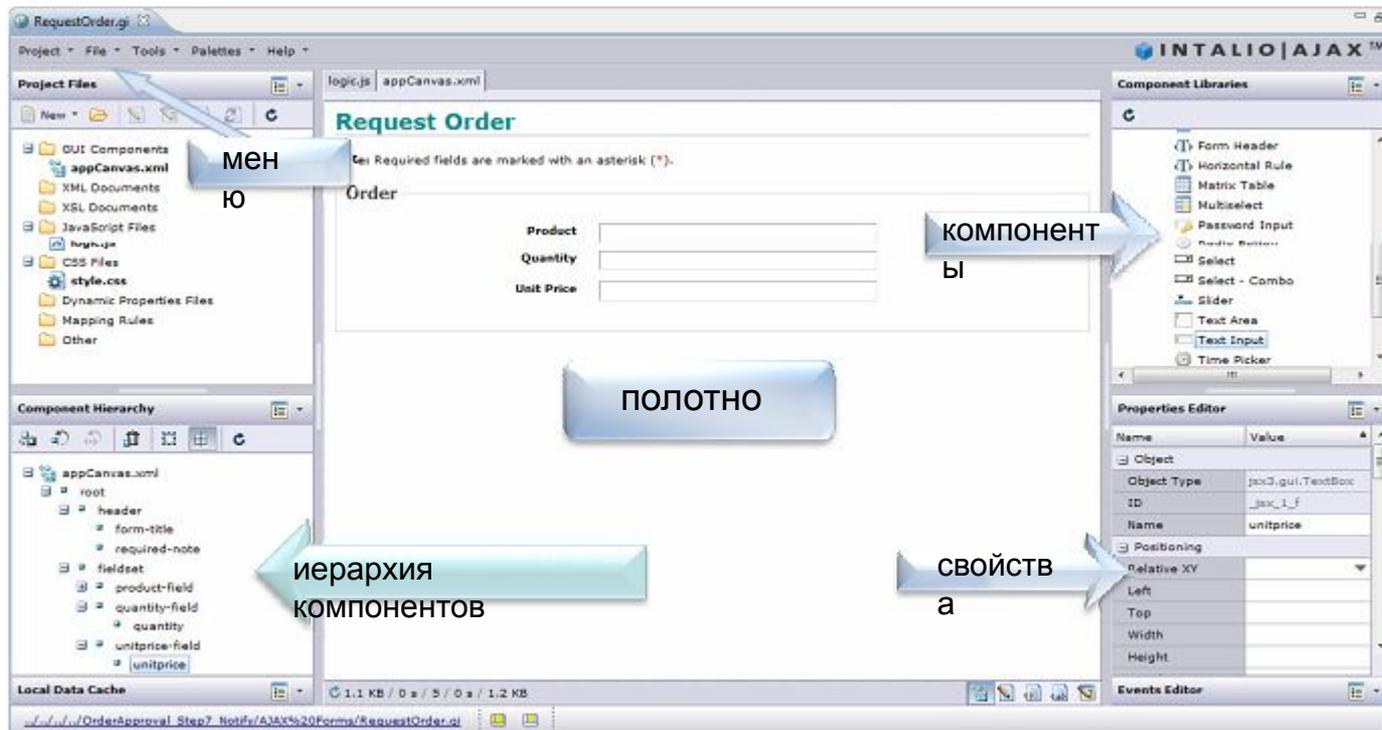


# ФОРМЫ АЈАХ

28

# INTALIO | AJAX

- Intalio | AJAX основан на интерфейсе **TIBCO**
- TIBCO – интегрированная среда разработки, в основе которой лежит JavaScript, что позволяет создавать AJAX -приложения на стороне клиента
- Каждая форма также является приложением. Оно может быть как сложным веб-приложением, так и простым текстовым полем.



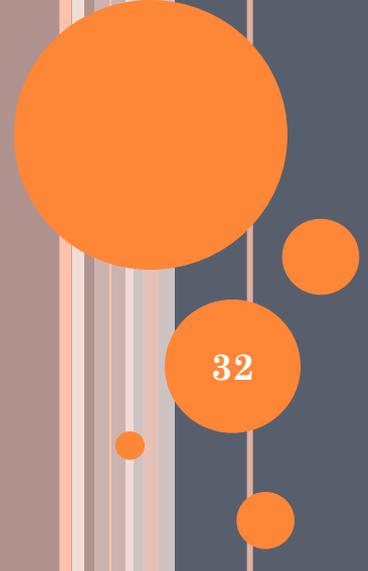
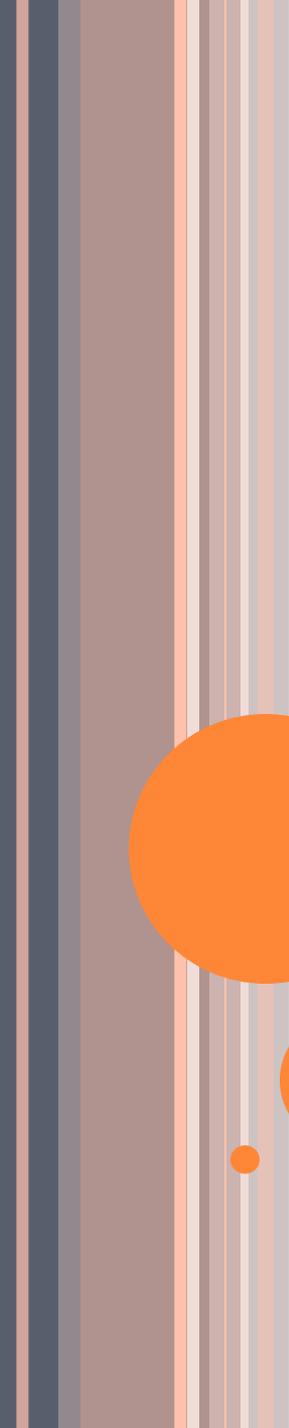
## ФОРМЫ: СОВЕТЫ СПЕЦИАЛИСТОВ

- Следите за названиями
  - Очень важно выбирать говорящие названия в качестве названий элементов
  
- Определяйте тип данных для каждого элемента управления
  - Редактор форм создает модель данных, основываясь на разработанной в редакторе данных форме. Для каждого элемента управления, которому можно задать тип, сопоставляется тип данных в процессе выполнения. Разработка какого-либо кода при этом не требуется.
  
- Не создавайте слишком сложные формы
  - Если в вашей форме много полей и они все не влезят на страницу лучше разбейте задачу на несколько задач



ЕСТЬ ВОПРОСЫ?





# ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС WORKFLOW

32

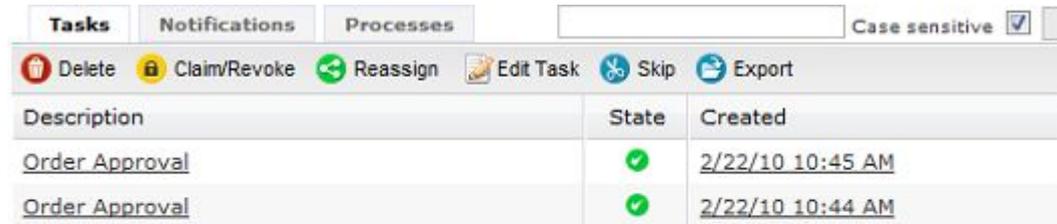
# BP4PEOPLE

- BP4People это предложение от IBM и SAP по расширению BP4WS для автоматизации человеческого труда
- Спецификация была предложена OASIS
- Введение
  - Принцип ручной задачи, которая выполняется человеком
  - Понятие списка задач, который используется чтобы хранить задачи или действия людей
  - Ссылки на людей используются для связывания группы людей с бизнес-процессом



# INTALIO | BPMS WORKFLOW

- Интерфейс позволяет пользователям получать доступ к их задачам
- Еще он позволяет запускать процессы (PIPA)
- Также разрешено получать уведомления



The screenshot shows the 'Tasks' tab in the INTALIO interface. At the top, there are tabs for 'Tasks', 'Notifications', and 'Processes'. Below the tabs is a toolbar with icons for 'Delete', 'Claim/Revoke', 'Reassign', 'Edit Task', 'Skip', and 'Export'. A search bar and a 'Case sensitive' checkbox are also visible. The main content is a table with columns 'Description', 'State', and 'Created'. Two rows are shown, both with the description 'Order Approval', a green checkmark in the 'State' column, and a timestamp in the 'Created' column.

Description	State	Created
<a href="#">Order Approval</a>	✓	2/22/10 10:45 AM
<a href="#">Order Approval</a>	✓	2/22/10 10:44 AM

 INTALIO



The screenshot shows the 'Notifications' tab in the INTALIO interface. It features a toolbar with a 'Delete' icon, a search bar, and a 'Case sensitive' checkbox. The table below has columns 'Description' and 'Created'. Two rows are visible: 'Order Request' and 'Examples - Absence Request', both with a timestamp of 2/22/10 10:40 AM.

Description	Created
<a href="#">Order Request</a>	2/22/10 10:40 AM
<a href="#">Examples - Absence Request</a>	2/22/10 10:40 AM



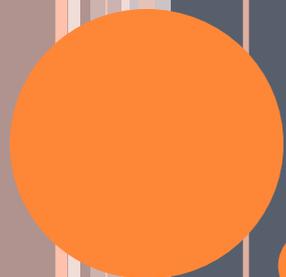
The screenshot shows the 'Order Request' form in the INTALIO interface. The form title is 'Order Request' and includes a note: 'Note: Required fields are marked with an asterisk (\*).'. Under the 'Product Data' section, there are three fields: 'Product' (a dropdown menu with 'Computer' selected), 'Quantity' (a text input field with '123'), and 'Unit Price' (a text input field with '456').



The screenshot shows the 'Notifications' tab in the INTALIO interface. It features a toolbar with a 'Delete' icon, a search bar, and a 'Case sensitive' checkbox. The table below has a 'Description' column. Five rows are visible, all with the description 'Order Status Notification'.

Description
<a href="#">Order Status Notification</a>





# ЦИКЛЫ

35

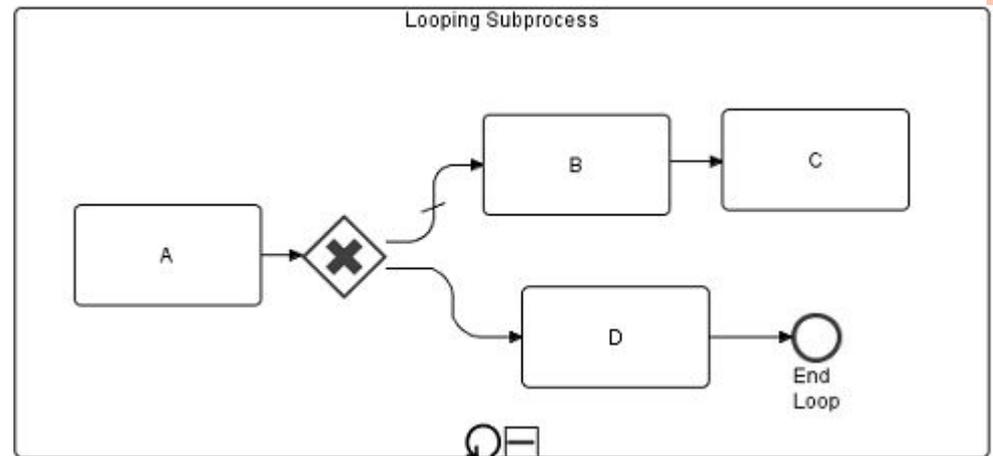
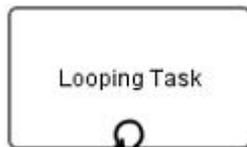
# ЦИКЛЫ

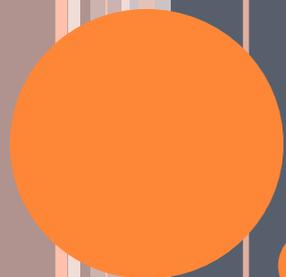
- Что такое циклы?
  - Это периодически повторяющееся действие или группа действий
  
- Технические циклы:
  - Должны проверить каждую строчку в заказе
  - Если резервное копирование не удалось, должны вернуться назад и попробовать снова
  
- Бизнес-циклы:
  - Необходимо подтверждение нескольких уровней доступа, чтобы создать большой и дорогой проект
  - Продолжают отправлять сообщения администратору каждый час до тех пор, пока он не ответит



# VRMN: ЦИКЛЫ

- Действие (в том числе и задача и подпроцесс) могут быть помещены в цикл.
- Существуют три типа циклов:
  - Цикл «ПОКА НЕ»: повторяется до тех пор, пока не выполнит событие выхода из цикла
  - Цикл «ПОКА»: повторяется определенное количество раз
  - Цикл « ДЛЯ КАЖДОГО»: выполняется для каждого экземпляра объекта
- В циклическом под-процессе все задачи и события содержатся в под-процессе который запускается на каждой итерации цикла



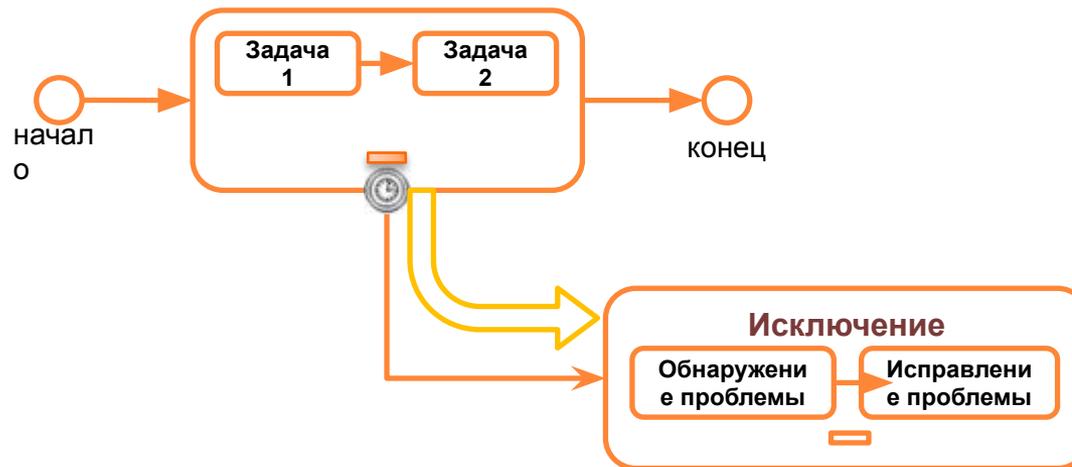


# ИСКЛЮЧЕНИЯ

38

# МАРШРУТ ИСКЛЮЧЕНИЯ

- Автоматически перенаправляет процесс в случае, если действие не завершилось за определенное время

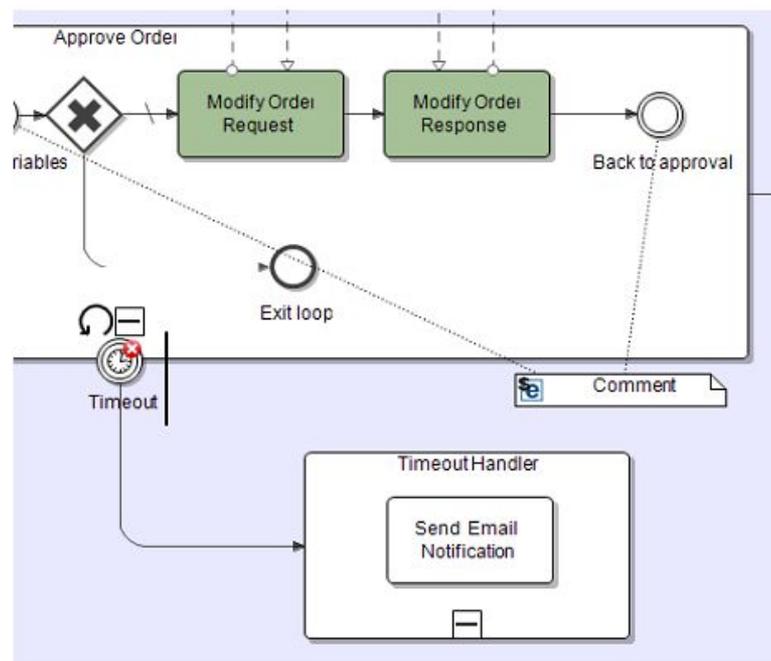


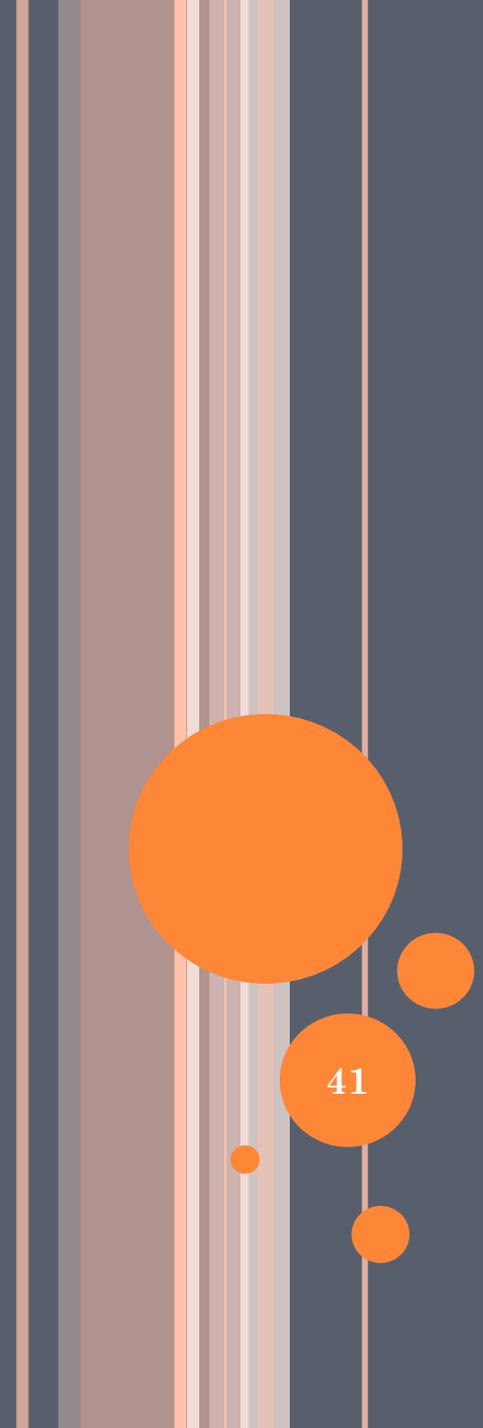
# ИСКЛЮЧЕНИЕ И ОГРАНИЧЕНИЕ ВРЕМЕНИ



## Создадим шаблон исключения

- Вставим промежуточное событие по таймеру (**Intermediate Timer Event**) на границу под-процесса подтверждения заказа (**Approve Order**)
- Добавим действие «отправить уведомление по e-mail» (**Send Email Notification**) в под-процесс **Timeout Handler**



The left side of the slide features a series of vertical stripes in shades of brown, tan, and white. To the right of these stripes are several orange circles of varying sizes, arranged in a cluster. The largest circle is at the top left, with several smaller ones scattered below and to its right.

41

**Спасибо за внимание!**