

# Интеграционное проектирование

## Урок 4

На лекции мы изучим технологии интеграции и освоим основные методы интеграции





# Знакомство и содержание урока



## Игорь Зуев

Руководитель ИТ-проектов

В Международном аэропорту Шереметьево

- 💥 10 лет в ИТ (управление проектами)
- 💥 3 года в преподавание (автор многочисленных обучающих программ по Project & Product Management)
- 💥 Группа компаний “Лукойл”, Проектный офис при Президенте одной из стран СНГ, компании из сферы строительства и нефтегаза
- 💥 ИТ-проекты по автоматизации финансовых систем
- 💥 ИТ-проекты в рамках строительства ГПЗ и магистрального газопровода



## План курса





## Что будет на уроке сегодня

- 📌 Определим роль и место интеграционного проектирования
- 📌 Поговорим о понятии интеграции в системном анализе
- 📌 Рассмотрим основные аспекты системной интеграции
- 📌 Изучим цель, способ, виды интеграции
- 📌 Разберем интеграционный процесс и технологии интеграции
- 📌 Изучим инструменты интеграционного проектирования
- 📌 Подсветим подводные камни системной интеграции





# Роль и место интеграционного проектирования



Вопрос

Какие случаи применения понятия интеграция Вам  
встречались?





Ответ

**Экономическая интеграция** - на уровне государств говорят об интеграции экономик когда речь идет о заключении взаимовыгодных соглашений.

**Культурная интеграция** - представляет собой глобальный процесс углубления национального культурного взаимодействия.





Ответ

**Политическая** интеграция - взаимное сближение политических взглядов различных институтов общества.

**Социальная интеграция** - зачастую рассматривается как принятие индивида группой и его адаптация в этой группе.



Ответ

**Сенсорная интеграция** - в физиологии это способность нашего мозга собирать, обобщать и анализировать информацию, поступающую от разных органов чувств.

**Интеграция данных** - объединение данных полученных от разных источников, их согласование и приведение к общему виду, пригодному для использования конечным потребителем.



Интеграция (от лат. integratio — «восстановление», «восполнение», «соединение») — процесс объединения частей в целое





Дезинтеграция - существенное ослабление взаимосвязей между элементами системы, создающее условия для ее последующего возможного распада.





Вопрос

Знакомы ли вы с понятием интеграция в системном анализе?





Ответ

**Интеграция на прикладном уровне системного анализа** - единый процесс, который объединяет технологии и системы в последовательную цепочку и в результате преобразует форматы данных между системами для достижения конечной цели их функционирования.



Подавляющее большинство  
задач, с которыми приходится  
сталкиваться системному  
аналитику – это проектирование  
интеграционного взаимодействия

## Сложность системной интеграции

Вопрос проектирования системной интеграции можно отнести к сложным в силу ряда факторов



**Системный анализ при объединении систем всегда уникален.**



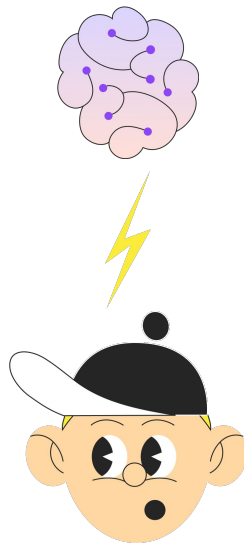
**Системы зачастую различаются технологически, а иногда идеологически.**



**Качество данных принципиальным образом влияет на успех интеграции.**



**Не стоит забывать об ограничениях присущих интеграции, которые связаны с особенностью обработки и передачи отдельных видов сведений.**







# Основные аспекты системной интеграции



Интеграция информационных систем — это процесс установки связей между информационными системами для получения единого информационного пространства



## Первый аспект - Цель интеграции

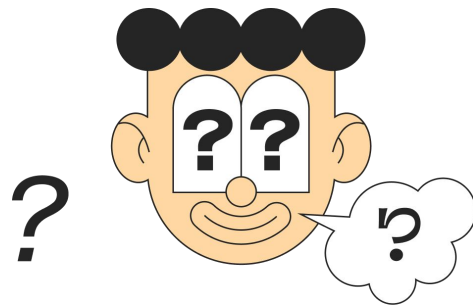


Важнейшая цель интеграции – обеспечить сквозной процесс между системами в соответствии с требованиями заказчика.

Вторичная цель – определить способ и характеристики взаимодействия систем.

Для определения цели интеграции необходимо найти ответ на несколько основных вопросов:

- 💡 **зачем требуется выполнить интеграцию**
- 💡 **кто является заказчиком, полномочным предъявлять требования к интеграции**
- 💡 **какие системы участвуют в интеграции**
- 💡 **сколько стоит интеграция для бизнеса**





Ошибка в цели интеграции ввиду  
невнимания к требованиям  
заказчика, особенностям  
процесса или существующим  
ограничениям может лишить  
интеграцию всякого смысла

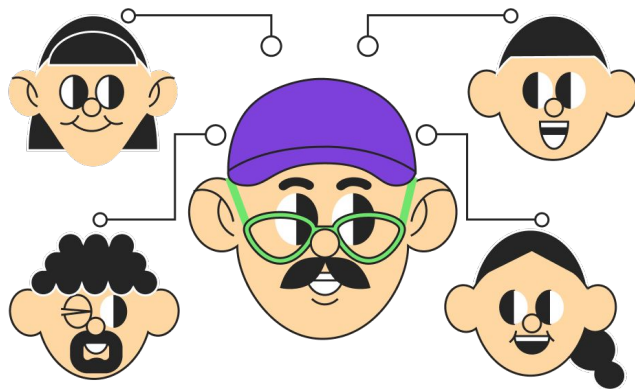
## Второй аспект – способ интеграции

Выделяют несколько основных способов интеграции, которые условно можно разделить на три различных типа:

💡 **Горизонтальная интеграция**

💡 **Вертикальная интеграция**

💡 **Диагональная интеграция**





Горизонтальная интеграция -  
процесс рассматривающий системы  
в качестве равноправных партнеров





В качестве примера  
горизонтальной интеграции  
может служить объединение двух  
систем для достижения общего  
результата



Вертикальная интеграция- процесс предполагающий включение одной системы в качестве подсистемы или компонента в другую







Как пример вертикальной  
интеграции - может выступать  
встраивание компоненты в  
проект.



Диагональная интеграция-  
предполагает наличие ресурсной  
зависимости между системами, при  
их определенной автономности.





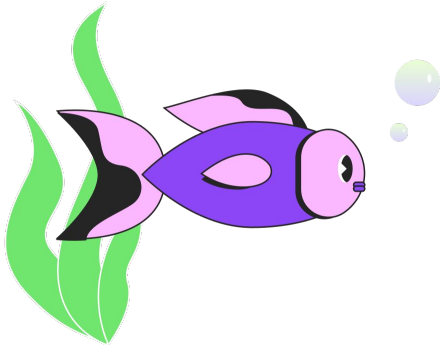
В качестве примера  
диагональной интеграции - когда  
одна система является  
справочной, а вторая использует  
эту информацию для обеспечения  
процесса.

## Третий аспект – виды интеграции

Выделяют несколько основных видов интеграции:

💡 **Внутренняя интеграция**

💡 **Внешняя интеграция**





Вопрос

Чем различаются внешняя и внутренняя интеграция?





Ответ

**Внутренняя интеграция** представляет собой процесс налаживания взаимодействия между отдельными элементами системы



Ответ

**Внешняя интеграция** представляет собой процесс организации взаимодействия с другими системами



Внутренняя интеграция служит  
цели улучшения системы,  
внешняя интеграция направлена  
на качественное развитие  
системы.

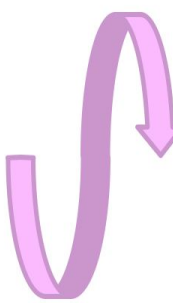


## Четвертый аспект – интеграционный процесс

Интеграция - это деятельность направленная на:

- 💡 **Объединение изначально разрозненных частей системы в одно целое**
- 💡 **Включение нового элемента в уже существующую систему**
- 💡 **Сближение и согласование элементов внутри системы**

На прикладном уровне процесс интеграции предусматривает несколько этапов:

- |   |   |
|---|---|
| 💡 Ознакомление с целью и задачами интеграционного проекта             | 💡 Построение бизнес процессов, бизнес-логики и архитектуры целевого решения (TO BE) |
| 💡 Личное знакомство с командой поддерживающий систему                 | 💡 Описание архитектуры переходного решения (TRANSITIVE)                             |
| 💡 Аудит или документарное знакомство с интегрируемой системой (AS IS) | 💡 Согласование контрактов методов   |
- 

## Четвертый аспект – интеграционный процесс

Дальнейшие шаги процесса относятся к реализации интеграции систем:

- 💡 **Разработка интеграционного решения**
- 💡 **Тестирование интеграционного решения**
- 💡 **Введение в эксплуатацию**
- 💡 **Сопровождение внедренного решения**
- 💡 **Вывод из эксплуатации**



Выполнение всех этапов процесса интеграции переводит систему в новое состояние, обусловленное изменением отдельных элементов системы и их связей.



# Выстраивание эффективного интеграционного процесса – залог успешной интеграции





## Пятый аспект – контракт интеграции

Для информационных систем справедливо утверждение, что основной интеграционный механизм это программные интерфейсы приложений (англ. API – Application Programming Interface)

Интерфейс можно рассматривать как сервисный контракт между двумя приложениями. Этот контракт определяет, как они взаимодействуют друг с другом, используя запросы и ответы. Документация API содержит информацию о том, как разработчики должны структурировать эти запросы и ответы.

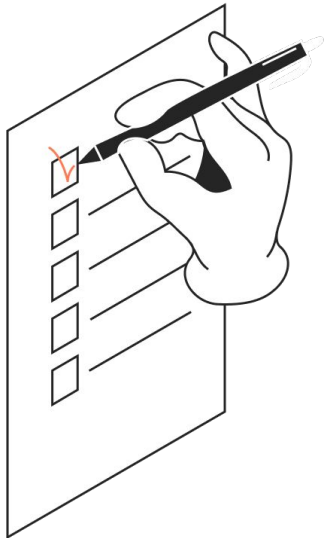
Интерфейсы устанавливают взаимные обязательства между элементами программной системы, что является фундаментом концепции программирования по контракту (англ. design by contract, DbC).

Если абстрагироваться, то интерфейс состоит из двух частей:

-  Имени интерфейса – идентифицирующего интерфейс
-  Метода интерфейса – идентифицирующего способ использования интерфейса

## Пятый аспект – контракт интеграции

**Для предсказуемого поведения интерфейса он покрывается взаимными обязательствами с потребителями - контрактом метода:**



- 💡 Описание операции, доступной для выполнения
- 💡 Возможные входные данные и их типы и значения
- 💡 Данные и их типы, которые являются результатом работы функции
- 💡 Условия возникновения исключений, их типы и значения
- 💡 Наличие побочного эффекта
- 💡 Предусловия
- 💡 Постусловия
- 💡 Гарантии производительности





Программный сервис как бы говорит  
«Ко мне можно обращаться так, а я  
обязуюсь делать то» - это и есть  
контракт воплощенный в API



Приложение, отправляющее запрос,  
называется **клиентом**, а приложение,  
отправляющее ответ, называется  
**сервером**.



## Шестой аспект – технология интеграции

Существует четыре различных способа работы API в зависимости от того, когда и почему они были созданы:

- 💡 **SOAP API** - SOAP – Simple Object Access Protocol, т. е. простой протокол доступа к объектам. Клиент и сервер обмениваются сообщениями посредством XML.
- 💡 **RPC API** - система удаленного вызова процедур. Клиент выполняет функцию (или процедуру) на сервере, и сервер отправляет результат обратно клиенту.
- 💡 **Websocket API** – это еще одна реализация концепции API, которая использует объекты JSON для передачи данных. WebSocket API поддерживает двустороннюю связь между клиентскими приложениями и сервером.
- 💡 **REST API** - На сегодняшний день это самые популярные и гибкие API-интерфейсы в Интернете. REST — это аббревиатура от Representational State Transfer («передача состояния представления»).





Первым использовавшим термин REST в 2000 году стал американский учёный Рой Филдинг в своей докторской диссертации «Архитектурные стили и дизайн программных сетевых архитектур».



## Шестой аспект – технология интеграции

REST требует, чтобы клиент сделал запрос на сервер для получения или изменения данных на сервере. Запрос обычно состоит из:

- 💡 HTTP-метода, который определяет вид операции;
- 💡 заголовка, который позволяет клиенту передавать информацию о запросе;
- 💡 пути к ресурсу;
- 💡 необязательного тела сообщения, содержащего данные.



## Шестой аспект – технология интеграции

Существует 4 основных метода HTTP, которые мы используем в запросах для взаимодействия с ресурсами в системе REST:

- 💡 GET — получение конкретного ресурса (по id) или коллекцию ресурсов
- 💡 POST — создание нового ресурса
- 💡 PUT — обновление конкретного ресурса (по id)
- 💡 DELETE — удаление конкретного ресурса по id



## Шестой аспект – технология интеграции

Существует 4 основных метода HTTP, которые мы используем в запросах для взаимодействия с ресурсами в системе REST:

- 💡 GET — получение конкретного ресурса (по id) или коллекцию ресурсов
- 💡 POST — создание нового ресурса
- 💡 PUT — обновление конкретного ресурса (по id)
- 💡 DELETE — удаление конкретного ресурса по id



Например:

PUT: /ticket – для создания билета

GET: /ticket/{uuid} – для получения  
информации по билету

DELETE: /ticket – для удаления билета и  
отката операции покупки



## Шестой аспект – технология интеграции

API классифицируются как по архитектуре, так и по сфере применения. Мы уже рассмотрели основные типы архитектур API, поэтому мы предлагаем рассмотреть сферы применения:

- 💡 Частные API - внутренние API организаций
- 💡 Общедоступные API – с общим доступом и могут быть использованы кем угодно
- 💡 Партнерские API - доступные только авторизованным сторонним разработчикам для содействия партнерским отношениям между предприятиями
- 💡 Составные API - объединяют два или более разных API для решения сложных системных требований или поведения



API помогают договариваться в  
миллионах приложений во всем мире  
и служат интеграционным стандартом  
при проектировании  
информационных систем



# Инструменты проектирования интеграционного взаимодействия





## Проектирование программных интерфейсов

В качестве основного инструмента для проектирования программных интерфейсов выступает Swagger. Изначально он представлял Open Source framework от компании SmartBear под названием Swagger, а с 2016 года он получил новое обновление и стал называться OpenAPI Specification.

Swagger - это инструментарий(фреймворк) для спецификации RESTful API.

Существует два похода к использованию Swagger:

- Сначала документация
- Сначала программный код





Основное преимущество Swagger перед остальными технологиями, это получение в результате описание программного интерфейса приложения по стандарту OpenAPI



Open API Specification - это язык спецификаций HTTP API, который служит основой для реализации открытых API

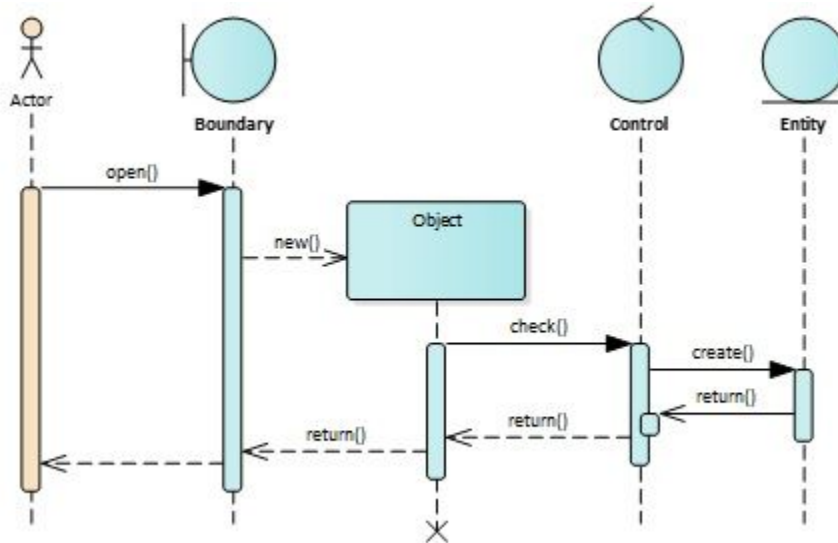




Открытые API – это одним из  
связующих элементов цифровой  
инфраструктуры финансового рынка  
и производственных систем

## Проектирование взаимодействий

В качестве инструмента проектирования взаимодействий выступает стандарт UML Sequence Diagram.



Объекты расположены в горизонтальной последовательности, между ними передаются сообщения. Ось времени ориентирована сверху вниз.

Каждый объект имеет пунктирную линию, называемую "линией жизни", где этот элемент существует и потенциально принимает участие во взаимодействиях.



# Принципы интеграционного проектирования



## Хорошо спроектированная интеграция систем должна обеспечивать:

- 💡 Согласованность контрактов и их неизменность, что означает что две системы должны безусловно понимать друг друга в любых ситуациях
- 💡 Обязательность ограничений, предписывающая устанавливать ограничения на обмен сообщениями при которых обработка происходит стабильно,
- 💡 Стабильное качество информации, что достигается за счет стандартизации данных и реализации форматно-логического контроля на основных этапах обработки данных,
- 💡 Наличие плана Б – что предполагает решения вопроса обеспечения надежности функционирования системы при отказе её компонентов.



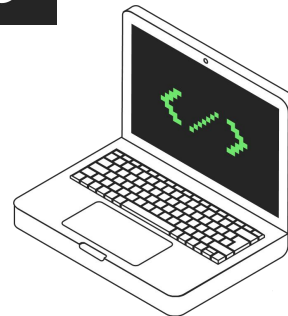
# Подводные камни системной интеграции



## Системная интеграция несет в себе существенные риски:

💡 Каждая новая интеграция требует уникального решения адекватного цели

💡 Новые интеграции могут приводить к снижению стабильности и надежности функционирования системы





Шаблоны отказоустойчивости - это набор установленных механизмов, позволяющих приложениям эффективно справляться с отказами.





Практическая реализация шаблонов  
может дать существенные  
преимущества при реализации  
сложной интегрированной системы



## В заключение

Мы рассмотрели основные вопросы интеграционного проектирования и познакомились с принципами и инструментами интеграционного проектирования, научились проектировать взаимодействия и оценивать их эффективность.

Ни одна современная система немыслима без интеграционных взаимодействий, поэтому изученная тема является исключительно важной для построения эффективных надежных систем несущих ценность для их потребителей.

