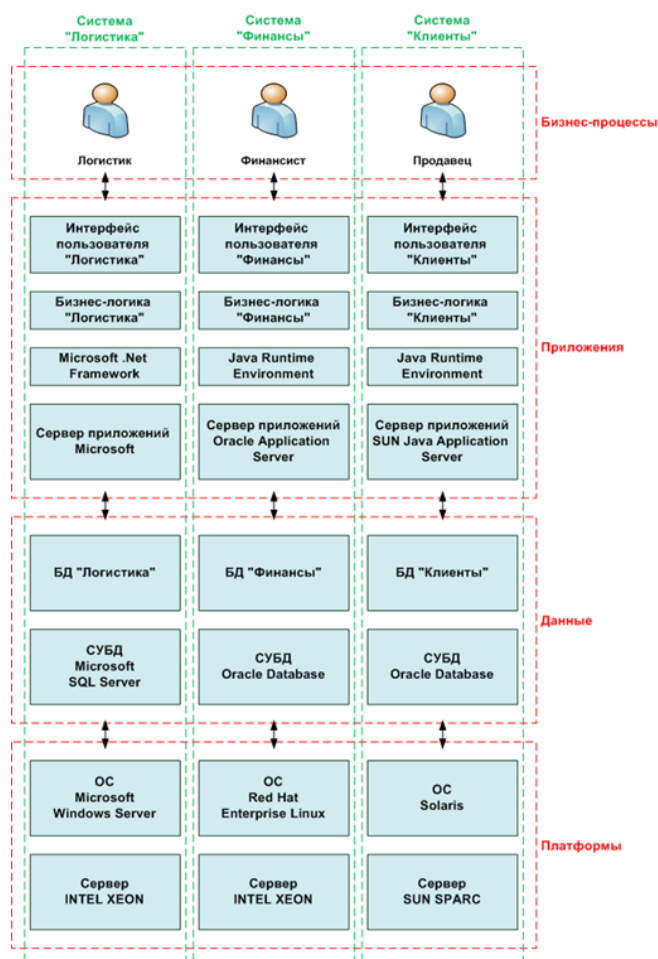




Integration_types

Интеграция информационных систем заключается в интеграции одного или нескольких компонентов интегрируемых информационных систем (объектов интеграции):

- Интеграция платформ;
- Интеграция данных;
- Интеграция приложений;
- Интеграция бизнес-процессов.



Интеграция платформ:

- Обеспечение возможности взаимодействия между приложениями, работающими на различных программно-аппаратных платформах (например, между приложениями, работающими на серверах Windows, Solaris, Linux и др.);
- Обеспечение возможности работы приложений, разработанных для одной программно аппаратной платформы, на других программно-аппаратных платформах (например, приложений Windows на платформах Linux, Solaris и др.);
- **Удаленный вызов процедур** (RPC, Web-сервисы, REST и пр). Позволяет передачу входящих параметров и получения выходных результатов, для приложений, работающих на других платформах;
- **ПО промежуточного слоя** (Microsoft.Net, Java Runtime). Приложения, разработанные для этой технологии работают на любых программно-аппаратных платформах (Windows, Linux и др.) без каких-либо доработок самих приложений;
- **Виртуализация** (Позволяет существенно упростить использования различных платформ. Виртуализация позволяет развернуть и одновременно использовать на одном физическом сервере несколько операционных систем: Windows, Linux и др. На каждом из таких «виртуальных» серверов могут быть развернуты соответствующие системы, которые будут доступны одновременно.).

Интеграция данных:

Интеграция на уровне данных предполагает совместное использования данных различных систем. Существует два подхода к интеграции:

- **Универсальный доступ к данным.** Технологии универсального доступа к данным позволяют обеспечить единообразный доступ к данным различных СУБД. Посредником для работы с конкретной СУБД в данном случае является драйвер для соответствующей СУБД. Наиболее распространенные технологии этого класса: ODBC, JDBC, ADO.NET и ORM (Entity Framework, Hibernate, NHibernate, Flexberry ORM);
- **Хранилища данных.** Хранилище данных - база данных, хранящая в себе данные, собираемые из баз данных различных информационных систем, для целей их дальнейшего анализа. Для создания хранилищ данных используются технологии OLAP.

Интеграция приложений:

Интеграция на уровне приложений подразумевает использование готовых функций приложений другими приложениями.

Подходы к интеграции приложений:

- **Интерфейсы прикладного программирования.** Программный интерфейс конкретной системы представляет собой «опубликованный» функционал этой системы, который может быть использован извне;
- **Обмен сообщениями (Корпоративная сервисная шина).** Интеграция нескольких систем заключается в передаче информации между ними (запрос, ответ). При этом может производиться шифрование сообщений;
- **Сервис-ориентированная архитектура (SOA).** Небольшая программная надстройка над функционалом приложения, преобразующая вызовы, получаемые через Веб, во внутренние вызовы функций приложения и возвращающая результаты обратно:
 - Публикация функционала корпоративных приложений в виде Веб-сервисов. Упорядочивание опубликованных сервисов в виде каталога;
 - Построение на основе Веб-сервисов новых приложений путем их комбинации.
- **Интеграция пользовательских интерфейсов.** Часто используется для создания приложений «одного окна». Внутри каждого фрейма содержится отдельное Веб-приложение. Благодаря фреймам, все эти приложения отображаются на экране одновременно.

Интеграция бизнес процессов:

В рамках интеграции бизнес-процессов происходит и интеграция приложений, и интеграция данных и, что не менее важно, людей, вовлеченных в этот бизнес-процесс.

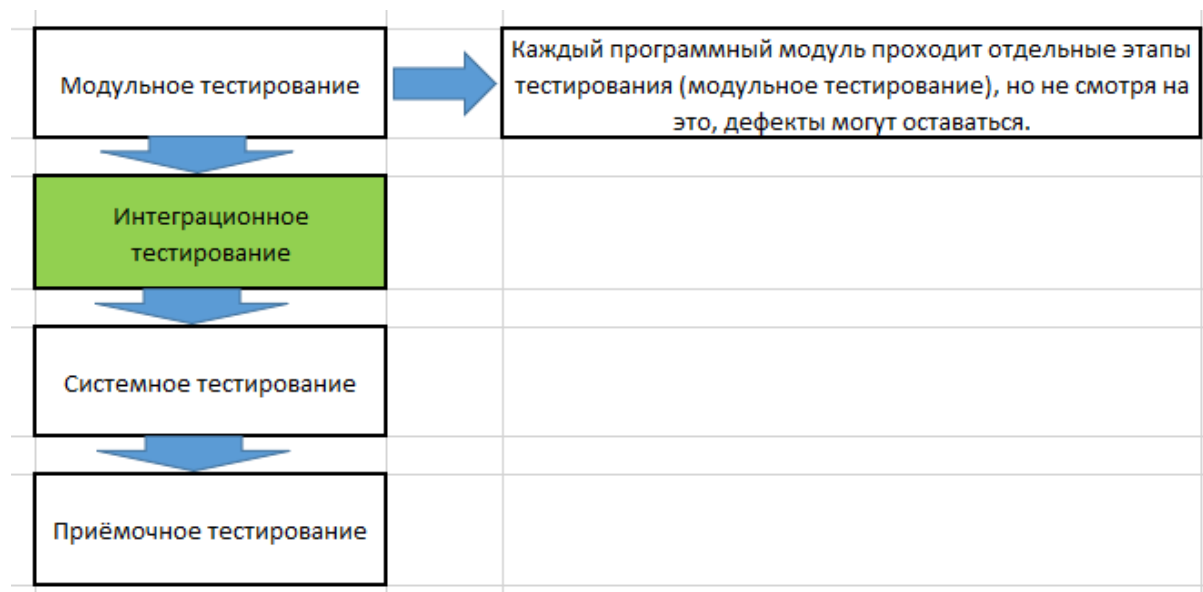
Идеи, для интеграции бизнес-процессов:

- Составить сценарий некоторого бизнес-процесса, происходящего в организации, описать в нем операции взаимодействия пользователей с различными системами и систем между собой. Сценарий создается при помощи специализированного программного продукта, который далее будет управлять ходом этого бизнес-процесса согласно сценарию;

- Операции взаимодействия с системами в рамках бизнес-процесса детально описываются в терминах информационного обмена: форматы обмена, используемые сервисы, приложения, события, правила, политики и т.п;
- К интегрирующему программному обеспечению, при помощи которого описан сценарий бизнес-процесса, подключаются посредством адаптеров интегрируемые системы, вовлеченные в бизнес-процесс. Таким образом, становится возможным автоматизированный информационный обмен между системами;
- Готовый к выполнению бизнес-процесс выводится на «пульт управления» менеджера, при помощи которого, он может запускать и останавливать бизнес-процессы, отслеживать их состояние, вводить данные и принимать решения на отдельных операциях бизнес-процессов, требующих участия человека и др. Взаимодействия между системами, не требующее участия человека осуществляется автоматически интегрирующим ПО.

Особенности тестирования интеграций:

Интеграционное тестирование - это тип тестирования, при котором программные модули объединяются логически и тестируются как группа. Как правило, программный продукт состоит из нескольких программных модулей, написанных разными программистами. Целью нашего тестирования является выявление багов при взаимодействии между этими программными модулями и в первую очередь направлен на проверку обмена данными между этими самими модулями.



Интеграционное тестирование отличается от других видов тестирования тем, что он сосредоточен в основном на интерфейсах и потоке данных (между модулями). Здесь приоритет проверки присваивается интегрирующим ссылкам, а не функциям блока, которые уже проверены.

Стратегии интеграционного тестирования:

- Подход Большого взрыва.
- Инкрементальный подход:
 - Нисходящий подход (сверху вниз)
 - Подход «снизу вверх»
 - Сэндвич – комбинация «сверху вниз» и «снизу вверх»

Подход большого взрыва - все элементы собираются вместе и тестируются.

Инкрементальный подход - В данном подходе тестирование выполняется путем объединения двух или более логически связанных модулей. Затем добавляются другие связанные модули и проверяются на правильность функционирования. Процесс продолжается до тех пор, пока все модули не будут соединены и успешно протестированы.

Может осуществляться сверху вниз, либо снизу вверх.

Заглушка и драйвер - Инкрементальный подход осуществляется с помощью фиктивных программ, называемых заглушками и драйверами. Заглушки и драйверы не реализуют всю логику программного модуля, а только моделируют обмен данными с вызывающим модулем.

Интеграция “Снизу вверх” - В восходящей стратегии каждый модуль на более низких уровнях тестируется с модулями более высоких уровней, пока не будут протестированы все модули. Требуется помощь драйверов для тестирования. невозможно реализовать ранний прототип и критические модули, управляющие потоком, тестируются последними и могут быть подвержены ошибкам.

Интеграция “Сверху вниз” - При подходе «сверху вниз» тестирование, что логично, выполняется сверху вниз, следуя потоку управления программной системы. Используется много заглушек для тестирования.

“Сэндвич” - Эта стратегия представляет собой комбинацию подходов «сверху вниз» и «снизу вверх». Здесь верхнеуровневые модули тестируются с

нижнеуровневыми, а нижеуровневые модули интегрируются с вышеуровневыми, соответственно, и тестируются.

Алгоритм интеграционного тестирования:

- Подготовка план интеграционных тестов;
- Разработка тестовых сценариев;
- Выполнение тестовых сценариев и фиксирование багов;
- Отслеживание и повторное тестирование дефектов;
- Повторять шаги 3 и 4 до успешного завершения интеграции.

Критерии старта интеграционного тестирования:

- Модули и модульные компоненты;
- Все ошибки с высоким приоритетом исправлены и закрыты;
- Все модули должны быть заполнены и успешно интегрированы;
- Наличие плана Интеграционного тестирования, тестовый набор, сценарии, которые должны быть задокументированы;
- Наличие необходимой тестовой среды.

Критерии окончания интеграционного тестирования:

- Успешное тестирование интегрированного приложения;
- Выполненные тестовые случаи задокументированы;
- Все ошибки с высоким приоритетом исправлены и закрыты;
- Технические документы должны быть представлены после выпуска Примечания.