

Web-architecture

«Web-architecture»

№	Уровни архитектуры веб-приложений	Описание
1	Уровень представления (PL)	PL отображает пользовательский интерфейс и упрощает взаимодействие с пользователем. Уровень представления имеет компоненты пользовательского интерфейса, которые визуализируют и показывают данные для пользователей. Также существуют компоненты пользовательского процесса, которые задают взаимодействие с пользователем. PL предоставляет всю необходимую информацию клиентской стороне. Основная цель уровня представления - получить входные данные, обработать запросы пользователей, отправить их в службу данных и показать результаты.
2	Уровень обслуживания данных (DSL)	DSL передает данные, обработанные уровнем бизнес-логики, на уровень представления. Этот уровень гарантирует безопасность данных, изолируя бизнес-логику со стороны клиента.
3	Уровень бизнес-логики (BLL)	BLL несет ответственность за надлежащий обмен данными. Этот уровень определяет логику бизнес-операций и правил. Вход на сайт - это пример уровня бизнес-логики.
4	Уровень доступа к данным (DAL)	DAL предлагает упрощенный доступ к данным, хранящимся в постоянных хранилищах, таких как двоичные файлы и файлы XML. Уровень доступа к данным также управляет операциями CRUD - создание, чтение, обновление, удаление.

Особенности тестирования монолитных и микросервисных веб-приложений

1. Технологические особенности

Классическое приложение работает с использованием одной или семейства родственных технологий.

Web-приложение работает с использованием принципиально различных технологий.

2. Структурные особенности

Классическое приложение “монолитное”. Состоит из одного или небольшого количества модулей. Не использует серверы БД, web-серверы и т.д.

Web-приложение — “многокомпонентное”. Состоит из большого числа модулей. Обязательно использует серверы БД, web-серверы, серверы приложений.

3. Особенности режимов работы

Классическое приложение работает в режиме реального времени, т.е. известно о действиях пользователя сразу же, как только оно выполнено.

Web-приложение работает в режиме “запрос-ответ”, т.е. известно о некотором наборе действий только после запроса на сервер. Особенности тестирования web-приложений, режим работы

4. Особенности формирования интерфейса

Классическое приложение использует для формирования интерфейса пользователя относительно устоявшиеся и стандартизированные технологии.

Web-приложение использует для формирования пользовательского интерфейса стремительно развивающиеся технологии, множество которых конкурирует между собой.

5. Особенности работы с сетью

Классическое приложение практически не использует сетевые каналы передачи данных.

Web-приложение активно использует сетевые каналы передачи данных.

6. Особенности запуска и остановки

Классическое приложение запускается и останавливается редко.

Web-приложение запускается и останавливается по факту поступления каждого запроса, т.е. очень часто.

7. Количество пользователей

Классическое приложение: количество пользователей, одновременно использующих приложение, подвержено контролю, ограничено и легко прогнозируемо.

Web-приложение: количество пользователей, одновременно использующих приложение, сложнопрогнозируемо и может скачкообразно меняться в широких

диапазонах.

8. Особенности сбоев и отказов

Классическое приложение: выход из строя тех или иных компонентов сразу становится очевидным.

Web-приложение: выход из строя некоторых компонентов оказывает непредсказуемое влияние на работоспособность приложения в целом.

9. Особенности в инсталляции

Классическое приложение — процесс инсталляции стандартизирован и максимально ориентирован на широкую аудиторию пользователей. Не требует специфических знаний. Добавление компонентов приложения выполняется стандартным способом с использованием одного и того же инсталлятора.

Web-приложение — процесс инсталляции часто недоступен конечному пользователю. Инсталляция требует специфических знаний. Процесс изменения компонент приложения не предусматривается или требует квалификации пользователей. инсталлятор отсутствует.

10. Особенности в деинсталляции

Классическое приложение: процесс деинсталляции стандартизирован и выполняется автоматически или полуавтоматически.

Web-приложение: процесс деинсталляции требует специфических знаний для вмешательства администратора и часто сопряжен с изменением кода среды функционирования приложения, БД, настройки системного ОС.

11. Особенности среды функционирования

Классическое приложение: среда функционирования стандартизирована и не сильно влияет на функционирование приложения.

Web-приложение: среда функционирования очень разнообразна и может оказать серьезное влияние на работоспособность и серверной, и клиентской части.