**Архитектура ReactJS**

**1. Инициализация проекта и структура**

Инициализация проекта в React происходит с использованием инструмента, такого как Create React App (CRA). Этот инструмент упрощает создание и настройку проекта. Пример типичной структуры проекта:

**my-react-app/**

**|-- node\_modules/**

**|-- public/**

**| |-- index.html**

**|-- src/**

**| |-- components/**

**| | |-- Header.js**

**| | |-- Footer.js**

**| |-- pages/**

**| | |-- Home.js**

**| | |-- About.js**

**| |-- App.js**

**|-- package.json**

**|-- README.md**

**node\_modules/:** Здесь хранятся зависимости проекта.

**public/:** Содержит основной HTML-файл и другие статические ресурсы.

**src/:** Основная директория для исходного кода.

**components/:** Компоненты React.

**pages/:** Компоненты, представляющие страницы приложения.

**App.js:** Основной компонент приложения.

**2. Компоненты и их создание**

Компоненты в React - это многоразовые блоки строительных элементов. Пример создания функционального компонента приложен в папке **KT** на Github.

**3. Стилизация компонентов и модульный подход**

Для стилизации компонентов можно использовать CSS, SCSS или любой другой препроцессор. Модульный подход позволяет изолировать стили компонента. Пример стилизации также приложен в папке **KT** на Github.

**4. Роутинг и архитектура приложения**

Для роутинга часто используется библиотека React Router. Пример архитектуры приложен в папке **KT** на Github, а также пример использования React Router.

**5. Хуки и примеры**

Хуки в React предоставляют возможность использовать состояние и другие возможности в функциональных компонентах. Пример с хуками приложен в папке **KT** на Github.

**6. Gitflow**

Gitflow - это методология работы с Git, предлагающая строгий набор правил для организации репозитория.

**Алгоритм:**

1. **Инициализация проекта:**

Git init

1. **Настройка Gitflow:**

Git flow init

1. **Создание и переключение на новую фичу:**

Git flow feature start my-feature

1. **Завершение фичи:**

Git flow feature finish my-feature

1. **Подготовка релиза:**

Git flow release start 1.0.0

1. **Завершение релиза:**

Git flow release finish 1.0.0

1. **Публикация изменений:**

Git push origin master

Git push origin develop

Git push --tags