

# Разработка решений в Creatio

Разработка приложений на платформе Creatio

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

# Содержание

<b>Разработка приложений на платформе Creatio</b>	<b>4</b>
Уровни кастомизации Creatio	4
Инструменты разработки приложений	6

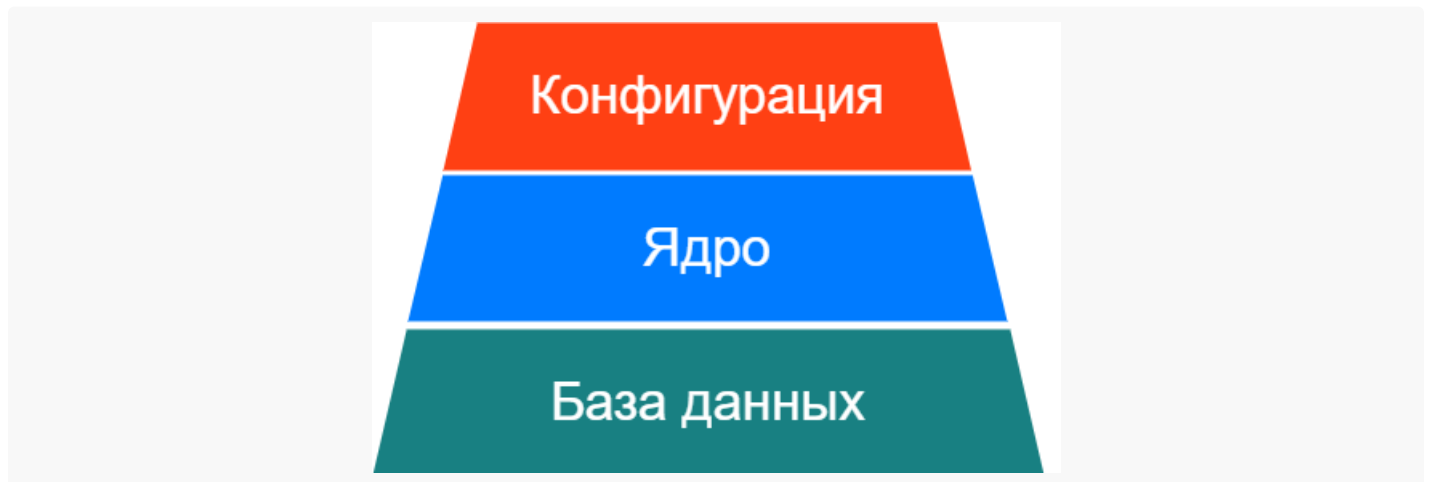
# Разработка приложений на платформе Creatio



**Creatio** — это low-code платформа, которая предоставляет широкие возможности для ускорения разработки, внедрения и масштабирования приложений. Платформа построена на открытых принципах кастомизации. Это позволяет создавать приложения разработчикам с разным уровнем квалификации — от бизнес-аналитика до разработчика полного цикла. В зависимости от сложности или типа бизнес-задачи, разработка приложений на платформе Creatio предполагает разные уровни кастомизации.

## Уровни кастомизации Creatio

С точки зрения логических уровней взаимодействия архитектура платформы Creatio может быть представлена следующим образом:



При разработке приложений в Creatio необходимо учитывать, что уровень ядра — неизменяемый компонент платформы, а разработка реализуется на уровне конфигурации и базы данных.

## База данных

**База данных** — это уровень физического хранения данных. В базе хранятся не только пользовательские данные, но и настройки приложения, а также настройки прав доступа к приложению.

Платформа Creatio предоставляет инструменты для работы с данными непосредственно из интерфейса приложения. Поэтому не возникает необходимости работы с объектами базы данных напрямую.

Существуют задачи, которые логичнее и быстрее реализуются на уровне базы данных.

Пользовательскую бизнес-логику можно реализовать на уровне базы данных с помощью представлений и хранимых процедур. Затем реализованную бизнес-логику можно вызывать в пользовательских конфигурационных элементах.

## Ядро

**Ядро** — неизменяемая часть платформы, которая представляет собой набор библиотек с реализацией базовой функциональности приложения.

**Back-end библиотеки** реализованы на языке C# с использованием классов платформы .NET Framework. У разработчика есть возможность создавать экземпляры back-end классов и использовать функциональность back-end библиотек, но при этом нельзя вносить изменения в эти классы и библиотеки.

Основные **back-end компоненты ядра**:

- [ORM-модель данных](#) и методы работы с ней. В большинстве случаев для доступа к данным рекомендуется использовать именно объектную модель, хотя прямой доступ к базе данных также реализован в back-end компонентах ядра.
- [Пакеты](#) и механизм замещения.
- [Библиотеки серверных элементов управления](#). К таким элементам управления относятся страницы, построенные на технологии ASP.NET, которые формируются на сервере, например, страницы раздела [ Конфигурация ] ([ Configuration ]).
- Системные [веб-сервисы](#).
- [Функциональность](#) основных дизайнеров и системных разделов.
- [Библиотеки для интеграции](#) с внешними сервисами.
- [Движок бизнес-процессов](#) ( ProcessEngineService.svc ). Это элемент платформы, который позволяет выполнять алгоритмы, настроенные в виде диаграмм.

**Front-end классы ядра** реализованы на языке JavaScript с использованием различных фреймворков. Они предназначены для создания пользовательского интерфейса и реализации других бизнес-задач на стороне браузера. Основная **задача** front-end компонентов ядра — обеспечение работы клиентских модулей.

Основные **front-end компоненты ядра**:

- [Внешние библиотеки](#) клиентских фреймворков.
- [Песочница \(sandbox\)](#) — специальный компонент клиентского ядра, предназначенный для обеспечения взаимодействия между клиентскими модулями путем обмена сообщениями.
- [Базовые модули](#) — файлы на языке JavaScript, в которых реализована функциональность основных объектов платформы.

## Конфигурация

**Конфигурация** — это набор функциональности, который доступен пользователям конкретного рабочего пространства, а именно:

- Серверная логика.
- Автогенерируемые классы, являющиеся продуктом работы настроек приложения.
- Клиентская логика (страницы, кнопки, действия, отчеты, бизнес-процессы и другие настраиваемые конфигурационные элементы).

Конфигурация является легко изменяемой частью приложения. Конкретную конфигурацию формируют

следующие типы элементов:

- **Объекты** — сущности, предназначенные для хранения данных, которые объединяют таблицу в базе данных и класс на серверной стороне.
- **Бизнес-процессы** — настраиваемые элементы, которые представляют собой визуальный алгоритм пользовательских действий.
- **Клиентские модули.**

## Инструменты разработки приложений

Creatio предоставляет широкий спектр инструментов для создания новых приложений и расширения существующих.

### Стек open-source технологий

Поддержка стандартных языков программирования и фреймворков ускоряет разработку, обеспечивает эффективную поддержку и разработку компонентов приложения, а также гарантирует наличие квалифицированных специалистов с необходимым стеком технологий.

Платформа Creatio поддерживает следующие **технологии**:



Для реализации сложной бизнес-логики, интеграции и настройки, Creatio предоставляет **встроенную IDE**, инструменты которой позволяют ускорить решение типовых задач по конфигурированию платформы. Встроенная IDE позволяет использовать C# или JavaScript для реализации следующих **задач**:

- Расширение и изменение функций Creatio.
- Организация взаимодействия с системами контроля версий.
- Передача изменений между средой разработки, предпромышленной и промышленной средами.

Для кастомизации Creatio разработчики могут использовать **стороннюю IDE** (например, Microsoft Visual Studio), которая позволяет работать с проектами в локальной [файловой системе](#). Это ускоряет и

упрощает процесс разработки за счет использования знакомой IDE, наличия множества плагинов, расширений и интеграций с различными инструментами разработки, системами контроля версий и т. д. Также позволяет разработчикам сэкономить время на изучении новых инструментов и сосредоточиться на разработке кода C# и JavaScript в знакомой IDE.

## Low-code/no-code разработка

**Инструменты** low-code/no-code разработки представляют собой набор визуальных drag-and-drop редакторов пользовательского интерфейса. Они позволяют выполнять следующие **задачи**:

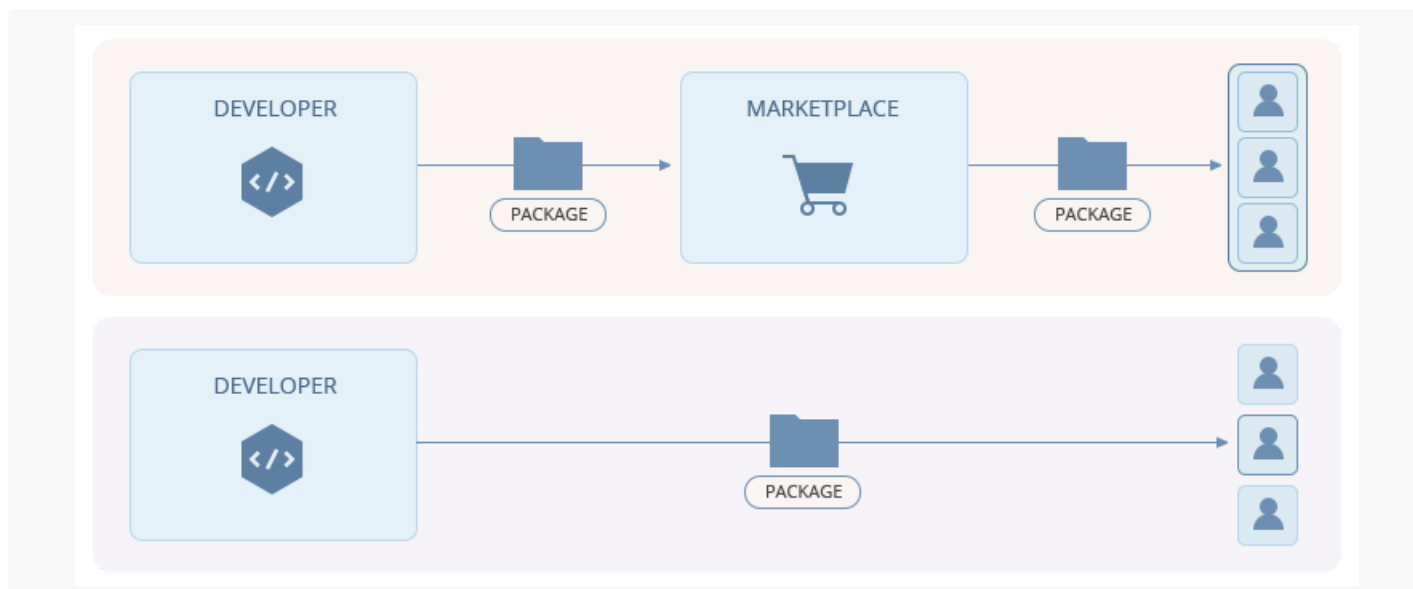
- Автоматизировать бизнес-процессы и динамические кейсы.
- Моделировать структуру данных.
- Моделировать пользовательский интерфейс для веб- и мобильных приложений.
- Создавать информационные панели и отчеты.
- Настраивать интеграции.
- Строить прогнозные модели и т. д.

Большинство вариантов использования покрываются low-code инструментами. Минимальное ручное кодирование, а также легкодоступные встроенные инструменты сделали разработку приложений доступной для бизнес-экспертов, опытных пользователей, аналитиков и [citizen developers](#). Описание no-code инструментов содержится в статье [Low-code/no-code](#).

## Механизм пакетов

Независимо от инструмента, используемого для разработки приложения Creatio, все кастомизации упаковываются в [пакеты](#), которые представляют собой ключевой компонент архитектуры Creatio. Пакетная архитектура позволяет выделять целостные функциональные блоки и оформлять их в виде отдельных модулей, управлять иерархией пакетов и версионностью. Это дает возможность быстро расширять конфигурацию, а также переносить изменения между средой разработки, предпроектной и проектной средами. Этот же механизм лежит в основе решений, публикуемых на площадке [Marketplace](#).

**Любой продукт Creatio** — это набор пакетов, которые устанавливаются поверх ядра Creatio. Пакет содержит схемы объектов, исходный код, бизнес-процессы, отчеты и т. д. Он также может содержать сторонние сборки, SQL-сценарии, системные настройки и пользовательские данные.



Модель расширения Creatio основана на **принципе открытости-закрытости**, при котором основная логика приложения закрыта для прямых манипуляций, но открыта для расширения и модификации путем добавления пользовательских пакетов. Каждый пакет может быть расширен пакетом другого издателя (партнера-интегратора, разработчика Marketplace или заказчика). Это позволяет платформе эффективно совмещать out-of-the box продукты, рыночные решения и настройки клиентов практически в любой комбинации.

**Архитектура пакетов** — это основной механизм доставки и развертывания пользовательских приложений, расширений и шаблонов, полностью интегрированный в Creatio Marketplace. Creatio Marketplace представляет собой экосистему для разработки, распространения и получения кастомизаций — как пользовательских приложений и шаблонов, так и обновлений и изменений отраслевых приложений.

Пакеты не влияют на логику ядра платформы, обеспечивая бесперебойную доставку обновлений для ядра Creatio, а также параллельное развертывание и обновление настраиваемых функций.

## Независимая от СУБД архитектура и собственная ORM

Creatio — это **СУБД-независимая платформа**, в основе которой лежит [ORM](#), разработанная Creatio. Это позволяет разработчикам легко создавать и доставлять пользовательские приложения для различных конфигураций в базе данных Oracle, PostgreSQL или MS SQL Server без изменений в кодовой базе.

## Бизнес-логика, основанная на процессах

Это среда для взаимодействия между разработчиками полного цикла и бизнес-аналитиками. Используя визуальные drag-and-drop редакторы бизнес-процессов, пользователи могут создавать собственную бизнес-логику, просто нарисовав необходимый процесс. Работа с бизнес-процессами описана в блоке статей [Бизнес-процессы](#).

## Интеграции

Платформа Creatio предоставляет необходимые инструменты для [интеграции](#) со сторонними системами и приложениями, включая поддержку REST API, протокола OData, SOAP-сервисов, возможности



аутентификации по протоколам OAuth и LDAP. Интеграции можно разрабатывать как часть приложения Creatio или стороннего приложения. Сложные инструменты обеспечивают безопасность данных во время интеграции для идентификации и контроля доступа, а также управления структурой пользователей.