

Low-code

в корпоративном интерьере



Денис БАХАЕВ,
руководитель по развитию
направления «Цифровые решения»,
Siemens Digital Industries Software

Эволюция быстрой разработки

По своей сути low-code – это эволюционный этап развития средств разработки программно-обеспечения (ПО). Low-code и no-code соотносят с понятием быстрой разработки ПО (Rapid application development – RAD), основная идея которой – снижение порога входа в мир разработки ПО. Такая технология становится все более популярной, можно даже сказать, что в некоторых отраслях рынка она является драйвером роста. Во многом это обусловлено экономическими соображениями: создание ПО по классической схеме требует команды профессиональных разработчиков, тестировщиков, дизайнеров. Подобная команда требует значительных затрат, к примеру, зарплата программиста среднего уровня на российском рынке в 2021 г. может составлять 120–200 тыс. руб. Добавляя



Дмитрий КУДРЯШОВ,
менеджер по сбыту управления
«Цифровое производство»,
компания «Сименс» в России

к этому время среднего проекта от 6 до 12 месяцев, получаем уже вполне ощутимую стоимость воплощения идеи в жизнь. Однако с появлением подходов RAD и low-code проблема с временем разработки и затратами трансформируется. Такие инструменты могут использовать уже непрофессиональные программисты, стоимость труда которых на рынке значительно меньше. Более того, разработчики, предлагающие low-code-платформы на рынке, при их создании, как правило, используют Agile-подход, что позволяет быстро проверять идеи и делать прототипы.

Первое применение класса платформ, поддерживающих low-code-технологии, состоялось в отраслях рынка, интегрирующих ИТ-решения с бизнес-процессами и выпуском продуктов наиболее активно, – это банки, страховые компании и ритейл. Общей чертой этих компаний является то,

Начиная разговор о low-code-продуктах, в первую очередь стоит определить, что такое low-code и no-code. Иными словами, речь пойдет о некой концепции класса продуктов, отвечающих определенным критериям.

что они предлагают на рынок информационные услуги. Сегодня обеспечение такого бизнеса уже невозможно без использования различных ИТ-систем, вследствие чего в компаниях создаются большие ИТ-подразделения, включающие в себя, в частности, отделы разработчиков. В средних банках отдел разработки составляет около 300 человек, создающих и поддерживающих продукты для своей компании.

Традиционно существуют две основные области разработки и внедрения ИТ-решений. Первая отвечает за бэк-офис, т. е. за решения, непосредственно обеспечивающие работу бизнес-процессов, – это могут быть, например, системы электронного документооборота или ERP. Крупные холдинги и компании, как правило, идут по пути приобретения уже готовых ERP-решений и не ставят перед собой задачи самостоятельной разработки такого класса продуктов. Вторая область – ПО, реализуемое непосредственно клиентами. Рассматривая банковскую отрасль в качестве примера, можно выделить мобильные банковские приложения – ПО, которое не существует на рынке в готовом виде. Фактически у компании есть только две возможности решения подобной задачи – самостоятельная разработка и аутсорсинг. Однако оба подхода весьма трудоемки. По оценке TechRepublic,

уровень безработицы разработчиков не превышает 1,9%, а спрос на разработку программных решений продолжает расти, и, согласно данным Gartner, к 2024 году он вырастет в пять раз. Передача разработки на аутсорсинг также не решает эту проблему, поскольку нет возможности полного контроля процесса, а большое количество согласований увеличивает стоимость проекта по сравнению с «домашней» разработкой. Именно эти аспекты и становятся фактором развития low-code платформ, решающих значительную часть трудностей.

Место low-code

Рассматривая low-code-платформы, можно заметить, что они занимают особое место в ИТ-ландшафте компаний и корпораций, приобретших особую популярность в США и Европе вследствие высокой стоимости программистов. Здесь стоит отметить, что low-code – это не просто среда разработки подобно Borland Delphi. Платформа включает в себя функции разработки, установки, управления командой и обратной связью, отслеживания разработанных решений. Такой платформой, например, является решение Mendix.

В 2018 г. компания Mendix стала активно продвигаться на рынок промышленности. Для рассмотрения возможностей применения Low-code платформ можно выделить три области:

- программные решения, которых нет на рынке. Здесь заказчик еще не может сформулировать все требования и пожелания к разработке. Условно назовем эту область инновационными программными решениями;
- интеграционные решения, возникающие вследствие потребности в получении информации и данных из разных систем, уже существующих в компании. При решении таких задач клиенты, как правило, самостоятельно создают порталы для интеграции;

- программные решения уровня ERP, MES (Manufacturing Execution System), PDM (Product Data Management), т. е. умеющие решать конкретную задачу.

Технология low-code оправдывает себя в первых двух случаях, когда еще нет четких и конкретных требований к системам, что означает необходимость осуществлять быстрые измене-

это сопоставимо с покупкой стороннего ПО, а внедрение готового продукта отнимет гораздо меньше времени.

Определив области применения low-code-платформ, остановимся на задачах, решаемых с помощью быстрой разработки. Классическими здесь будут являться обеспечение автоматизации бизнес-процессов и увеличение

Рассматривая low-code-платформы, можно заметить, что они занимают особое место в ИТ-ландшафте компаний и корпораций, приобретших особую популярность в США и Европе вследствие высокой стоимости программистов.

ния. При классическом подходе разработки программных решений это становится уязвимой областью. Внести изменения в продукт на финальных стадиях готовности уже довольно сложно, поскольку будет необходимо повторить более половины пройденного процесса создания ПО: согласование технического задания, разработка, отладка, тестирование, рефакторинг. При использовании методологии Agile, применяющейся в инструментах разработки low-code, можно менять логику визуально с помощью перетаскивания мышкой соответствующих объектов. Более того, это по силам даже непрофессиональному программисту. Здесь стоит отметить, что в некоторых сложных проектах использование low-code хотя и позволяет значительно сократить количество профессиональных разработчиков, но не исключает их полностью. Что касается третьей области решений уровня ERP, то целесообразно приобретение готового продукта. Разработать подобное решение с использованием low-code возможно, однако по экономическим и трудозатратам

эффективности предприятия. Это может быть электронный документооборот, управление проектами, B2B, B2C и HR-порталы, а также аналогичные модульные системы.

Обычно такие задачи находились рядом с ERP, поэтому большинство вендоров low-code-платформ предлагают свою ERP-систему с возможностью доработки low-code-средствами. Однако сегодняшние задачи не ограничиваются ERP, особенно в крупных холдингах с требованиями по тонкой настройке и интеграции. Как пример, low-code-платформа Mendix делает ставку именно на это, а потому не имеет привязки к какой-либо конкретной ERP или другой системе.

Другим типом задач, решение которых лежит в области low-code платформ, являются коллаборативные решения и устаревшие системы (legacy). Здесь подразумевается умение быстро получать данные из различных систем и построение на их основе функционала для бизнеса. В таких задачах low-code можно перепутать с классом BI-систем, однако области применения этих

систем различны. BI решают конкретные задачи, в частности, сбор информации из различных источников с последующей аналитикой. Low-code также обеспечивает возможность выполнять подобные задачи, однако лучше их оставлять BI-системам с последующим использованием результатов в приложениях на базе low-code. RAD позволяет заниматься вопросами увеличения бизнес-эффективности (например, управление персоналом – создание портала для руководителя с полной прозрачностью данных о сотрудниках). Можно получить данные от HR, от системы контроля и управления доступом (СКУД) для мониторинга времени прихода/ухода с работы, от системы управления проектами (чтобы понимать, в какие активности вовлечен сотрудник) и многое другое. Именно такие задачи можно легко и быстро решать методами low-code.

В этой же области лежит работа с устаревшими данными. На определенном этапе жизненного цикла многие предприятия сталкиваются с тем, что существующие системы уже не отвечают требованиям бизнеса – нет нужной функциональности, стабильности, надежности. Эта проблема может решаться несколькими классическими способами. Первый и самый очевидный – приобретение новой системы, закрывающей потребности организации. Очевидно, что у такого подхода есть большой плюс – новая система уже присутствует на рынке вместе с ее интеграторами. Однако неизбежно возникает необходимость значительных временных затрат на подготовку ТЗ, изучение рынка и опыта, а если речь идет о крупной организации, то и кастомизации. Более того, приходится оплачивать поддержку этой системы, а также всегда быть готовым к дополнительным затратам в случае любой доработки. Отдельный пункт здесь вопрос обучения персонала. Как правило, сотрудники не любят изменений и внедрения новшеств, что неизбежно приводит к задержкам в получении реальной пользы.

Другой вариант – модернизация уже существующей системы внешними специалистами или собственными силами. Такая задача не выглядит легче, ведь она требует, как минимум, технологических возможностей для модернизации, а также компетентных и мотивированных специалистов в штате, готовых взяться за подобные работы. Рассматривая внештатных специалистов, можно увидеть все те же проблемы – сложность контроля процесса, дороговизна, значительные временные затраты.

Low-code-технологии предлагают иной вариант решения таких задач – возможность плавного перехода на собственные разработки. Устаревшая система все еще функционирует некоторое время, однако из нее выгружаются данные и с использованием коннекторов строится дополнительный функционал. Стоит заострить внимание на том, что реализовать подобное возможно не дорогими высококвалифицированными разработчиками и программистами, а штатными специалистами и бизнес-аналитиками со значительной денежной экономией. Выбрав такой путь, компания решает и параллельные задачи: персонафикация решения непосредственно под требования заказчика, обучение сотрудников без отрыва от производства. Примером реализации подобного подхода является компания Copoco Phillips, где более 50% персонала умеет работать на платформе low-code-разработки. Если у сотрудников возникает новая бизнес-идея, связанная с улучшением производительности, они могут самостоятельно реализовать ее. Результат такой стратегии – появление более 100 приложений. В компании low-code внедрена на уровне корпоративной культуры – даже при приеме на работу знание этой технологии является ощутимым конкурентным преимуществом.

Инженерное программирование

Сейчас в low-code появляется новый класс задач

– инженерные. Стоит отметить, что производственные компании и холдинги всегда отставали в области ИТ-решений от других сегментов рынка, однако текущий уровень зрелости таких предприятий подошел к этапу, при котором low-code-технологии являются востребованными. Кроме того, в отличие от банковских, страховых и других подобных компаний на предприятиях существует особый ИТ-мир – PLM-системы. Применение low-code-производственными компаниями позволяет интегрировать оба мира – промышленность и ИТ-технологии для решения задач операционного бизнеса. Такая синергия приводит к повышению эффективности бизнес-процессов и соответствии трендам рынка в части роста их интеграции в эпоху гиперавтоматизации производств. Применение low-code-технологий в промышленности пока не получило большого распространения, тем не менее имеет большой задел на будущее.

Современные low-code-решения дают заказчикам возможность не обращаться в специализированные компании для разработки мобильных приложений, а заниматься этим самостоятельно с минимальными издержками. Использование дополненной реальности (AR) и искусственного интеллекта (AI) позволяет промышленным компаниям создавать инновационные продукты и осуществлять поддержку на новом уровне. Безусловно, эти технологии будут развиваться в сторону увеличения абстракции инструментов программирования в целях решения большого пласта различных ИТ-задач непрофессиональными разработчиками. Профессиональные разработчики по-прежнему останутся востребованными – распространение low-code-технологий не означает их ненужности. Хотя подобные решения и снижают планку необходимой компетенции для разработки, low-code-технологии все равно не смогут стать универсальным ответом на все существующие ИТ-задачи. ■