

ФАКУЛЬТЕТ РАДИОФИЗИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

ИССЛЕДОВАНИЕ ТЕХНОЛОГИИ NOCODE И ЕЕ ИНТЕГРАЦИИ С ТЕХНОЛОГИЯМИ ВИЗУАЛЬНОГО АНАЛИЗА ТЕКСТА

Р. И. Андросюк, А. А. Ленюски, В. П. Можейко

*Белорусский государственный университет, г. Минск;
rtn.androsiuk@gmail.com; vladislav.mozheiko321@gmail.com
науч. рук. – К. С. Мулярчик, канд. техн. наук, доц.*

Изучена распространенность и популярность использования NoCode технологий. Выявлены причины возникновения NoCode инструментов и их целевая аудитория. Разработан прототип NoCode системы, эмулирующий NoCode среду разработки. Сформулированы основные принципы создания эффективных NoCode платформ. Выведен концепт использования текстового представления программы вместо визуального. Изучены на практике возможности программного извлечения информации из текста.

Ключевые слова: технологии Nocode, бизнес решения, визуальная абстракция, визуальные формы, смысловой анализ текста.

ВВЕДЕНИЕ

NoCode — это концепт, при котором пользователь имеет возможность создать собственное приложение без написания программного кода. Сам по себе NoCode не нов [1], но сильную популярность данный подход к разработке ПО получил в последние несколько лет, в чём мы убеждаемся согласно сервису Google Trends на рис. 1.

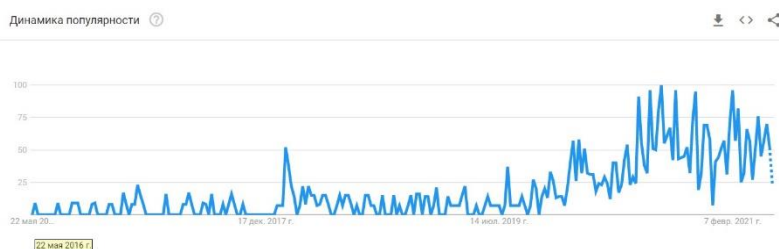


Рис. 1. Популярность запроса «nocode» согласно Google Trends

NoCode сервисы презентуют себя на рынке как «программирование для непрограммистов». Однако во многих ситуациях NoCode инструменты не справляются со своими обещаниями. Большая конкуренция между различными сервисами и поспешные попытки заполнить свой продукт всевозможным функционалом неизбежно наносят урон по удобству и доступности использования данного сервиса.

Несмотря многие мифы, что NoCode системы небезопасны, непроизводительны и нестабильны, на данный момент вполне возможно создать конечный продукт, пользуясь исключительно NoCode технологиями.

ПРОТОТИП NOCODE СИСТЕМЫ

В качестве практической части нашей работы мы разработали прототип NoCode системы. Его целью является изучить технологии NoCode, их принципы и как они могут быть использованы на реальном продукте, а также проанализировать недостатки NoCode и проблемы при разработке NoCode сервисов.

По своей сути прототип является визуальным редактором для создания уникальных вариаций классической игры «змейка». Редактор изображен на рис. 2. Правила игры выстраиваются тремя типа блоков: функциональными, логическими и блоками событий. Также присутствует возможность войти в код каждого блока и отредактировать его. За счет создания связей и настройки блоков пользователь создаёт собственную уникальную вариацию игры.

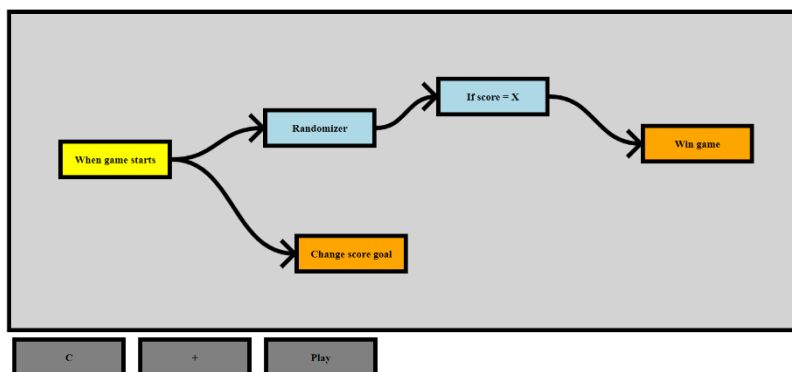


Рис. 2. Интерфейс нашего NoCode редактора

Один из принципов, который мы сформулировали в процессе разработки нашего прототипа — чтобы в перспективе создать успешный NoCode сервис, в нём необходимо соблюсти баланс практичности и доступности. Инструмент должен иметь чёткую и интуитивную логику построения продукта, а также сильный набор функционала, опирающийся на заранее установленные правила, однако не нарушающий их.

Второй принцип можно сформулировать так: NoCode платформы зависят от абстракции. Успешные NoCode инструменты позволяют изолировать от пользователя все сложности и тонкости классической разработки на том уровне абстракции, куда владельцу бизнеса вмешиваться нет ни желания, ни необходимости.

Почти все ныне появляющиеся NoCode сервисы растут из-за быстро растущего спроса на IT разработчиков, которых зачастую попросту не

хватает. Более того, не каждый бизнес может себе позволить иметь за собой команду или разработчиков, или дизайнеров, или аналитиков и т.п. Потому и возникает необходимость в доступных инструментах для достижения хоть и примитивных, но кастомизированных решений малого и среднего бизнеса.

ВИЗУАЛЬНЫЕ ФОРМЫ И ИХ ПРИМЕНЕНИЕ В NOCODE

Технология NoCode, которая была описана выше, уже давно существует и нашла свое применение. Но для её использования нужно всё же знать какие-то базовые понятия алгоритмов. Поэтому для большинства людей, пусть и в визуальном редакторе, но составление программы с использованием NoCode является трудоёмкой задачей. В то же время, словами люди могут легко описать, что они хотят получить в результате. Поэтому предоставление людям возможности составлять текстовое описание необходимых требований вместо составления алгоритмов из блоков сделает написание программ на основании NoCode ещё более доступным для большинства людей.

Для исследования возможности выбирать необходимые элементы из текста была сделана попытка создать программу, которая будет строить визуальные формы на основании предоставленного ей текста.

Кратко расскажем о том, что подразумевается под визуальными формами. За основу был взята теория из книги Дэна Роэма [2]. Суть её заключается в том, что любую визуальную информацию мы воспринимаем мгновенно, просто посмотрев, например, на изображение. То есть воспринимаем общую картину происходящего перед нашими глазами. Но при использовании только визуальной информации мы упускаем мелкие детали. Например, мы не знаем точного расстояния между объектами, которые видим. В то же время при чтении текстовой информации, мы получаем её последовательно, слово за словом и постепенно строим у себя в голове общую картину. Поэтому визуальная информация дополняет текстовую и наоборот.

Также следует отметить, что следуя данной теории, полностью описать любую информацию можно, используя шесть основных вопросов: «Кто?/Что?», «Где?», «Сколько?», «Когда?», «Почему?», «Как?».

Каждая визуальная форма отвечает на определённые вопросы, например, диаграмма на рис. 3 отвечает на вопрос «Сколько?». Таким образом, используя определённые визуальные формы и алгоритмы для их создания, мы можем получать из текста необходимые фрагменты информации. На рис. 3 приведён пример получения данных из предложения: «На первом занятии присутствовало 85,5% студентов и 78,2% на последнем».



Рис. 3. Результат работы программы для поиска и построения диаграмм

Таким образом, если рассчитать, ответов на какие вопросы требует использование того или иного блока из NoCode, то мы сможем составить программу на основании текста, используя алгоритм, похожий на алгоритм поиска визуальных форм.

В случае, если программе, написанной на NoCode, будет не хватать данных или же описание какого-либо блока она не сможет понять, она может сама вывести пользователю вопрос, используя вышеупомянутые шесть вопросов. В случае со змейкой, если пользователь, например, не задал условие победы, то можно вывести вопрос: «Сколько необходимо набрать очков для победы?». Ответив на этот вопрос, пользователь дополнит свой изначальный запрос и задаст необходимые параметры.

ЗАКЛЮЧЕНИЕ

Подводя итог проделанной работе, можно сказать, что не существует "лекарства от всех болезней", решения для всех проблем. В роли разработчика ПО вам необходимо рассматривать и приоритезировать время до выхода на рынок, гибкость развития продукта, полную стоимость владения и обслуживания продукта и т.д. Фокусируясь на одном, можно упустить другое, и именно в таких ситуациях целесообразно применение NoCode инструментов. Хотя NoCode технологии и развиваются в экспоненциальной прогрессии, мы ещё далеки от того, чтобы полностью отказаться от классических языков программирования.

Библиографические ссылки

1. “No Code” is not a new thing // medium.com [Электронный ресурс]: URL: <https://medium.com/always-be-coding/no-code-is-not-a-new-thing-92cda0109c57> (дата обращения: 21.3.2021).
2. Визуальное мышление. Как «продавать» свои идеи при помощи визуальных образов / Дэн Роэм; пер с англ. О. Медведь – М.: Манн, Иванов, Фербер, 2013.