Технологическая платформа G2 ускоряет разработку IT-систем

Зачем Газпромбанк создал собственную платформу и какие проблемы разработчиков она решает? Какие решения уже созданы с помощью платформы G2? Об этом «Б.О» рассказали вице-президент — начальник департамента развития платформ корпоративного бизнеса Александр Петров и заместитель начальника департамента Илья Челпанов


Александр Петров


Илья Челпанов



Текст
ВАДИМ ФЕРЕНЕЦ,
ОБОЗРЕВАТЕЛЬ **«Б.О»**

— Что стало причиной разработки технологической платформы G2 для клиентских сервисов и автоматизированных систем?

Александр Петров: Наша команда более 20 лет занимается разработкой ПО на самых разных IT-платформах и технологических стеках. Со временем появилось понимание как плюсов, так и минусов различных решений и желание создать собственный продукт, который объединил бы все плюсы, а от минусов отказаться вовсе.

На нашем рынке пользуются популярностью две отечественные вендорские платформы, которые действительно помогают программистам быстро разрабатывать код и быстро выводить продукты на рынок. Но они имеют или устаревший «десктопный» интерфейс, или специфический и сильно ограниченный по возможностям расширения веб-интерфейс, или неразвитые средства работы с данными, или несколько из этих особенностей одновременно. Особенности интерфейсов существенно ограничивают возможность применения этих платформ для разработки дистанционных клиентских сервисов. А если платформа «Oracle-зависимая», что не редкость для банковского сектора, то ситуация усугубляется.

Актуальная тенденция — появление достаточно мощных open-source-фреймворков, что сделало возможным смещение точки разработки прикладных систем в сторону «инхаус»-разработки. Фреймворки созданы разработчиками для разработчиков, хорошо решают свои задачи по облегчению их труда. С точки зрения их применения для разработки прикладных бизнес-систем они являются достаточно низкоуровневыми, что приводит к необходимости содержания значительных по численности команд и к большим трудозатратам.

Другой актуальной тенденцией является большой прогресс в области эргономики интерфейсов и визуализации данных. Сейчас работа дизайнеров занимает значительную долю в общих усилиях, и в этой части нам хочется достичь лучших результатов за счет сокращения рутины и перераспределения усилий на глубокий анализ и творчество.

Анализ представленных на рынке платформ выявил пять основных групп проблем. В виде шутки их можно обобщить так: «Если использовать современный стек, то это очень дорого, но зато красиво. Если использовать большинство платформенных решений, то это быстро, но некрасиво».

Итак, перечень недостатков. Во-первых, это дороговизна самих решений, а также высокая совокупная стоимость владения ими. Во-вторых, — долгая разработка. В-третьих, — несовременные и неэргономичные интерфейсы.

— Четвертая проблема, очевидно, это отсутствие какой-либо системности при проектировании?

Александр Петров: Совершенно верно! Это та самая боль руководителей IT-проектов, от которой все мы достаточно часто страдаем. В небольших проектах она еще не очень чувствуется. Но когда речь идет о разработке и развитии больших и зрелых систем, с которыми работают тысячи пользователей, ситуация становится совсем иной.

В некоторых наших решениях речь идет примерно о 150 обновлениях в неделю, поэтому проблемы с системностью процесса проектирования встают в полный рост. Это, например, локализация точек изменений, а также оценка степени влияния производимых изменений на окружение.

Опыт развития мировой софтверной индустрии говорит о том, что подобный набор проблем никак по-другому не решается, кроме как путем перехода всех процессов на системные рельсы. Добавлю, системный проект начинается с самого начала, как системный. Что имеется в виду? Когда инженеры приступают к разработке программного обеспечения, это вовсе не означает, что они сразу же должны садиться и кодировать. Вовсе нет. Сначала необходим этап проектирования в специальной автоматизированной среде. Полученный здесь продукт должен опять же автоматически транслироваться в некий каркас, в некую среду реализации, и при этом должна быть сохранена однозначная связь между средой проектирования и средой реализации. Для чего она нужна? В случае необходимости внесения изменений в какую-то конкретную точку, как на чертеже, должны быть видны сопутствующие эффекты влияния этой процедуры на пограничные блоки, функции, процедуры, процессы и т.д.

Вот этого системного подхода нам не хватало всегда и везде. Да, есть определенные практики, но зачастую они используются как-то не так либо не используются вовсе. Мы же планировали эту системность сделать по умолчанию частью нашей платформы, на базе которой создаются новые прикладные автоматизированные системы.

— А какова пятая проблема?

Александр Петров: Финальный пункт в нашем списке проблем — зависимость от конкретных решений разработчиков. Если разработчики уходят из проекта, то руководитель остается один на один с недоделанным решением, на которое пытается найти новых людей. Ситуация сравнима с ремонтом в квартире, который не доделали. Один мастер ушел, а новый говорит, что предыдущий мастер что-то сделал не так, поэтому все нужно переделывать с нуля. Вот так же и в нашей отрасли. Приходит новый программист и говорит, что все сделано неправильно, а если даже и правильно, то он лично все равно не понимает, что это такое и как оно вообще работает. В итоге что? Правильно: нужно все переделывать.

И это стало вызовом для нас, поскольку мы хотели получить платформу, которая легко бы подхватывалась в случае возникновения форс-мажора другими разработчиками, благодаря чему программный продукт продолжал бы дорабатываться без потери качества с соблюдением установленных сроков. Достигается это за счет того, что слой программного кода прикладной логики над слоем технологической платформы достаточно «тонкий», при этом используемый язык программирования дает лаконичный, компактный код, а логика оперирует объектами предметной области.

Газпромбанк разработал технологическую платформу G2 для клиентских сервисов и автоматизированных систем. Разработка позволит отказаться от ПО поставщиков, ушедших с российского рынка.

Задача импортозамещения решается благодаря использованию опенсорсных компонентов технологического стека с поддержкой от отечественных компаний, в частности СУБД PostgreSQL.

Использование платформы G2 позволяет:
- сократить время разработки новых систем в два раза и более;
- уменьшить количество задействованных разработчиков;
- стандартизировать архитектуру приложений.

Другими преимуществами разработанной Газпромбанком платформы являются гибкий функционал, включающий более 140 инструментов, а также удобный и эргономичный интерфейс. Грамотно выстроенный процесс проектирования приложений помогает быстро, эффективно и системно управлять изменениями.

— Что именно вы понимаете под термином «платформа»?

Илья Челпанов: Это действительно важный вопрос. В нашем случае технологическая платформа — это средство разработки приложений, включающее инструменты разработчика, дизайн-систему и среду исполнения. Но зачастую под платформой понимают некое готовое бизнес-приложение, которое помогает вести бизнес.

Александр Петров: Это совокупность неких системных подходов с их программной поддержкой, обеспечивающая создание «правильного» системного каркаса, который дальше легко наращивается прикладной логикой. Это low-code в буквальном смысле — слой прикладного кода «тонкий», но, что важно отметить, мы не заменяем код рисованием схем. Я включаю в понятие «платформа» и методологию, подход к проектированию «от модели», позволяющий при изменении требований к системе четко определить место внесения изменений и границы влияния изменений.

Илья Челпанов: Наша платформа G2 не может «из коробки» заменить ERP-системы, поскольку это платформы для прикладных бизнес-систем, за некоторыми исключениями. А у нас G2 — технологическая платформа, в принципе не ограничивающая тип системы, которая может быть на ней реализована, но в то же время она не дает и какого-то готового решения. Для его получения необходимо запрограммировать логику целевой системы. G2 поможет сделать это быстрее, и система будет иметь универсальный и привлекательный интерфейс.

Александр Петров: На платформе G2 уже разработан ряд прикладных решений. Это в первую очередь система управления ликвидностью как дистанционный сервис для клиентов банка, масштабный проект, который мы вели на протяжении трех лет, в числе прочего свой бизнес-модуль класса Business Intelligence (BI). В подобных решениях идет работа с таблицами и связями между ними. Обычным пользователям крайне сложно с этим работать, поэтому зачастую привлекаются программисты.

Мы сделали такой модуль, работающий в терминах предметной области, которые собираются для каждой отрасли, и выглядящий, как дерево с ответвлениями. Пользователи, погружаясь в предметную область, получают возможность строить различные витрины данных, а на их основе — самые разные отчеты. С учетом расширенных возможностей G2 в области дизайна интерфейсов вставить получаемые отчеты в презентацию в виде настоящих графиков или таблиц, а не картинок не составляет проблем.



— Таким образом, для чего предназначена платформа G2?

Александр Петров: Наша платформа нужна для того, чтобы в автоматическом режиме создавать каркас приложения. Причем работа над самим каркасом имеет целью сделать его как можно более зрелым. Что это означает? Период времени между генерацией и началом его использования в «боевых» условиях должен быть как можно меньше. И чем меньше это время, тем более зрелыми являются каркас и сама платформа.

Наша задача — в идеале сделать такую платформу, чтобы нажатием кнопки формировалось приложение, а на следующий день оно могло быть уже установлено в промышленную среду и отдано пользователю. Понятно, что еще предстоит преодолеть немалый путь, тем не менее платформа уже сейчас берет на себя более 50% той рутины, которую раньше приходилось выполнять разработчикам вручную.

Резюмируя, скажу, что основная цель платформы G2 — это формирование каркаса бизнес-приложения, которое обладает необходимым набором функций, работает с данными, а также современным инновационным и эргономичным интерфейсом. Дальше разработчики доделают какие-то недостающие алгоритмы, и готовое приложение уйдет в эксплуатацию к заказчику.

— Вы не раз выделяли функционал интерфейса. Почему?

Александр Петров: Да, это правда. Мы добились того, что нет необходимости привлекать фронтендеров и дизайнеров, и без их участия программы, созданные на платформе, имеют стильный интерфейс. Все эти дизайнерские вещи, знания, подходы, принципы эргономики в том числе, по умолчанию заложены в платформу и наследуются каркасами, которые она создает.

На наш взгляд, именно таких решений на рынке недостаточно, а мы серьезно вкладываемся в это направление и активно его развиваем. Более того, мы поставили этот процесс на научные рельсы благодаря сотрудничеству с Университетом ИТМО из Санкт-Петербурга. Лучшим его студентам и аспирантам оказались интересны наши задачи, поэтому они на взаимовыгодной основе помогают их решать.

— Расскажите немного об истории создания G2.

Илья Челпанов: Не следует думать, что платформа G2 появилась исключительно вследствие процесса обретения технологического суверенитета. Мы эту идею вынашивали давно. И когда пришло время решать задачу импортозамещения, у нас уже был собственный IT-инструмент.

Можно сказать, что версия G1 была нашей внутренней экспериментальной разработкой, которую мы сделали в 2014 году. Если говорить в целом про эти G-платформы, то они рождались вокруг системы управления ликвидностью.

Важно отметить преемственность платформы, поскольку работа над ней основывалась на классических принципах, и разработчики собрали все лучшее, что и было реализовано на современных языках в современном стеке.

Александр Петров: На G2 была создана IT-система для казначеев корпораций, состоящая примерно из 20 модулей. Сейчас с ней работают более 1,3 тыс. пользователей из нескольких сотен организаций. Что касается литеры G — это наша любимая буква, которая ассоциируется с такими словами, как «Газпромбанк», Great — «великий», Global — «глобальный».

Второе направление нашего развития — работа с банковским сектором: в частности, в рамках программы импортозамещения мы избавились от зависимости от системы SWIFT Archive.

Скажу честно: мы раза три раза «подходили к снаряду» при создании этой платформы и делали это на разных языках программирования. И у нас на самом деле неплохо все получалось, хотя по большому счету это был факультатив, все приходилось делать нам самим, условно говоря, по ночам. Чтобы закончить работу, не хватало ресурсов. Но так получилось, что результаты наших исследований и экспериментов прекрасно вписались в основную работу, и вот тогда все и «выстрелило».

— Есть ли планы попробовать силы на реально амбициозном проекте?

Александр Петров: Такой масштабный вызов перед нами действительно поставлен. На платформе G2 сейчас создается собственное решение класса «Расчетный центр банка». Подобная вендорская система была внедрена в Газпромбанке еще в 2008 году. Сейчас в рамках программы импортозамещения принято решение этот продукт заменить собственным.

Проект открыт, и вся команда самым активным образом включена в работу над этой сложной инновационной системой.

— Какие архитектурные особенности можно выделить?

Илья Челпанов: Одной из важных особенностей является реализация проверенного временем подхода «Разработка, управляемая моделью». Мы реализовали интеграцию с используемыми нами средствами моделирования систем и планируем развивать платформу в этом направлении.

Другой особенностью является применение кодогенерации по метаданным модели и высокоуровневым описаниям с использованием Kotlin DSL для этих описаний.

Еще важный момент: часто известные нам платформы работают так, что простые вещи делаются на них очень просто, а сложные сделать попросту невозможно, поскольку они отклоняются от функционала какого-либо платформенного фреймворка или возможностей языка. Использованные нами архитектурные подходы позволяют реализовать принцип: «Простые вещи сделать просто, сложные вещи сделать можно». Это достигается за счет использования универсальных языков и неограниченных возмож-

ностей по применению разработанных для них библиотек и компонентов.

— Что можно увидеть «под капотом» G2?

Александр Петров: Платформа представляет собой набор библиотек и инструментов разработчика на языках Kotlin и TypeScript, в качестве СУБД применена свободная PostgreSQL.

Выше уже говорилось о том, что когда ты используешь псевдоязык в платформе, то вольно или невольно формируешь вокруг него свой узкий рынок. Понимая эту проблему, мы сознательно не стали придумывать свой нишевый язык и выбрали широко распространенный язык программирования Kotlin, чтобы не создавать свой рынок, а использовать знания специалистов и по Java, и по Kotlin. А таких на рынке немало.

Кроме того, текст программ на Kotlin более лаконичный, то есть нужно писать меньше, а сам код при этом более красивый и понятный. Это, безусловно, является его плюсом.

Илья Челпанов: Еще один используемый язык: TypeScript — это расширение JavaScript, поэтому исполнение происходит на стандартной JavaScript-машине в браузере. Его синтаксис обеспечивает проверку типов на этапе написания, потому что это реальная проблема JavaScript. В G1-платформе как в клиентской части, так и в серверной, использовался чистый JavaScript. Именно тогда на этапе прототипа стали очевидны все минусы этого подхода.

Оба указанных языка программирования выбраны неслучайно, поскольку они обладают встроенной статической типизацией. Поэтому большое количество логических ошибок они позволяют выявить еще на этапе написания программы, а не тогда, когда программа уже работает. Таким образом, статическая типизация хороша для написания сложного, но быстрого кода.

— Часто мы слышим вопросы о том, как реализована шина данных (ведь на платформах клиентов она есть), как будет происходить совместная работа.

Илья Челпанов: Поскольку в прикладном решении самое важное — это логика системы, у нас вопрос шины находится на втором месте. Кроме того, использование современного стека позволяет подключиться к любой шине данных, включая отечественные разработки. Так что это второстепенный вопрос.

Александр Петров: Согласен! Мы планируем сотрудничать с разработчиками IT-систем пользователей с точки зрения объединения прикладных бизнес-функций, а не с точки зрения полной интеграции. Для этого у нас предусмотрены API. Поэтому, если на нашей платформе кто-либо разрабатывает приложение, то неважно, какая у него IT-архитектура. Она может быть монолитная, микросервисная и т.д. Важно, что разработан набор модулей, которые взаимодействуют между собой посредством межмодульных API. Если чего-то не хватает, то написать дополнительный модуль не представляет особых проблем.

— А как же BPM?

Александр Петров: Мы сознательно не стали использовать BPM-движки. Почему так? Когда их предлагают, уверяют, что больше от программистов ничего не понадобится. «Вот смотрите, зайдете в графический редактор, настроите что-то, вот тут стрелочку протянете от одного объекта до другого, здесь галочку поставите, и все», — примерно такие объяснения приходится слушать.

Но наступает момент внедрения. Вендор на первых порах чем-то помогает, дотягивая время до этапа эксплуатации. И тут приходит первая задача по изменению. Если схема простая, протянуть стрелочку от пункта А в пункт Б еще как-то можно. А если задача «развесистая и кудрявая», то бедолага-технолог садится и думает: «Так, а если я здесь внесу изменения, как это и на что повлияет? Я не знаю. Пойду-ка я к программисту». Разработчик смотрит на это все и говорит: «А я вообще не понимаю, что это такое. Это какая-то филькина грамота. Если все это было в отладчике или в среде разработки, смог бы понять. А так — извините, идите и сами решайте свои проблемы».

И тут все встает. Программисты помочь не могут, технологи решить задачу не в состоянии, зато BPM-система успешно внедрена. Уверен, что многие сталкивались с такими историями. Поэтому мы в своей работе стараемся не использовать такие движки. При этом мы очень уважаем подход low-code, но он достигается несколько другими способами, в частности (в нашем случае) через автоматическое формирование продвинутого каркаса приложения с помощью G2. И все, что хочется запрограммировать, лучше делать в среде разработки, которая архитектурно правильно спроектирована, где все разложено по полочкам, легко масштабируется, просто производится рефакторинг кода и т.д.

— С кем и как можно коммуницировать по поводу G2?

Александр Петров: Что касается способов взаимодействия, возможны несколько вариантов. Во-первых, можно получить доступ к платформе и самостоятельно разрабатывать на ней необходимые приложения. Как мы уже говорили, это помогает существенно сократить время на разработку и соответственно снизить T2M.

Второй вариант — заказать у нас разработку приложения на платформе. Третий вариант — воспользоваться одним из наших готовых решений, таким как «Казначейство» или «Архив».

Во всех вариантах мы оказываем полную консультационную поддержку, помогаем с внедрением решений и объясняем, как по максимуму использовать возможности платформы. Б.О