Domain Driven No-code Platforms

I'm generally skeptical (like most developers) about general purpose no-code platforms. Interestingly though, I myself think all the time about how to build without writing code when writing most softwares. By the time I get to fleshing out the 11th feature, I'm thinking about abstracting them away enough, so that my 12th feature is a simple configuration tweak in some JSON / XML.

For example, in one of the complex projects I worked on, it demanded a boatload of forms (with some amounts of custom tailoring) like most projects do. After writing several NewFormForX.vue components, I realized I can get away with writing a general purpose form renderer that parses a special JSON and renders a dynamically constructed Vue component.

Strictly speaking this is not even remotely close to a no-code approach, but build a GUI for constructing this JSON itself and you are one step towards that direction. So maybe, just maybe the industry's recent love affair with no-code isn't just another hype.

But most low-code / no-code platforms feel like they promise way too much than what they are really capable of. I recently put out an HN comment with my personal checklist of how I measure these platforms and frankly not many of them cross that benchmark.

A more interesting perspective is that any business critical complex software will eventually have some no-code way of extending itself.

Take spreadsheets for example. Nobody typically associates it with no-code, but in reality it's the most practical no-code platform that has already penetrated the business market.

Or take ERPNext - an open source ERP system. At its core, every single business object in ERPNext is nothing but a special purpose JSON file. They threw in a GUI builder for constructing that JSON and now it's the most underrated no-code platform to build an entire ERP system.

Ditto with any sufficiently complex software like Salesforce, ServiceNow, Workday or SAP. They all extend a way to extend the base software without writing much code, yet as incredibly useful as they are, we never call them no-code platforms.

Comparing these two realities, I'm increasingly convinced that there are parts that the new age no-code platforms get right and some important parts that they overlook as non-significant. Each business has their own definition of roles, users, groups and email lists. Each business has its own business objects, some that are independent of any existing system and some that are tightly integrated with some existing system - like a lead in a CRM.

General purpose no-code platforms don’t take these differences very seriously. But the domain oriented no-code systems (Spreadsheets, SAP, Workday, etc) get these business concepts right enough to build useful things with.

In my opinion, the next really successful no-code platform will be the one which gets these nuances right, such that a business can tell where their role hierarchy resides, what business domain objects reside in what system. This way these tools can be used to build truly useful and complex business processes, instead of being cornered into some simple use cases.

- Joe
- joe-lewis.in