

The Rise of LLM-Powered Software: How Open Source is Fueling a New Era of App Development

[Assem Hijazi – Experience, Knowledge & Ideas](#)

Open-source software has always been seen as a game-changer. Free, flexible, and community-driven. But let's be honest—how many people actually open the source code, review thousands of lines, and check for bugs or security holes? Almost no one. The reality is, *open-source code is only as useful as what you can do with it*.

Now, imagine flipping the script. Instead of open-source code being something you install and trust blindly, what if it became **fuel for LLMs (Large Language Models)** to build custom apps for you? Not just snippets of code, but **fully functional apps** tailored to your exact needs—whether it's an email organizer, a project tracker, or even a lightweight CRM.

That's the future we're heading toward. LLMs, like ChatGPT, can already write code. But as they get trained on more open-source repositories, they'll move beyond coding assistance to **app generation**. Think of it like this: instead of downloading an app, you'll just **ask your LLM to build one**. Need a tool to manage your files? Just say the word. Want to integrate it with your personal or work systems? The LLM handles it.

For companies, it's even more powerful. Businesses won't need to buy off-the-shelf software anymore. They'll **build their own apps on demand**, fully integrated with their existing CRM, ERP, or other platforms. Imagine a company needing a small custom CRM for a specific project. Instead of spending weeks (and thousands of dollars) on development, they'll simply prompt their LLM to build it in hours.

This shift is already starting, but the real transformation will come as **developers fine-tune LLMs with open-source software**. Instead of writing code from scratch, they'll feed LLMs with robust, well-structured projects. These models will not only **replicate functionality** but also **optimize architecture, improve security**, and **customize design**—all tailored to the user's needs.

In this new world, the traditional **"one-size-fits-all" software model is dead**. We're moving toward **personalized, LLM-built apps**, where users—not vendors—define what software looks like. **Open-source will no longer be about code access; it'll be about code intelligence**. And those who don't adapt to this trend—whether app developers, software vendors, or platform providers—risk being left behind.

The future is clear: **software will be built by users, for users, powered by LLMs trained on open-source knowledge**. It's not just a trend; it's the next evolution of digital transformation.

Open Source as Fuel—and a Red Flag for the Software Industry

For years, open-source software has been a symbol of freedom and collaboration. But let's be honest—**open-source alone doesn't change the game** unless you have the skills to use it. What's really changing the landscape is how **LLMs are learning from open-source code**, not just to write snippets but to **build entire apps**.

But here's where it gets interesting. **LLMs don't just learn by reading code**; they learn through **training on structured libraries and patterns** built by developers. Every time an LLM encounters a well-coded open-source project, it's not just understanding the lines of code—it's absorbing the **architecture, design patterns, and workflows** that make that software work.

Now, here's the catch:

- **Big open-source projects**, like CRM or ERP systems, aren't easy for today's LLMs to fully digest. The complexity of these platforms—spanning multiple modules, workflows, and databases—goes beyond simple code generation.
- However, the **next generation of LLM training** will focus not just on writing code but on **understanding software architecture** itself—how components interact, how workflows are built, and how design choices impact functionality.
- In fact, **LLMs are already halfway there**. Today, they can build workflows and processes based on user needs. What's missing is their ability to **design the architecture and code structure** for large-scale, multi-functional software. But let's be clear—that gap is **closing faster than anyone expected**.

Now imagine this scenario:

- An open-source ERP system has multiple business modules—accounting, procurement, inventory, HR.
- What if you **trained an LLM on just one module** and asked it to **build a better version**? Or even to **design a new module** based on specific business needs?
- The LLM wouldn't just replicate code. It would **restructure the architecture**, optimize the design, and suggest **workflow-based integrations**—essentially creating **custom ERP-like systems** without vendor lock-in.

That's not just a futuristic dream. It's **happening now**, whether software vendors like it or not. **Open-source code is no longer just free software—it's training data for LLMs**. And the more open-source projects feed into these models, the closer we get to a world where **every user or company can build their own software**, tailored to their needs.

For software vendors, that's a **massive red flag**. The very open-source movement that once seemed like an alternative to proprietary platforms is now **fueling the rise of user-driven software development**. Why pay for expensive ERP licenses when an LLM can **build a custom version** using open-source blueprints?

In short, **open-source is no longer the product—it's the foundation**. And as LLMs get smarter, the line between user and developer will blur. We're not just talking about small apps anymore. We're talking about **LLM-generated enterprise-grade software**, built by users, for users, without relying on traditional vendors.

The Rise of Multi-Software Platforms: LLMs as Digital Architects

We're entering an era where software will no longer come in pre-packaged solutions from vendors. Instead, **LLMs will act as digital architects**, building multi-functional platforms based on user needs. These platforms won't be monolithic applications but **custom-built ecosystems**, where each app, feature, and workflow is tailored to the individual or company.

Here's the big shift:

- Today, most apps—whether it's a CRM, ERP, or project management tool—are built as **single, isolated platforms** with rigid structures.
- LLMs, however, don't think in terms of apps—they think in **workflows and processes**. If an LLM understands how a task flows from one step to the next, it can **generate software that mirrors those processes**, no matter how many systems are involved.
- **Example:** Instead of buying a separate CRM, project tracker, and invoicing tool, an LLM could build a **single, connected app** that combines the features you actually use, while ignoring unnecessary complexity.

But here's where it gets even more powerful: LLMs won't just copy existing software—they'll improve it.

- Imagine feeding an LLM the open-source code for a CRM module. The LLM wouldn't just recreate it. It would **analyze the architecture**, identify **bottlenecks**, suggest **better design patterns**, and **optimize the code** for efficiency and security.
- Need the same functionality but in a **different user interface**? Just ask.
- Want to **integrate with an existing ERP**? The LLM can adapt the API connections and ensure smooth data flow.

This isn't just hypothetical. **Big players like Microsoft and Salesforce are already embedding AI into their platforms**, allowing users to build workflows without writing code. But here's the difference: **those platforms are still vendor-controlled**. The future LLM-driven platforms will be **user-controlled**, where you decide what software looks like, how it behaves, and how it evolves.

And it's not just for small apps. **Multi-functional platforms** will emerge from LLM-driven development, where:

1. **Features come together like building blocks**, combining CRM, project management, accounting, and more.
2. **Workflows drive the architecture**, not the other way around.
3. **Customization happens in real-time**, based on user prompts, not developer roadmaps.

This approach will **shrink development cycles** from months to days and **reduce costs** dramatically. Why pay for an off-the-shelf solution when you can build a custom platform for your exact needs?

For software vendors, this is another **red flag**. Users won't just build micro-apps—they'll build **full-featured platforms**, bypassing traditional software ecosystems entirely. **The future isn't about apps. It's about workflows powered by LLM-driven architectures.**

In short, **LLMs are becoming the new software developers**, not just for writing code but for designing entire platforms. The more open-source code they train on, the smarter they become. And the smarter they become, the less users will need traditional vendors.

From Users to Builders: How LLMs Empower Individuals and Enterprises

The shift from traditional software to LLM-driven platforms doesn't just benefit developers—it **empowers everyone**, from individual users to enterprise-level organizations. The line between “**software user**” and “**software builder**” is disappearing, and LLMs are making it possible for anyone to build exactly what they need, without relying on vendors or developers.

Here's how it's unfolding:

1. Individuals: Personal Apps Built on Demand

For individual users, **personal LLMs will become app builders**, transforming ideas into functional tools in minutes.

- Need an app to **track personal expenses**? Just ask.
- Want to **organize your files** based on project names and deadlines? The LLM will write the code, build the interface, and connect it to your cloud storage.
- Planning a **fitness tracker** with personalized reminders and progress charts? Your LLM can build one that syncs with your calendar and wearable device.

The key here is **simplicity**. Users won't need coding skills or low-code platforms. They'll just **describe the app they need**, and the LLM will handle the rest—**design, code, deployment, and even maintenance**.

It's like having a **personal software developer** on standby, 24/7.

2. Businesses: Enterprise Software Without Vendors

For enterprises, the impact is even bigger. Companies spend millions on **off-the-shelf software**, much of which is bloated with unnecessary features. With LLMs, they can:

- **Build custom business apps** tailored to specific workflows, skipping the vendor's roadmap entirely.
- **Modify existing platforms** like ERP or CRM systems, adding or removing features as business needs change.
- **Integrate workflows across departments**, creating seamless systems without juggling multiple platforms.

Example:

Imagine a company that needs a **custom procurement system** that integrates with their ERP but adds AI-driven supplier recommendations. Instead of buying an expensive add-on from the ERP vendor, they can **ask their LLM to build the module**, fine-tuned to their specific process.

Or think about **small businesses** that can't afford enterprise-grade software. With an LLM trained on open-source ERP modules, they could build their own **lightweight ERP system**, covering accounting, HR, and inventory management—all tailored to their exact needs.

3. Developers: From Coders to Architects

For developers, this shift doesn't mean extinction—it means **evolution**. Instead of spending hours writing code from scratch, they'll focus on:

- **Training LLMs** with high-quality open-source code.
- **Building app templates** that users can customize.
- **Designing architectures** for multi-functional platforms.

In fact, developers will become **LLM architects**, guiding the models to produce **cleaner, more efficient code** while ensuring **security and scalability**.

4. Software Vendors: Adapt or Get Left Behind

For traditional software vendors, this trend is a **wake-up call**. The rise of LLM-driven software means:

- **Vendors can't rely on licensing models** anymore—users will build their own apps instead.
- **Customization will be user-driven**, not vendor-controlled.
- **Features will be modular**, and users will pick what they need, rather than paying for bloated platforms.

Some forward-thinking vendors will adapt by offering **LLM-compatible modules**, but those who ignore this shift risk becoming obsolete.

The bottom line?

LLMs are turning everyone into a software builder. Whether you're an individual user, a small business, or a global enterprise, you'll no longer be stuck with off-the-shelf solutions. You'll build exactly what you need, when you need it, at a fraction of the cost.

We're not just talking about **apps for tasks** anymore. We're talking about **software ecosystems**, built by users, for users. The future of software isn't in app stores—it's in **your LLM-powered workspace**.

LLM App Stores: The New Marketplace for Personalized Software

Imagine a world where you don't search for apps—you just **build them on the fly**. That's exactly where we're heading with **LLM-powered app stores**. But here's the twist: these stores won't just offer pre-built apps like today's app marketplaces. Instead, they'll provide **LLM-generated apps**, tailored to each user's personal or business needs.

In fact, I've already discussed this concept in my previous article, [The Rise of the LLM Store](#). Back then, the idea seemed futuristic. Today, it's becoming more real than ever.

Here's how it will work:

1. **Personalized App Generation:** Instead of downloading a one-size-fits-all app, you'll prompt your LLM to build an app for a specific task—whether it's project tracking, file management, or customer follow-ups.
2. **Modular App Components:** The app store won't just have complete apps. It'll also offer **modular features**—like a task manager, invoice generator, or scheduling tool—that users can plug into their LLM-generated platforms.
3. **Customizable Templates:** Developers and companies will upload **custom app templates**, allowing users to **generate tailored apps** with a few prompts.
4. **Compatibility with Personal LLMs:** As I've mentioned in my [On-Premises Personal LLMs](#) article, each user will soon have their own LLM. The LLM app store will ensure that apps are **compatible with users' private LLM environments**.

What Does This Mean for Users and Businesses?

- **For individuals:** You'll no longer depend on third-party apps. Need a travel planner? A personal finance tracker? A study organizer? Just ask your LLM to build it using app store templates.
- **For businesses:** Companies can build custom tools without hiring developers or buying expensive software. A sales team, for example, could prompt their LLM to build a **mini CRM** tailored to their workflow.
- **For developers:** The app store will be a **marketplace for LLM-compatible templates**, where developers create and share app structures, workflows, and features.

The End of Traditional App Stores?

The LLM app store won't replace existing platforms overnight, but the shift is inevitable. Why pay for bloated software when you can **generate lightweight, custom apps** on demand?

And it doesn't stop at apps. The same model could apply to **enterprise software, APIs, and even data workflows**, all personalized and built within the user's private LLM ecosystem.

The **app store of the future** won't sell apps. It'll sell **intelligent app-building frameworks**, empowering users to create tools they actually need—without coding, without subscriptions, and without vendor lock-in.

The End of One-Size-Fits-All Software: How LLMs Are Disrupting the Software Industry

The rise of LLM-driven app creation is more than just a technological shift—it's **reshaping the entire software landscape**. As users and businesses gain the power to build their own apps using LLM-generated templates and workflows, the traditional model of buying pre-packaged software from vendors is rapidly becoming outdated.

This transformation ties directly to what I've previously discussed in my article, [Four Types of LLM-Driven Apps: Building a New Era of User-Centric Software](#). As LLMs become more capable, the software landscape will increasingly revolve around these four types of apps:

1. Pre-Built LLM Apps (Vendor-Driven)

These are the **traditional apps** we know today—platforms like CRMs, ERPs, or project management tools with **LLM add-ons**. Vendors integrate AI features, but the core product remains under their control.

However, with the rise of **open-source LLMs** and **personal LLM ecosystems**, these pre-built apps will become **less attractive**. Why pay for software packed with features you don't need when your LLM can build a **customized version** from open-source templates?

2. LLM-Assisted Apps (User-Enhanced)

These are **existing apps enhanced by LLMs**, where users can **automate workflows, generate content, or analyze data** within the app itself. Think of AI-powered Excel functions or CRM chatbots.

The challenge for vendors here is **maintaining value** when users can **replicate these enhancements** through personal LLMs. With open-source training data, users will increasingly **build their own LLM-powered apps**, making vendor-driven enhancements less relevant.

3. LLM-Generated Apps (User-Built)

This is where the **real disruption happens**. LLMs can already generate **small, task-specific apps** on demand. As LLMs become more sophisticated—fueled by open-source libraries—they'll start building **multi-functional platforms**, not just standalone tools.

Example:

A company could train its LLM on **open-source ERP modules**, then ask the LLM to build a **customized ERP system**—covering accounting, HR, and inventory—**without paying for vendor licenses**. This type of user-built software will dominate in the future, giving users **full control over features, workflows, and design**.

4. LLM-Native Apps (Fully AI-Driven)

These apps are **built entirely by LLMs**, without traditional codebases. They're **dynamic, modular, and continuously evolving**, based on user prompts and feedback.

Example:

A sales team could ask their LLM to build a **custom lead tracker** that integrates with their existing CRM and email

platform. The app would **adapt in real-time**, adding new fields, automating workflows, and even suggesting better processes—all without developer involvement.

The Open-Source Factor: Accelerating the Shift

What's fueling this transformation? **Open-source code and LLM training.**

- As **open-source platforms** become training data for LLMs, users will build **customized apps** that match or surpass vendor-built solutions.
 - **Pre-built apps** will lose ground to **user-generated platforms**, while **modular templates** from open-source projects will power LLM-generated apps.
 - Even **enterprise-grade platforms** like ERP and CRM will become **user-driven ecosystems**, where LLMs design architecture, build workflows, and maintain code.
-

A Clear Choice for Software Vendors: Adapt or Become Obsolete

For traditional software vendors, the message is clear:

1. **Adapt:** Shift from selling full platforms to offering **LLM-compatible templates, APIs, and app modules**.
 2. **Empower users:** Let customers **build and customize features** within their personal LLM ecosystems.
 3. **Focus on value:** Provide **premium templates, advanced workflows, and LLM training services**, rather than locking users into rigid platforms.
-

Ultimately, **the software industry is evolving**, and LLMs are at the center of it. Whether through open-source training data, user-built platforms, or modular app stores, the future of software isn't **something you buy—it's something you build**.