wavemaker

Converting Intent into Code

Custom app development has been the Achilles Heel of low-code platforms. Its neither cookie cutter nor simple. But the infusion of AI is a new opportunity to conquer the problem of variety in long-lived custom-made apps. The AutoCode initiative gives professional developers one more reason to adopt WaveMaker in the enterprise.

wavemaker

ĀutoCode

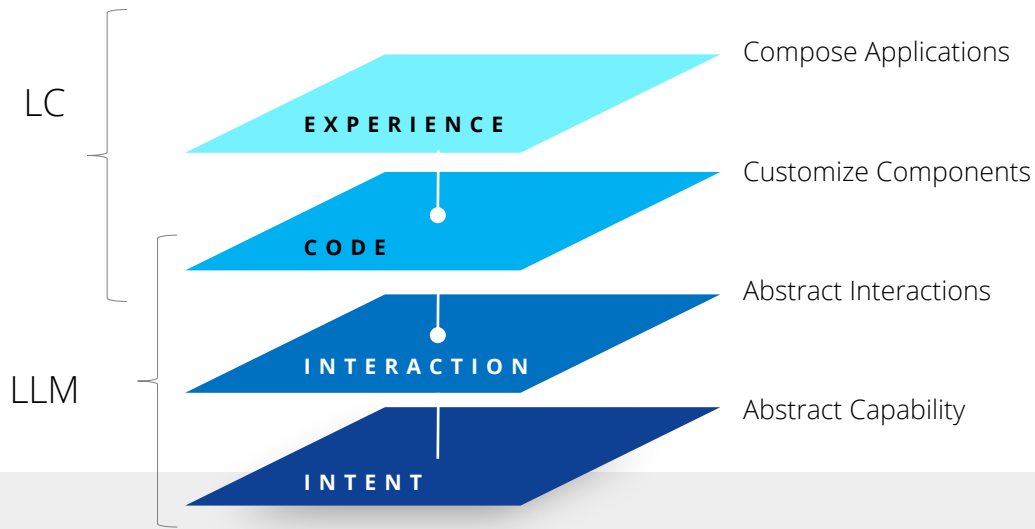## WAVEMAKER'S MISSION STATEMENT FOR AI IN LOW-CODE

Enable multi-disciplinary teams to lower complexity and improve efficiency of converting business intent to working code and build in the shortest time.

Mid-2023, McKinsey declared, "Generative AI is poised to unleash the next wave of productivity." They estimate that Generative AI could add $4 trillion, increasing the impact of AI by up to 40%. One of the top areas in which AI can deliver value is pegged to be software engineering. On the other hand, Gartner puts it right at the top of the peak of inflated expectations, predicting a trough of disillusionment is soon to follow.

We have yet to learn about the grand market value that AI can deliver, but the use cases are undeniable. At WaveMaker, our business is to reduce the grunt work involved in writing software applications. AI opens up possibilities for doing this more pervasively and effectively.

BETTER ABSTRACTION, HIGHER GROUND

Right off the starting block, AI allows software designers to create higher levels of abstraction. Typically, the UI widget library — tables, lists, tabbed panels, what have you — is the first rung of abstraction. With AI inside the WaveMaker low-code platform, we are leap-frogging multiple levels up to convert the intent in the user's mind to working code in a few clicks. Instead of thinking about tables and buttons,

LC

EXPERIENCE — Compose Applications

CODE — Customize Components

INTERACTION — Abstract Interactions

LLM

INTENT — Abstract Capability

developers are free to think about intent or purpose — for instance, a leave management app with approvals and access control.

With AI inside WaveMaker trained on widgets and prefabs (basically packaged business capabilities), developers can state this intent in natural language – in plain English – and swiftly get working code in return. This code conforms to open standards that form the foundation of WaveMaker. The generated UI can be refined by developers using WaveMaker Studio.

WaveMaker Studio provides an intuitive drag-and-drop environment that signifies low-code app development models. With added AI, developers can focus sharply on business scenarios and user interactions that can be instantly prototyped or developed. With such capability, developers get a headstart when writing custom applications that require them to think of unique use cases beyond the cookie-cutter.

This sounds like a personal productivity win, which it is. However, building custom applications is an iterative and collaborative effort among various engineering teams, UX designers, and business analysts.
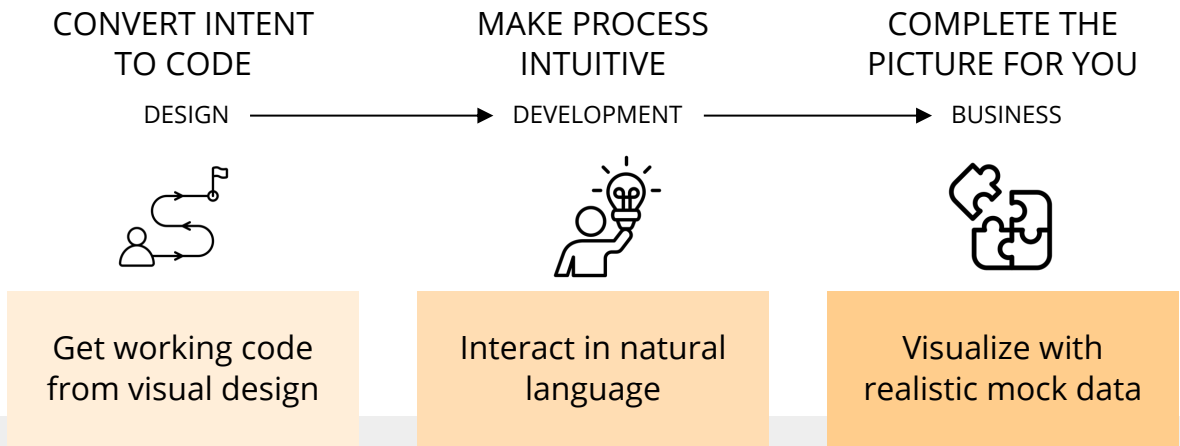
WaveMaker's focus on reducing grunt work and producing pixel-perfect apps using AI leads to the twilight zone between the two worlds of design and development, two semantically different worlds. Closing the design-code gap is the first step from intent to code.

Individual acceleration does not add up to overall delivery acceleration or team productivity in such environments. Anyone familiar with optimization theory will know what we are saying. Project delays and quality failures still happen. There's a ceiling on low-code acceleration when building custom applications. So, how can AI be used to break through? How can it increase the value from each iteration where tasks are owned by multiple teams?

TWILIGHT ZONE: BRIDGING TWO DIFFERENT WORLDS

The ease of generating code expands the use of the development platform itself and raises the expectations of developers. A significant area of collaborative software development needs to be addressed as platforms scramble to add AI capability, most of it for increasing individual developer productivity. In large collaborative enterprise app projects, iteration cycles are still very long, frustrating and riddled with losses in translation among various teams. In particular, the UI cycles between designers and engineers.

Perfecting the UI is often so tedious that it leads to sub-optimal user interfaces. Iterations involving hand-offs between design and UI teams

wavemaker

CONVERT INTENT TO CODE	MAKE PROCESS INTUITIVE	COMPLETE THE PICTURE FOR YOU
DESIGN	DEVELOPMENT	BUSINESS
Get working code from visual design	Interact in natural language	Visualize with realistic mock data

result in the loss of output quality, generating a long list of bugs. Then, someone needs to manage a list of issues, prioritize them, allocate work, monitor progress, etc. This is the real grunt work. By making collaboration seamless, AI can eliminate this grunt work, deliver higher-quality software, and add value more pervasively.

WaveMaker's focus on reducing grunt work and producing pixel-perfect apps using AI has led us to the twilight zone between the two worlds of design and development, two semantically different worlds. The developer world is more deterministic and syntax-driven, while the design world is largely heuristic and imaginative. This difference is accentuated by how the tools and processes work. WaveMaker started with Figma, the popular UI design and prototyping tool to address this.

The WaveMaker promise to Figma users is wireframes can be seamlessly converted to near-perfect working app code — not just plain HTML pages with CSS, but a demo-ready app populated with mock data+. Final changes and refinements to the app can be made in the existing WaveMaker Studio. Popular Figma UI libraries like Material Design collection can be swiftly converted by WaveMaker into

+ New UI screens are difficult to review without real world data in them. WaveMaker Mockingbird is a platform utility that generates and connects mock data based on the API signatures.

Building custom applications is an iterative and a multi-disciplinary effort. This puts a ceiling on low-code acceleration. So, how can AI be used to break through? How can it increase the value from each iteration where tasks are owned by multiple teams?

appropriately themed apps. Further down the road, WaveMaker will allow business users to preview and refine the Figma output in WaveMaker using natural language inputs.

Applying AI this way, WaveMaker facilitates better collaboration among multi-disciplinary teams, resulting in unprecedented iteration velocity, compounding with each iteration. The ability to iterate quickly dramatically reduces time to market while keeping the final product close to the intent — accuracy and speed together.

LOW CODE'S ACHILLES HEEL: CUSTOMIZATION

With every iteration, more gets done, and there is less frustration, less chasing of bugs, and higher chances of being pixel-perfect. But the challenge does not end here. While the design-to-code gap is significant, it isn't the only obstacle to engineering efficiency. The proverbial Achilles Heel for most low-code platforms is custom app development or customization, in general.
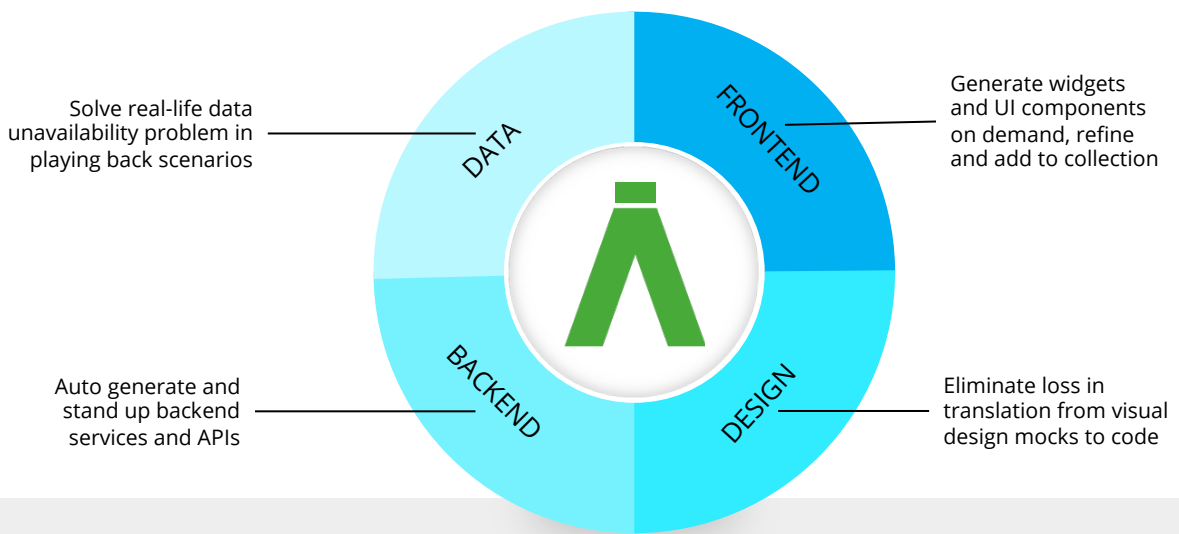
Apps come in various flavors. So do app developers. Three mainly. First is the developer who builds apps quickly using cookie-cutter components with some mild customization. Second, is the developer

CUSTOM APP CHALLENGES	ENGINEERING CHALLENGES	LOW-CODE SOLUTIONS	AI INFUSION IN LOW CODE	IMPACT OF AI IN LOW-CODE
Unrealized UX goals	Non-fungible skillsets	Complementary developer aids	Lossless iterations	**ACCURACY**
Slow response to demand	Code quality & iterations	Visual studio with code generation	Predictive code gen	**ACCELERATION**
Bespoke requirements	Inadequate componentry	Build own components	Just-in-time customization	**DIGITIZATION**

who completely trusts and delegates most of the work to generative AI models. Both cover use cases that are typically short-lived or internal. Third is the developer who builds custom applications with bespoke features, rich user interactions, and enterprise-grade class. Such apps are largely still hand-coded, outside of using open-source libraries.

Unlike most AI-driven low-code products today, WaveMaker doesn't strongly lean into any domain or industry. WaveMaker is not using AI to create a product for finance or manufacturing. Instead, a "horizontal AI" strategy allows it to generate UI specific to a vertical without building the whole product for a niche. Instead, it takes a horizontal approach that addresses the everyday needs of professional developers hand-crafting multivarious applications. Professional developers rely on the power of abstraction. They want the ease of low code with the precision and control of custom coding. This is precisely what AI-infused low-code enables.

Customization can also make the code vulnerable, which makes enterprises wary. There are regulatory compliances, security standards, accessibility and various other aspects that define an enterprise-grade app. This needs strong guardrails to be set in place

Solve real-life data unavailability problem in playing back scenarios

DATA

FRONTEND

Generate widgets and UI components on demand, refine and add to collection

Auto generate and stand up backend services and APIs

BACKEND

DESIGN

Eliminate loss in translation from visual design mocks to code

while the software is being developed so that product managers don't pay the penalty later. Thankfully, unlike design, these are deterministic and easy to set rules for.

With AI, it just gets better. Developers have the freedom to create their abstractions using natural language inputs. Such abstracted components generated using AI prompts already embody best practices and can easily be further refined and assembled into applications in WaveMaker. What makes the WaveMaker platform developer-friendly is its architecture. It has the following cornerstones: an open standards technology stack that makes it investment-proof, real extendable generated code that allows customization, and a standard software development model that fits in with how most delivery teams operate.

FROM CLEAR INTENT TO CLEAR CODE

In the end, what we don't imagine happening is the use of large language models to create cookie-cutter UI all the time across all teams. In our experience working with product ISVs and enterprises alike, we have learnt that teams want to iterate because it helps clarify their intent. During that process, there is a need – not just for speed,

ease, and accuracy – but a space for creating higher business value. That's where we see the combination of low–code and AI winning.

By adapting to how large enterprise development teams build their apps, WaveMaker is already the platform of choice for building long–lived customer applications. The WaveMaker AutoCode initiative infuses AI to enable creative exploration and innovation in enterprise software development. Watch this space.

FOR MORE INFORMATION ON WAVEMAKER LOW-CODE PLATFORM, VISIT WWW.WAVEMAKER.COM.

WAVEMAKER INC. | DALLAS TX | HYDERABAD INDIA