



MASSIMA FROG

ARQUITETURA DE SOFTWARE NA ERA DA INTELIGÊNCIA ARTIFICIAL

COMO GOVERNAR SISTEMAS CADA VEZ MAIS COMPLEXOS EM UM MUNDO DE CÓDIGO GERADO POR MÁQUINAS

Fausto Bacchi Neto | março-2026

INDICADORES DO MERCADO
DE DESENVOLVIMENTO
COM IA

60%

DOS DESENVOLVEDORES
JÁ UTILIZAM IA NO
DESENVOLVIMENTO

30% - 45%

DO CÓDIGO PRODUZIDO
HOJE JÁ É ASSISTIDO
POR IA

90%

DAS EMPRESAS
FORTUNE 100 USAM
FERRAMENTAS DE
CODIFICAÇÃO COM IA

Até 55%

DE GANHO DE
PRODUTIVIDADE COM
ASSISTENTES DE
PROGRAMAÇÃO

EXECUTIVE BRIEF

Na era da Inteligência Artificial, o custo de produzir código caiu drasticamente. O custo de compreender sistemas complexos não.

A aceleração do desenvolvimento de software na era da Inteligência Artificial

Nos últimos anos, a **inteligência artificial** começou a transformar profundamente a forma como software é desenvolvido. Ferramentas baseadas em IA — como **copilots**, **assistentes de programação** e **agentes de desenvolvimento** — estão reduzindo drasticamente o esforço necessário para produzir código funcional. O que antes exigia horas ou dias de trabalho humano agora pode ser gerado em minutos com o apoio de modelos de linguagem e automação.

Estudos recentes indicam que **mais de 60% dos desenvolvedores profissionais já utilizam ferramentas de IA** em alguma etapa do processo de desenvolvimento, enquanto uma parcela adicional planeja adotá-las em breve. Em muitos ambientes corporativos, estimativas apontam que entre 30% e 45% do código produzido atualmente já conta com algum nível de assistência de IA.

Essa transformação não se limita a equipes experimentais ou startups de tecnologia. Pesquisas de mercado indicam que aproximadamente **90% das empresas da Fortune 100 já utilizam ferramentas de desenvolvimento assistido por IA**, e o impacto na produtividade é significativo. Desenvolvedores que utilizam esses recursos conseguem concluir tarefas até 50% mais rapidamente, acelerando ciclos de entrega e ampliando a capacidade de inovação das equipes.

QUOTES INTERESSANTES SOBRE DESENVOLVIMENTO E ARQUITETURA DE SOFTWARE



“ *The hottest new programming language is English.* ”

ANDREJ KARPATY

AI Researcher
Former Director of AI
— Tesla / OpenAI

(Post on X (Twitter),
Jan 24, 2023)



“ *Within five years, 95% of code may be generated by AI.* ”

KEVIN SCOTT

CTO
Microsoft

(Interview on the
20VC podcast)

O resultado é uma **mudança estrutural na engenharia de software**. A produção de código está se tornando cada vez mais rápida, acessível e abundante. Em um cenário onde agentes de IA conseguem gerar grandes volumes de código em pouco tempo, a forma como sistemas evoluem também começa a mudar — abrindo espaço para novos desafios na gestão da complexidade dos sistemas.



Quando produzir código fica barato, a complexidade cresce

A aceleração do desenvolvimento impulsionada pela inteligência artificial trouxe **ganhos claros de produtividade**. Equipes conseguem gerar novas funcionalidades mais rapidamente, experimentar soluções com maior velocidade e reduzir o tempo necessário para transformar ideias em software funcional.

Mas essa nova realidade também introduz um paradoxo. À medida que produzir código se torna mais fácil e rápido, **o volume de software produzido cresce exponencialmente**. Mais funcionalidades, mais serviços, mais integrações e mais dependências passam a fazer parte dos sistemas.

Esse crescimento nem sempre ocorre de forma linear ou totalmente planejada. **Pull requests se multiplicam**, novos componentes surgem continuamente e bases de código crescem em ritmo acelerado. Com o apoio de agentes de IA, pequenas mudanças podem ser implementadas rapidamente — mas seus efeitos podem se propagar por diferentes partes do sistema.

O resultado é que, enquanto a capacidade de produzir código aumenta, compreender o comportamento e a estrutura dos sistemas torna-se cada vez mais desafiador. **O desafio deixa de ser apenas escrever software** e passa a ser entender o sistema complexo que esse software cria.

Entropia arquitetural: um problema antigo em uma nova escala

Sistemas de software complexos têm uma característica bem conhecida na engenharia: ao longo do tempo, sua estrutura tende a se tornar mais difícil de compreender e evoluir. À medida que novas funcionalidades são adicionadas, dependências surgem, integrações se multiplicam e decisões técnicas se acumulam.

Esse fenômeno é frequentemente descrito como **entropia arquitetural** — a **tendência natural de sistemas evoluírem** de forma gradual para estruturas mais complexas, menos previsíveis e mais difíceis de modificar.

QUOTES INTERESSANTES SOBRE DESENVOLVIMENTO E ARQUITETURA DE SOFTWARE



“ Architecture is about the important stuff. Whatever that is. ”

MARTIN FOWLER

Chief Scientist -ThoughtWorks
Author of Refactoring

(Who Needs an Architect?,
IEEE Software.)



“ Architecture represents the significant design decisions that shape a system, where significance is measured by cost of change. ”

GRADY BOOCH

IBM Fellow
Co-creator of UML

(Object-Oriented
Analysis and Design
with Applications)

Na prática, muitas equipes reconhecem esse problema por meio de um sintoma recorrente: mudanças aparentemente simples passam a gerar **efeitos inesperados em outras partes do sistema**. Um pequeno ajuste em um serviço pode provocar falhas em componentes que, à primeira vista, pareciam não ter relação direta com aquela alteração.

Com a aceleração da produção de código impulsionada pela inteligência artificial, esse fenômeno tende a se intensificar. À medida que novas mudanças são introduzidas com maior frequência e em maior volume, compreender o impacto arquitetural dessas alterações torna-se um desafio cada vez maior para equipes de engenharia.

O novo gap da engenharia de software: governança arquitetural

Tradicionalmente, a engenharia de software desenvolveu diversas práticas para manter a qualidade dos sistemas ao longo do tempo. Revisões de código, testes automatizados, integração contínua e pipelines de entrega ajudam equipes a garantir que novas mudanças funcionem corretamente e não introduzam erros imediatos.

No entanto, essas práticas foram concebidas principalmente para verificar o comportamento do código, não necessariamente o impacto dessas mudanças sobre a arquitetura do sistema como um todo. Em muitos casos, **revisões continuam ocorrendo linha por linha de código**, sem oferecer uma visão clara de como uma alteração afeta a estrutura geral do software.

À medida que o desenvolvimento se acelera com o apoio de inteligência artificial, esse descompasso começa a se tornar mais evidente. Equipes conseguem produzir código em um ritmo cada vez maior, mas a capacidade de compreender o **impacto arquitetural dessas mudanças** nem sempre evolui na mesma velocidade.

O resultado é um **novo tipo de lacuna na engenharia de software**: o gap entre a velocidade de evolução do código e a capacidade de **governar a arquitetura dos sistemas**. Em ambientes onde mudanças acontecem com grande frequência, manter a integridade arquitetural passa a exigir novos mecanismos de visibilidade e controle.

Visibilidade arquitetural como nova capacidade da engenharia

À medida que sistemas evoluem em velocidade cada vez maior, compreender a arquitetura de software deixa de ser apenas uma atividade pontual de documentação e passa a se tornar uma capacidade operacional contínua das equipes de engenharia.

Em ambientes onde novas mudanças são introduzidas constantemente, **entender como componentes se relacionam**, como dependências se formam e como modificações afetam o sistema como um todo torna-se essencial para preservar a integridade arquitetural.



“Software development is an attempt to understand and influence the world through symbols and language.”

KLAUS WUESTEFELD

CEO

Jux.House

(Agile Trends)

Nesse contexto, a arquitetura precisa deixar de ser apenas um conceito abstrato descrito em diagramas estáticos e passar a ser uma **representação viva do sistema**, capaz de refletir sua estrutura real e sua evolução ao longo do tempo.

Isso exige um **novo nível de visibilidade arquitetural** — a capacidade de enxergar, compreender e acompanhar o impacto estrutural das mudanças introduzidas no software. Sem essa visibilidade, governar a evolução de sistemas complexos torna-se progressivamente mais difícil à medida que o volume de código cresce.

Governança arquitetural em escala

A aceleração do desenvolvimento impulsionada pela inteligência artificial está redefinindo a dinâmica da engenharia de software. À medida que produzir código se torna cada vez mais rápido e acessível, manter a **integridade e a coerência arquitetural dos sistemas** torna-se um desafio cada vez mais estratégico para organizações digitais.

Nesse novo cenário, práticas tradicionais de engenharia continuam essenciais, mas já não são suficientes para lidar com o volume e a velocidade das mudanças introduzidas em sistemas modernos. Preservar a **saúde arquitetural dos sistemas** passa a depender também da capacidade de visualizar, compreender e governar sua evolução estrutural ao longo do tempo.

Ferramentas capazes de tornar a arquitetura visível e acompanhar sua evolução permitem que equipes entendam o impacto arquitetural das mudanças, identifiquem padrões emergentes de complexidade e estabeleçam mecanismos de governança mais eficazes para sistemas em constante transformação.

É nesse contexto que soluções como o **BELA — The Browser for Enterprise-Level Architecture**, da Jux, ganham relevância. Ao oferecer uma **visão dinâmica da arquitetura de software** e de sua evolução, o BELA ajuda organizações a manter controle sobre a complexidade crescente de seus sistemas, permitindo que equipes aproveitem os ganhos de velocidade proporcionados pela inteligência artificial sem perder a governança da arquitetura.

Saiba mais em:



folheto do **BELA**



contact-us@massimafrog.com



www.massimafrog.com



[linkedin.com/company/massimafrog/](https://www.linkedin.com/company/massimafrog/)



MASSIMA FROG