

The background of the entire page is a grayscale image of an hourglass. The top bulb is larger and contains a significant amount of sand, while the bottom bulb is smaller and has a thin stream of sand falling from its neck. The lighting creates highlights on the glass surfaces, giving it a three-dimensional appearance.

softo

IA no  
*desenvolvimento*  
de software

Como empresas estão entregando mais

rápido sem comprometer a qualidade.

## SEÇÃO 01

# A fila que não para não é culpa do time

Você já deve ter ouvido alguma versão dessa conversa:



QUANDO ISSO ENTRA EM PRODUÇÃO?

ESTÁ NO BACKLOG. ENTRA NO PRÓXIMO CICLO.



ISSO FOI PEDIDO HÁ TRÊS MESES.

EU SEI. MAS TEM DEZ COISAS NA FRENTE.



Esse diálogo não é sinal de time ruim. É o sintoma natural de um modelo que não foi construído para o ritmo que o negócio exige hoje.

Durante décadas, desenvolver software significava levantar requisitos, estimar, dividir em tarefas e entregar. O modelo funcionava quando o mercado mudava em trimestres. **Hoje muda em semanas.** Esse modelo ficou para trás.

O resultado é conhecido: quanto mais a empresa cresce, mais demandas entram no backlog. A fila nunca zera. O time está sempre sobrecarregado. Tecnologia corre atrás do negócio, sempre.



**Contratar mais desenvolvedores resolve o sintoma por um ciclo. Depois a fila volta maior.**

O que está mudando, para as empresas que perceberam, não é o tamanho do time. **É a lógica que organiza o trabalho.**

E a IA entrou nessa equação. Mas não da forma que a maioria imagina.

## SEÇÃO 02

# IA não é o desenvolvedor escrevendo código mais rápido

Essa é a primeira coisa que a maioria das empresas entende errado sobre IA no desenvolvimento.

Copilot, Cursor, Claude no terminal, são ferramentas úteis. Um desenvolvedor com IA escreve mais rápido, erra menos na sintaxe, encontrar bugs com mais agilidade. Isso é real e já gera ganho.

**Mas isso é o uso mais superficial possível.**

O impacto estrutural de IA num time de desenvolvimento aparece quando ela deixa de ser uma ferramenta individual e passa a ser parte do fluxo de trabalho. Quando entra como elemento principal da fase de codificação, não só no momento em que o dev está digitando. É uma inversão de posicionamento. Não se trata da IA ajudar o dev a escrever código, mas sim da IA ser o codificador. E também o testador, documentador, etc.

Na prática, isso significa:

Código criado muito mais rápido, com qualidade

Revisão de código feita por IA antes de chegar no humano: o revisor humano recebe código já limpo

Protótipos funcionais em dias: antes de qualquer discussão longa sobre viabilidade

Testes gerados automaticamente a partir das mudanças feitas: cobertura maior sem ninguém precisar escrever caso a cas

Documentação gerada junto com o código: não fica para depois, não some com a rotatividade

Bugs analisados por IA com contexto do histórico do sistema: o desenvolvedor chega na causa, não fica caçando sintoma

**O resultado não é 'o desenvolvedor ficou 30% mais rápido'. É que o mesmo time consegue atacar três vezes mais problemas em paralelo, com mais qualidade em cada um.**

Isso muda o cálculo de capacidade. E muda o que é possível fazer sem contratar.

A Softo observa, nos times que operam com IA integrada ao ciclo completo de desenvolvimento, entregas até 10x mais rápidas em comparação com times que usam IA só como assistente de código. A diferença não está na ferramenta. Está em **como o processo foi desenhado** e qual **nível de autonomia** você dá para as ferramentas.

## SEÇÃO 03

# O que empresas que acertaram nisso têm em comum

Nas empresas que estão usando IA para entregar mais, não é stack, orçamento ou ter contratado o melhor engenheiro de IA do mercado.

Essas empresas que estão mudando o jogo com IA têm quatro coisas em comum:

### **ENTREGAM MAIS VEZES, NÃO SÓ MAIS RÁPIDO**

A cadência de entrega aumentou. Ao invés de um grande lançamento por trimestre, essas empresas estão fazendo pequenas entregas constantemente, às vezes dezenas de entregas por dia, testadas em produção, ajustadas com base em dados reais. Ou seja, o ciclo encurtou de meses para semanas, de dias para horas.

**Consequência direta:** quando uma ideia não funciona, o custo de descobrir isso é baixo. A empresa experimenta mais porque errar cedo e corrigir rápido ficou barato.

## RETRABALHO CAIU E LIBERA TEMPO PARA O QUE IMPORTA

Bug em produção custa caro. Não só o tempo de consertar, custa a confiança do usuário, o tempo do time que poderia estar construindo algo novo, a discussão sobre quem foi responsável.

IA aplicada a testes e revisão reduz o que chega quebrado em produção. O time para de viver apagando incêndio e começa a construir.

## QUALIDADE VIROU PROCESSO, NÃO DEPENDÊNCIA DE INDIVÍDUO

Em times sem IA sistematizada, qualidade depende da atenção de cada pessoa em cada momento. Um desenvolvedor cansado revisa pior. Um onboarding mal feito gera código inconsistente.

Com IA no ciclo de revisão, padrões são aplicados de forma consistente, independente de quem está no time essa semana.

**O efeito composto:** o time entrega mais, com mais previsibilidade, gastando menos tempo em trabalho que não exige julgamento humano. O custo por funcionalidade entregue cai e a capacidade de inovar sobe.

## SEÇÃO 04

# Por que a maioria das empresas não está colhendo isso

Ferramentas de IA estão disponíveis para qualquer empresa com cartão de crédito. A maioria dos desenvolvedores já usa alguma.

## Então por que o backlog continua crescendo?

Porque acesso à ferramenta e uso estratégico são coisas diferentes. A maioria das empresas está no primeiro estágio: cada desenvolvedor escolhe sua ferramenta, usa do jeito que acha melhor, e o ganho de produtividade, quando existe, fica preso no nível individual.

Não vira processo. Não vira previsibilidade. Não aparece nas métricas do negócio.

Há três lacunas que separam quem usa IA para crescer de quem usa IA para 'ser mais eficiente':

## **SABER ONDE IA GERA MAIS VALOR PARA AQUELA OPERAÇÃO ESPECÍFICA**

IA não melhora tudo igualmente. Numa empresa, o maior ganho está em automação de testes. Em outra, em geração de documentação. Em outra, em análise de dados de produto. Sem clareza sobre isso, o esforço se pulveriza e os ganhos ficam marginais.

## **INTEGRAR IA AOS PROCESSOS QUE EXISTEM**

Um workflow de desenvolvimento que foi desenhado para atender um contexto de trabalho onde a codificação era escassa, não pode servir bem a uma abordagem nova e diferente. Plugar uma ferramenta de codificação com IA à um ciclo de desenvolvimento antigo não agrega tanto valor quando repensar o ciclo para o novo cenário.

## **CONNECTAR PRODUTIVIDADE A RESULTADO MENSURÁVEL**

Um desenvolvedor 40% mais rápido é bom. Mas 40% mais rápido em quê? Se o ganho de velocidade está em funcionalidades que não afetam nenhuma métrica do negócio, ele não aparece em lugar nenhum. A maioria dos times não tem esse elo definido.

Fechar essas três lacunas ao mesmo tempo não é trabalho de ferramenta. É trabalho de modelo operacional.

## SEÇÃO 05

# O que muda quando o foco é o resultado, não a entrega

Existe uma pergunta que a maioria dos times de tecnologia nunca faz de forma explícita:

**Se esse projeto der certo, qual número da empresa vai mudar?**

Parece simples, mas **muda tudo**.

Quando a resposta não existe, quando o projeto é construir uma funcionalidade sem uma métrica clara atrelada, o que acontece depois da entrega? Ninguém sabe se funcionou. O time parte para o próximo ticket. A funcionalidade vai para produção e some na lista de 'coisas que existem'.

Algumas empresas estão organizando tecnologia de forma diferente: cada iniciativa tem um resultado específico, mensurável, acordado antes de começar. O time não entrega código. Entrega o resultado e o código é o meio.

Esse é o modelo por trás dos Outcome Pods da Softo. Um Pod não é uma squad terceirizada que executa o seu backlog. É uma equipe montada em torno de outcomes de negócio, que combina:

Especialistas de tecnologia com experiência em IA aplicada

IA integrada ao ciclo completo de desenvolvimento, não como acessório

Automação conectada aos sistemas de controle

Métricas de sucesso definida antes de começar

	MODELO TRADICIONAL	OUTCOME PODS
Pergunta que organiza o trabalho	"O que vamos construir?"	"Qual número precisa mudar?"
O que o time entrega	Funcionalidades	Resultados mensuráveis
Como IA entra	Ferramenta individual de cada dev	Infraestrutura do processo de entrega
Como sucesso é medido	Entregou no prazo?	Moveu a métrica?
O que acontece com o backlog	Cresce a cada trimestre	Somente funcionalidades que realmente impactam o negócio

O objetivo não é apenas desenvolver software mais rápido.  
**É fazer a empresa chegar mais rápido ao resultado que o software precisa produzir.**

PRÓXIMO PASSO

# Se faz sentido, o próximo passo é uma conversa

Queremos te apresentar os Outcome Pods da Softo. Sem demo padronizada, sem formulário com dez campos. Uma conversa direta sobre o que a sua empresa está tentando resolver e se Outcome Pods faz sentido para isso.

Se sua empresa quer usar IA para desenvolver mais rápido e conectar tecnologia a resultados reais de negócio: o próximo passo não é adotar mais ferramentas.

**CONHEÇA OS OUTCOME PODS DA SOFTO**

**DESCUBRA COMO APLICAR IA PARA**

**ACELERAR RESULTADOS DENTRO DA SUA OPERAÇÃO.**

**softo**

[softo.to/pt-BR/services/outcome-pods](https://softo.to/pt-BR/services/outcome-pods)