



softo

# A nova *economia* do software

Por que a forma como empresas contratam,

desenvolvem e compram software está mudando

e o que isso significa para quem lidera essa decisão.

## SEÇÃO 01

# Cada geração comprou software de um jeito diferente

A indústria de software mudou mais nos últimos três anos do que nos vinte anteriores. Mas essa mudança tem um padrão e entender o padrão ajuda a enxergar para onde estamos indo.

Cada era da tecnologia redefiniu o que significa **'ter software'**. A cada virada, as empresas que entenderam cedo o novo modelo ganharam vantagem sobre as que continuaram operando com a lógica anterior.

## PERÍODO

## ERA

## O QUE MUDOU

ANOS

**80-90**

Software  
como  
licença

Você comprava o produto. Instalava. Usava até a próxima versão, que você também comprava.

ANOS

**2000-10**

Software  
como  
serviço  
(SaaS)

Você assina o acesso. O fornecedor mantém, atualiza, escala. O custo vira previsível. A dependência, permanente.

ANOS

**80-90**

Software  
como  
plataforma  
(PaaS /  
ecossistem  
as)

O produto vira ecossistema. APIs, integrações, marketplaces. A empresa constrói em cima do que outros construíram.

**Agora**

Software  
como  
outcome

A empresa para de comprar software e começa a contratar resultados. A tecnologia é o meio. O número que precisa mudar é o produto.

A linha que conecta essas eras é clara: o software foi se tornando progressivamente menos um ativo e mais um serviço. Menos algo que você possui e mais algo que você acessa. E agora, na próxima virada, menos algo que você usa e mais algo que produz um resultado para você.



**Cada transição deixou para trás empresas que demoraram a perceber que a lógica havia mudado. A transição atual está acontecendo agora.**

Líderes de tecnologia que entenderam essa virada pararam de perguntar 'qual software comprar?' e começaram a perguntar 'qual resultado quero que a tecnologia produza?'"

## SEÇÃO 02

# O que IA mudou na economia do software

A chegada da IA generativa mudou três variáveis fundamentais da indústria de software ao mesmo tempo. Individualmente, cada mudança seria relevante. Juntas, elas redefinem o jogo.

### O CUSTO DE DESENVOLVER SOFTWARE CAIU DRASTICAMENTE

Um desenvolvedor com IA integrada ao processo produz, em termos de volume de código funcional, o que antes exigiria uma equipe maior. Protótipos que levavam semanas saem em dias. Integrações que demandavam meses de desenvolvimento passam a levar semanas.

**A consequência direta:** a barreira para construir software caiu. Qualquer empresa com um modelo claro consegue construir mais, mais rápido, com menos gente.

## **A VELOCIDADE DE CRIAÇÃO AUMENTOU E APROXIMOU QUEM ITERA DE QUEM PLANEJA**

Empresas que operam com ciclos longos de planejamento e desenvolvimento estão sendo ultrapassadas por empresas que testam hipóteses em dias. A capacidade de iterar, lançar, medir, ajustar, tornou-se mais valiosa do que a capacidade de planejar com precisão.

O planejamento de seis meses que produzia um lançamento passou a competir com equipes que planejaram rápido, lançaram, testaram e ajustaram 6x no mesmo período

## **A QUALIDADE DA ENTREGA DEIXOU DE DEPENDER DO TAMANHO DO TIME**

Times menores com IA bem aplicada entregam o que antes exigia equipes maiores. A senioridade que antes era escassa para revisar código, garantir arquitetura e reduzir retrabalho passou a ser amplificada pela IA, não substituída por ela.

O que mudou: o gargalo deixou de ser o número de pessoas e passou a ser a clareza sobre qual resultado o time deve produzir.

**O efeito combinado dessas três mudanças:** o custo de construir caiu, a velocidade de testar subiu, e a automação passou a ser viável em escala. A empresa que souber combinar essas três alavancas ao mesmo tempo está operando em uma categoria diferente das que usam cada uma isoladamente.

## SEÇÃO 03

# O código virou commodity. O que restou como diferencial?

Por muito tempo, ter uma equipe técnica forte era o diferencial. Poucas empresas conseguiam construir software bom. Quem conseguia, ganhava.

IA democratizou a capacidade de construir. Um time médio com IA bem aplicada entrega o que um time excelente entregava sem ela. A vantagem de ter os melhores engenheiros encolheu.



**O código, em si, deixou de ser barreira de entrada.**

O que passou a separar empresas que crescem das que estagnam é outra coisa:

### O QUE ERA O DIFERENCIAL

Ter o software

Velocidade de desenvolvimento

Qualidade do código

Headcount do time técnico

Acesso a ferramentas de IA

### O QUE É O DIFERENCIAL HOJE

Saber extrair resultado do uso do software


Velocidade de aprendizado e iteração

Qualidade da decisão sobre o que e como construir

Estrutura mínima para conectar tecnologia à resultado

Método de trabalho que converte IA em impacto

A empresa que entende isso muda a pergunta: não é mais "como construo software melhor?", agora passa a se perguntar: "qual resultado quero que a tecnologia produza?"



**Execução sempre foi importante. O que mudou é que a capacidade de construir ficou barata. A capacidade de conectar o que foi construído a um resultado de negócio ficou valiosa.**

Integração, velocidade de aprendizado e clareza sobre o que medir passaram a valer mais do que a qualidade técnica do código que virou o mínimo esperado, antes de ser o diferencial.

## SEÇÃO 04

# O surgimento dos sistemas orientados a outcome

Existe uma consequência lógica de tudo o que mudou: se o custo de construir caiu e se o diferencial está em conectar tecnologia ao resultado, então o modelo de organizar desenvolvimento de tecnologia em torno de funcionalidades e projetos ficou obsoleto.

O modelo de funcionalidade tem uma lógica interna: define o que construir, estima o esforço, entrega, passa para o próximo.

Funciona bem quando construir é difícil e caro, quando é preciso preservar o recurso escasso, ou seja, o desenvolvedor. Quando construir ficou mais fácil, o gargalo mudou de lugar.

**O gargalo hoje é clareza sobre qual resultado a tecnologia deve produzir.**

Sistemas orientados a outcome partem do resultado desejado. A pergunta inicial é 'qual número precisa mudar e em quanto tempo?'. O que construir, quais ferramentas usar, como aplicar IA e automação: tudo isso vira consequência de uma resposta clara a essa pergunta.

Esse modelo muda o que o time de tecnologia é responsável por entregar. Em vez de funcionalidades que saem do backlog, o time se torna responsável por um resultado mensurável e a tecnologia é o meio para chegar lá.

**A diferença prática:** no modelo tradicional, o projeto termina quando o código está em produção. No modelo orientado a outcome, o trabalho termina quando o resultado foi alcançado ou se expande quando o resultado pede mais.

Empresas que adotarem essa lógica podem perceber três mudanças consistentes: o backlog para de crescer, o alinhamento entre tecnologia e negócio deixa de ser uma reunião recorrente e passa a ser a estrutura do trabalho, e o custo por resultado entregue cai ao longo do tempo, ao contrário do modelo tradicional, onde tende a subir.

## SEÇÃO 05

# Outcome Pods: o modelo operacional para essa nova lógica

Outcome Pods é como a Softo estrutura sistemas orientados a resultado dentro das operações de uma empresa. A ideia central é simples: unidade de engenharia de software AI-Native em torno de um outcome específico, uma métrica, um objetivo, um número que precisa mudar e dar a essa equipe tudo o que precisa para chegar lá.

Cada Pod combina especialistas de tecnologia com experiência em IA aplicada, automações integradas aos sistemas da empresa e um critério único de sucesso definido antes de começar.

**Outcome Pods opera na intersecção de três forças que IA tornou possível combinar ao mesmo tempo: custo de desenvolvimento, automação inteligente e velocidade de iteração alta.**

Na prática, o Outcome Pods pode ser responsável por reduzir o tempo de onboarding de clientes, automatizar um processo operacional crítico, construir um sistema de geração de pipeline ou entregar inteligência operacional contínua para decisões mais rápidas.

O que define o Pod sempre é o mesmo: um resultado mensurável que a empresa quer alcançar. A tecnologia, qual stack, quais ferramentas de IA, quais automações, é desdobramento desse resultado, nunca o ponto de partida.

### **Por que esse modelo faz sentido agora**

O custo de construir caiu: times menores conseguem entregar mais com IA integrada ao processo

O diferencial mudou de lugar: estar à frente exige velocidade de resultado, não volume de funcionalidades

O mercado não espera mais: empresas que operam com ciclos de entrega longos estão sendo ultrapassadas por equipes que iteram em dias


Nos próximos anos, empresas que continuarem comprando software da forma tradicional, projetos, funcionalidades, backlogs estarão competindo em desvantagem contra empresas que contratam resultados e deixam a tecnologia ser o meio.

PARA FECHAR

# A virada já está acontecendo

Cada vez que a indústria de software mudou de era, havia dois grupos de empresas: as que perceberam cedo e reposicionaram suas operações, e as que continuaram com o modelo anterior até que a desvantagem se tornou grande demais para ignorar.

A transição para sistemas orientados a outcome está acontecendo agora. As empresas que estão construindo essa capacidade hoje não estão se preparando para o futuro, estão ganhando o presente.



**A vantagem competitiva dos próximos anos vai para quem conseguir mover métricas de negócio mais rápido. Outcome Pods é como a Softo constrói essa capacidade dentro das operações de uma empresa.**

Se você quer entender como esse modelo se aplica ao contexto da sua empresa, o próximo passo é uma conversa direta.

**CONVERSE COM A SOFTO SOBRE OUTCOME PODS**

**DESCUBRA COMO OPERAR NO MODELO**

**DE SISTEMAS ORIENTADOS A OUTCOME.**

**softo**

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