

A leitura do artigo *Managing Technical Debt*, de Steve McConnell, permite compreender com clareza a importância do conceito de dívida técnica e seus impactos na gestão de projetos de software. O texto parte da ideia criada por Ward Cunningham, que comparou as decisões de desenvolvimento que priorizam atalhos no presente ao funcionamento de uma dívida financeira: é possível contrair uma obrigação imediata que gera benefícios rápidos, mas, se não for controlada e paga, ela acumula juros que comprometem o futuro da organização. McConnell diferencia dois tipos fundamentais de dívida: a não intencional, resultante de erros, inexperiência ou descuido, e a intencional, fruto de escolhas conscientes para atender prazos, reduzir custos imediatos ou acelerar o tempo de chegada ao mercado. O autor dá maior atenção a esta segunda, destacando que ela pode ser útil em situações estratégicas, desde que seja monitorada e quitada em momento oportuno.

Um dos pontos fortes do artigo é a analogia com dívidas financeiras de curto e longo prazo. Dívidas de curto prazo no desenvolvimento são aquelas contraídas taticamente, geralmente para viabilizar a entrega de uma versão ou produto, devendo ser pagas logo após o ciclo. Já as de longo prazo são assumidas com uma perspectiva mais ampla, podendo permanecer por anos sem gerar grandes impactos, desde que haja clareza e controle. McConnell alerta, entretanto, que há dívidas que nunca compensam: pequenos atalhos acumulados, como variáveis genéricas, código mal comentado ou desvios dos padrões de programação. Esse acúmulo, comparável a uma dívida de cartão de crédito, é difícil de rastrear e cobra um preço elevado em produtividade e qualidade.

Outro aspecto interessante do artigo é o modo como aborda a comunicação da dívida técnica dentro da organização. Para muitos executivos, o conceito é abstrato, e os impactos não são visíveis de imediato. Ao traduzir a questão para uma linguagem financeira — custos, juros, ativos e passivos —, o autor sugere que a conversa se torna mais clara e acessível para líderes de negócio. Exemplos como “40% do orçamento de P&D está sendo gasto em manutenção de sistemas antigos” ou “X milhões de dólares por ano estão sendo consumidos em débitos técnicos” ajudam a tangibilizar o problema e embasar decisões. Além disso, ele recomenda registrar essas dívidas em backlogs de defeitos ou de produto, tratando-as como histórias que precisam ser estimadas e priorizadas junto às demais demandas.

McConnell também enfatiza que a capacidade de assumir dívida técnica varia entre equipes, assim como acontece com crédito financeiro. Times mais maduros, com baixo nível de dívida não intencional, podem arcar com mais obrigações estratégicas. Já equipes que acumulam muitos erros ou retrabalho têm menos margem de manobra. Para mitigar os efeitos, o autor sugere práticas como dedicar parte do ciclo de desenvolvimento exclusivamente à redução da dívida logo após uma entrega, ou incluir

percentuais fixos de esforço em cada sprint voltados a correções estruturais. Essa disciplina evita que a dívida cresça a ponto de consumir a maior parte do tempo da equipe apenas em manutenção.

O artigo apresenta ainda uma contribuição valiosa ao discutir opções intermediárias entre o “caminho bom e caro” e o “atalho rápido e sujo”. Frequentemente, existem alternativas híbridas que permitem acelerar a entrega sem comprometer o futuro de forma tão severa, como criar camadas de abstração que reduzem o custo posterior de substituição. Esse olhar mais nuançado evita que equipes pensem em termos binários, incentivando decisões que equilibram velocidade e sustentabilidade. Ao oferecer exemplos práticos de cálculo de custos imediatos, custos diferidos e juros da dívida, McConnell mostra como um raciocínio estruturado pode orientar escolhas melhores.

Na prática, esse raciocínio é extremamente aplicável ao mercado de tecnologia atual, marcado pela pressão por lançamentos rápidos e pela competitividade global. Empresas que desenvolvem produtos digitais frequentemente enfrentam dilemas entre investir tempo em arquiteturas escaláveis ou priorizar funcionalidades que gerem tração no curto prazo. Em startups, por exemplo, pode fazer sentido adiar práticas mais sofisticadas de engenharia para preservar capital e validar hipóteses de negócio. Entretanto, se essa dívida não for reconhecida e endereçada quando o produto amadurece, o resultado pode ser uma base de código difícil de manter, com alto custo de adaptação e comprometimento da inovação. No caso de grandes empresas de tecnologia, um exemplo prático está nos sistemas legados que sustentam operações críticas: manter código envelhecido gera despesas de manutenção contínuas, mas reescrevê-lo exige investimentos vultosos. A decisão de quando e como pagar essa dívida deve considerar o impacto na velocidade de lançamentos futuros, no risco de falhas e na experiência do cliente.

Uma reflexão prática para o mercado de tecnologia é a necessidade de instituir métricas de dívida técnica tão objetivas quanto as de performance financeira. Assim como CFOs acompanham índices de alavancagem e liquidez, líderes de produto e engenharia poderiam monitorar indicadores de retrabalho, tempo médio de entrega de hotfixes ou proporção do orçamento gasto em manutenção. Esses números criam visibilidade e ajudam a alinhar expectativas entre áreas técnicas e executivas. Outra aplicação é inserir a discussão sobre dívida técnica nos rituais ágeis: cada sprint pode reservar espaço para avaliar novos débitos contraídos, decidir se vale a pena registrá-los como histórias e definir quando serão pagos. Dessa forma, a gestão deixa de ser pontual e passa a ser contínua.

Em síntese, a contribuição de McConnell é mostrar que a dívida técnica não deve ser encarada como um mal absoluto, mas como um instrumento de gestão que pode gerar valor quando utilizado com disciplina e transparência. A chave está em distinguir a

dívida estratégica da dívida irresponsável, tornar os compromissos visíveis para todos os stakeholders e manter a capacidade de pagamento sempre sob controle. No mercado de tecnologia, onde o equilíbrio entre velocidade e qualidade define o sucesso, essa abordagem fornece um quadro mental poderoso para orientar decisões. Aplicar esses conceitos significa evitar que atalhos necessários se transformem em armadilhas, permitindo que empresas inovem com rapidez sem comprometer a sustentabilidade de longo prazo.