

Aluno: Thomás Ramos Oliveira

Gestão de Dívida Técnica: Abordagens Estratégicas para o Desenvolvimento de Software

Introdução ao Conceito de Dívida Técnica

O termo "dívida técnica" foi cunhado por Ward Cunningham para descrever a obrigação que uma organização de software contrai quando opta por uma abordagem de projeto ou construção que é vantajosa no curto prazo, mas que aumenta a complexidade e se torna mais custosa no longo prazo. Assim como a dívida financeira, algumas dívidas técnicas podem servir a propósitos comerciais valiosos, enquanto outras são simplesmente contraproducentes.

A dívida técnica pode ser compreendida através de uma analogia com o mundo financeiro. Quando uma empresa contrai uma dívida, ela assume um compromisso futuro de pagamento, geralmente com juros. Da mesma forma, quando uma equipe de desenvolvimento opta por um atalho técnico, ela está "contraindo uma dívida" que precisará ser "paga" no futuro, seja através de retrabalho, correções ou refatoração. A capacidade de contrair dívida com segurança, rastreá-la, gerenciá-la e quitá-la varia entre diferentes organizações, sendo fundamental uma tomada de decisão explícita antes de contrair a dívida e um acompanhamento mais transparente dessa dívida.

Classificação e Tipos de Dívida Técnica

Existem dois tipos básicos de dívida técnica, cada um com características distintas:

A Dívida Tipo I (Não Intencional) é aquela incorrida de forma não estratégica, geralmente como resultado de um trabalho de baixa qualidade. Por exemplo, uma abordagem de projeto que se mostra propensa a erros ou um programador júnior que escreve código inadequado. Em alguns casos, essa dívida pode ser contraída inconscientemente, como quando uma empresa adquire outra que acumulou dívida técnica significativa não identificada antes da aquisição.

A Dívida Tipo II (Intencional) é contraída deliberadamente, geralmente quando uma organização toma a decisão consciente de otimizar para o presente em vez do futuro. Isso comumente ocorre em situações onde prazos de entrega são críticos, levando a decisões como adiar a reconciliação de bancos de dados, limpar código fora dos padrões posteriormente ou escrever testes de unidade após o lançamento.

Dentro da dívida intencional, é possível fazer uma segunda distinção importante entre dívida de curto e longo prazo. **A Dívida de Curto Prazo (Tipo II.A)** é contraída tática e reativamente, geralmente como uma medida de estágio final para liberar uma versão

específica. Já a **Dívida de Longo Prazo (Tipo II.B)** é contraída de forma estratégica e proativa, como a decisão de não suportar uma segunda plataforma por pelo menos cinco anos.

Estratégias para Contração e Gestão da Dívida

A contração de dívida técnica por motivos estratégicos fundamenta-se sempre na percepção de que o custo do trabalho de desenvolvimento hoje é mais caro do que será no futuro. Isso pode ser verdadeiro por várias razões, incluindo time to market crítico, preservação de capital inicial em startups ou quando um sistema está próximo do final de sua vida útil.

É essencial contrair o tipo correto de dívida. Algumas dívidas são contraídas em grandes blocos identificáveis (semelhante a um empréstimo de carro), enquanto outras acumulam-se a partir de centenas de pequenos atalhos (semelhante à dívida de cartão de crédito). Esta última não traz benefícios mesmo no curto prazo e deve ser evitada.

Uma das implicações mais importantes da dívida técnica é que ela deve ser "servida", ou seja, uma vez que um projeto contrai uma dívida, haverá cobranças de juros. Se a dívida crescer o suficiente, eventualmente a empresa gastará mais servindo sua dívida do que investindo no aumento do valor de seus outros ativos. Um exemplo comum é uma base de código legado onde tanto trabalho é dedicado a manter o sistema em produção que sobra pouco tempo para adicionar novas capacidades ao sistema.

Transparência e Comunicação sobre Dívida Técnica

Um dos principais problemas com a dívida técnica é que as equipes de projeto a contraem intencionalmente, mas o acúmulo não é rastreado de forma visível. Isso é semelhante à forma como um indivíduo pode gastar numerosos itens em férias e depois se surpreender com a conta total quando a fatura do cartão de crédito chega no final do mês.

Para aumentar a transparência no rastreamento da dívida técnica, duas abordagens são recomendadas: manter uma lista de dívidas dentro do sistema de rastreamento de defeitos ou manter a lista de dívidas como parte de um backlog de produto Scrum. Ambas as abordagens podem ser usadas para aumentar a transparência da carga de dívida e do trabalho de serviço da dívida que precisa ocorrer dentro ou entre ciclos de lançamento.

O vocabulário da dívida técnica fornece uma maneira de comunicar com a equipe não técnica em uma área que tradicionalmente sofre com falta de transparência. Mudar o diálogo de um vocabulário técnico para um vocabulário financeiro fornece uma

estrutura mais clara e compreensível para essas discussões. Embora a terminologia de dívida técnica não esteja atualmente em uso generalizado, verifica-se que ela ressoa imediatamente com executivos e partes interessadas não técnicas.

Tomada de Decisão e Exemplos Práticos

Quando uma equipe chega a um ponto em que debate contrair dívida técnica, normalmente considera dois caminhos possíveis: o caminho "bom, mas caro" e o caminho "rápido e sujo". No entanto, existem três questões adicionais que devem ser consideradas: o custo de preencher retroativamente o caminho bom depois de já ter seguido o caminho rápido; o pagamento de juros sobre a dívida técnica; e a possibilidade de existir um terceiro caminho que seja mais rápido que o caminho bom, mas que não afete o resto do sistema.

Um exemplo geral de tomada de decisão ilustra que, ao enfrentar a perspectiva de contrair dívida técnica, é fundamental gerar mais de duas opções de projeto. Não se deve simplificar excessivamente a tomada de decisão de dívida técnica apenas para os dois extremos. Muitas vezes, uma abordagem híbrida acaba sendo a melhor opção.

Um exemplo específico envolve o desenvolvimento de um aplicativo que requer relatórios. A opção "boa" envolve escrever classes personalizadas de relatório; a opção "rápida e suja" usa o gerador de relatórios do banco de dados; e a opção "rápida, mas não suja" envolve usar o gerador do banco de dados, mas envolvendo o código do banco de dados em uma camada de tradução. Esta última opção, embora mais cara no curto prazo que a opção puramente rápida, não incorre em pagamentos de juros contínuos e permite adiar indefinidamente a decisão de implementar o código personalizado originalmente planejado.

Considerações Finais sobre Gestão Eficaz

A capacidade de pagar a dívida depende, pelo menos em parte, do tipo de software em que a equipe está trabalhando. Se uma equipe contrai dívida de curto prazo em uma aplicação web, uma nova versão pode ser facilmente implementada após a equipe completar seu trabalho de redução de dívida. No entanto, se uma equipe contrai dívida de curto prazo em firmware de aviação - cujo pagamento exige a substituição de uma caixa em um avião - essa equipe deve ter um limite mais alto para assumir qualquer dívida de curto prazo.

Trabalhar na redução da dívida pode ser motivador e bom para o moral da equipe. Uma boa abordagem, quando a dívida de curto prazo foi contraída, é dedicar a primeira iteração de desenvolvimento após um lançamento para pagar a dívida técnica de curto prazo. No entanto, desviar uma grande equipe por meses para se concentrar na redução da dívida raramente faz sentido. Em vez disso, recomenda-se dividir os

pagamentos de redução da dívida em partes muito menores e incluir uma porcentagem do trabalho de redução da dívida no fluxo de trabalho normal da equipe.

A gestão eficaz da dívida técnica requer um equilíbrio cuidadoso entre necessidades imediatas do negócio e sustentabilidade de longo prazo do software. Ao adotar uma abordagem estratégica e transparente para a dívida técnica, as organizações podem tomar decisões mais informadas que equilibrem adequadamente a velocidade de desenvolvimento com a qualidade e manutenibilidade do código.