# Débito técnico

O débito técnico se refere ao acúmulo de soluções não ideais adotadas durante o desenvolvimento de um projeto. É como se fosse uma "dívida" que a equipe contrai com o código, precisando ser paga com refatoração ou retrabalho no futuro.

A cada sprint é interessante que o PO defina um período para “pagamento” de Undone Work, para que no futuro os débitos técnicos no futuro não sejam de um tamanho grande.

O que pode ser considerado como um débito técnico

* Defeitos
* Falta de testes de aceitação
* Código duplicado
* Regra de negócio em lugares errados
* Impossibilidades de voltar para a versão anterior
* Algoritmos de difícil manutenção

Causas do Débito Técnico

* Pressão por prazos apertados: A necessidade de entregar funcionalidades rapidamente pode levar a soluções rápidas e não ideais.
* Falta de conhecimento: A equipe pode não ter o conhecimento técnico necessário para implementar a solução ideal.
* Escopo do projeto em constante mudança: Mudanças frequentes nos requisitos do projeto podem levar a soluções que não são mais adequadas.

Consequências do Débito Técnico

* Dificuldade em manter o código: O código com débito técnico é mais difícil de entender e modificar, o que pode aumentar o tempo e o custo de desenvolvimento.
* Bugs e problemas de qualidade: Soluções não ideais podem levar a bugs e problemas de qualidade no produto final.
* Desmotivação da equipe: Trabalhar com código com débito técnico pode ser frustrante e desmotivador para a equipe.

Como Gerenciar o Débito Técnico

* Conscientização: É importante que a equipe esteja ciente do que é débito técnico e quais são seus impactos.
* Priorização: A equipe deve priorizar a refatoração do código com débito técnico mais crítico.
* Comunicação: O Product Owner e stakeholders devem ser informados sobre o débito técnico e seus impactos no projeto.
* Planejamento: O débito técnico deve ser considerado no planejamento do projeto, reservando tempo e recursos para refatoração.

Dicas para Evitar o Débito Técnico

* Investir em testes: Testes automatizados podem ajudar a identificar bugs e problemas de qualidade antes que eles sejam entregues ao cliente.
* Refatorar regularmente: É importante reservar tempo para refatorar o código regularmente, evitando o acúmulo de débito técnico.
* Adotar boas práticas de desenvolvimento: Seguir boas práticas de desenvolvimento, como SOLID e Clean Code, pode ajudar a reduzir o débito técnico.

Lembre-se, o débito técnico é inevitável em qualquer projeto de software. O importante é gerenciar o débito técnico de forma eficaz para minimizar seus impactos.

Existem diversas técnicas e ferramentas que podem ajudar a gerenciar o débito técnico.