Como lidar com a dívida técnica

Luiz Pedone

Luiz Pedone

- Desenvolvedor de software na ThoughtWorks
- + 5 anos de experiência com desenvolvimento de software
- PHP, Java, Scala

@luizpedone
luiz.pedone@thoghtworks.com

O que é?

Imagine este cenário...

Um time de desenvolvimento tem que desenvolver uma nova funcionalidade. Esta funcionalidade tem o potencial de gerar um aumento significativo no faturamento da empresa, então quanto mais rápido a funcionalidade for para produção melhor.

Existem dois caminhos possíveis: um rápido e fácil de implementar, que resolve o problema no curto prazo mas torna a extensão da implementação ou manutenção difíceis. Existe outro, mais difícil e demorado de implementar mas que não gera problemas futuros.

Qual caminho seguir?

Baseado no artigo TechinalDebt do Martin Fowler: https://martinfowler.com/bliki/TechnicalDebt.html

Definição

"O termo 'dívida técnica' foi cunhado por Ward Cunningham para descrever a obrigação que uma organização de software incorre quando escolhe um design ou um tipo de construção que é prático no curto prazo mas que aumenta a complexidade e é mais custoso no longo prazo."

http://www.akitaonrails.com/2008/12/18/tradu-o-d-vida-t-cnica

Tipos de dívida técnica

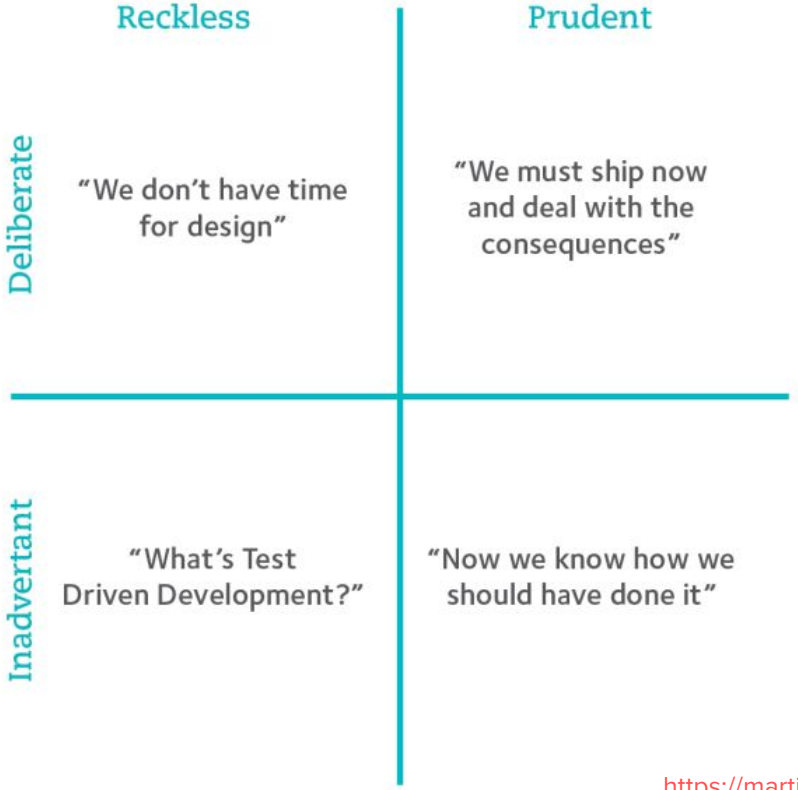
Não Intencional

Ocorre por decisões de implementação e design que não foram acertadas. Pode ser um código ruim ou difícil de mantido ou evoluído.

Intencional

Ocorre quando decisões de implementação são tomadas tendo em mente eventuais problemas futuros, deixando problemas já conhecidos para serem implementados no futuro.

Quadrante da dívida técnica



https://martinfowler.com/bliki/TechnicalDebtQuadrant.html

Sinais da dívida técnica

Sinais

- A velocidade de entrega do time é menor do que a esperada
- Dificuldade de implementar simples extensões de funcionalidade
- Código que o todos do time tem medo de tocar (se mudar algo na classe X tudo quebra)
- Acoplamento
- Baixa cobertura de testes
- Muitos bugs em produção

Como lidar?

Testes automatizados

- Código sem testes é uma dívida técnica.
- Modificar um código sem testes é muito mais custoso que um código testado.
- Refatorar sem testes é o caos. Isto desestimula melhorias. Se não há testes onde você está fazendo uma modificação, adicione.

"Deixe o código sempre melhor do que ele estava antes.", Uncle Bob

Backlog Técnico

- Ter um backlog com as melhorias técnicas que o time deseja realizar na base de código;
- Este backlog não deve ser o mesmo de funcionalidades novas;
- Uma prática legal é realizar reuniões mensais com o time de desenvolvimento para levantar ***pain points*** do código que podem ser melhorados. A conversa pode ser pautada pela pergunta: *"se você tivesse tempo livre, o que gostaria de melhorar?"*

Tempo para pagar a dívida

- Crie uma rotina que permita o pagamento da dívida técnica
    - Exemplo: tardes de sextas livres para focar em melhorias nas aplicações

Quais dívidas pagar?

Definir prioridades

- Nenhuma base de código é perfeita: sempre haverá o que ser melhorado. :)
- Como definir qual dívida pagar?
  - O custo da dívida (tempo gasto no desenvolvimento) é maior do que a refatoração
  - Trechos de código difíceis de usar mas que são modificados frequentemente
  - Bugs em produção gerados por causa da dívida técnica

Perguntas?

Luiz Pedone
luiz.pedone@thoughtworks.com