



Geek University

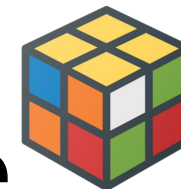
Evolua seu lado geek!

www.geekuniversity.com.br

Antipadrão no Desenvolvimento de Software



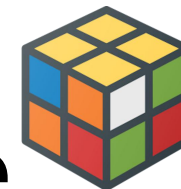
Antipadrão no Desenvolvimento de Software



Quando damos início ao desenvolvimento de software em uma aplicação ou projeto, pensamos na estrutura do código.

Esta estrutura é consistente com a arquitetura do projeto, o design, os casos de uso do cliente e muitas outras considerações de desenvolvimento.

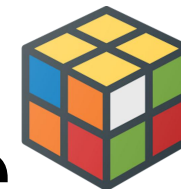
Antipadrão no Desenvolvimento de Software



Com frequência, enquanto o software é desenvolvido, ele se desvia da estrutura de código original por alguns dos motivos:

- O processo de raciocínio do desenvolvedor evolui com o desenvolvimento;
- Os casos de uso tendem a mudar com base no feedback dos clientes;
- As estruturas de dados projetadas inicialmente podem passar por mudanças de acordo com considerações de funcionalidade ou de escalabilidade;

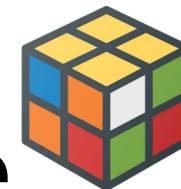
Antipadrão no Desenvolvimento de Software



Pelos motivos apresentados, o software com frequência passa por refatorações.

A refatoração tem uma conotação negativa para muitas pessoas; na verdade, porém, ela é uma das partes mais relevantes na jornada do desenvolvimento de software, oferecendo uma oportunidade aos desenvolvedores de revisar as estruturas de dados e pensar na escalabilidade e nas necessidades sempre em evolução dos clientes e sistemas.

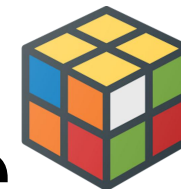
Antipadrão no Desenvolvimento de Software



São vários os antipadrões observados no desenvolvimento de software, dentre eles:

- Código Espaguete;
- Martelo de Ouro;
- Fluxo de Lava;
- CTRL + c / CTRL + v;

Antipadrão no Desenvolvimento de Software



Código Espaguete:

Este é o antipadrão mais comum e mais conhecido no desenvolvimento de software. O código espaguete é difícil de manter e de otimizar.

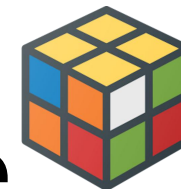
As causas típicas do código espaguete incluem:

- Desconhecimento de programação e análise orientada a objetos;
- A arquitetura ou o design do produto não são considerados;
- A mentalidade do desenvolvedor é voltada à correções rápidas;

Você saberá que tem um código espaguete em mãos quando os seguintes pontos forem verdadeiros:

- Somente uma reutilização mínima das estruturas é possível;
- Os esforços de manutenção são muito altos;
- A capacidade de extensão e flexibilidade para mudar são reduzidas;

Antipadrão no Desenvolvimento de Software



Martelo de Ouro:

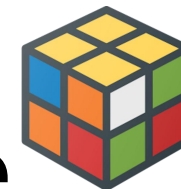
No desenvolvimento de software, você deve ter visto vários exemplos (ou poderá ver) em que uma dada solução (tecnologia, design ou módulo) é usada em muitos lugares porque esta solução traria vantagens a diversos projetos, funcionando quase como uma *bala de prata*.

Felizmente você fez este curso de Padrões de Projetos aqui na Geek University e aprendeu que uma solução é mais adequada em um determinado contexto e é aplicada a determinados tipos de problema.

No entanto as equipes de desenvolvimento de software tendem a optar por uma solução comprovada sem considerar se ela é apropriada às necessidades.

Este é o motivo pelo qual este antipadrão é conhecido como Martelo de Ouro: um martelo para todos os pregos possíveis (ou uma solução para todos os problemas).

Antipadrão no Desenvolvimento de Software

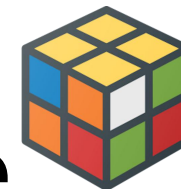


Martelo de Ouro:

As causas típicas do Martelo de Ouro, incluem:

- A solução chega como uma recomendação vinda de cima (de arquitetos ou líderes de tecnologia), de pessoas que não estão próximas ao problema específico em mãos;
- Uma solução gerou muitos benefícios no passado, mas em projetos com um contexto e requisitos diferentes;
- Uma empresa está presa a essa tecnologia, pois investiu dinheiro em treinamento dos funcionários ou estes se sentem mais confortáveis com ela.

Antipadrão no Desenvolvimento de Software

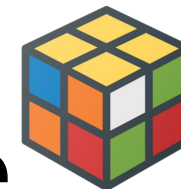


Martelo de Ouro:

As consequências do Martelo de Ouro, incluem:

- Uma solução é obsessivamente aplicada a todos os projetos de software;
- O produto é descrito não pelos seus recursos, mas pela tecnologia usada no desenvolvimento;
- Nos corredores da empresa, você ouve os desenvolvedores dizendo que “aquilo poderia ter sido melhor do que isso”;
- Os requisitos não são concluídos e não estão em sincronia com as expectativas dos usuários.

Antipadrão no Desenvolvimento de Software



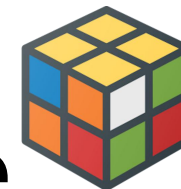
Fluxo de Lava:

Este padrão está relacionado ao Código Morto (Dead Code), isto é, um trecho de código inutilizável, que permanece na aplicação de software por medo de que ele cause falhas em outros lugares caso seja modificado.

À medida que o tempo passa, este trecho de código continua no software e solidifica a sua posição, como se fosse lava se transformando em rocha sólida.

Pode ocorrer nos casos em que começarmos a desenvolver um software para dar suporte a determinado caso de uso, mas este muda com o tempo.

Antipadrão no Desenvolvimento de Software

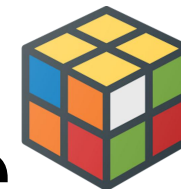


Fluxo de Lava:

As causas de um Fluxo de Lava (Lava Flow) incluem:

- Muito código de tentativa e erro na produção;
- Código escrito por uma só pessoa, que não é revisado e é passado para outras equipes de desenvolvimento sem qualquer treinamento sobre este código;
- O raciocínio inicial usado na arquitetura ou no design de software é implementado na base de código, mas ninguém mais o entende.

Antipadrão no Desenvolvimento de Software

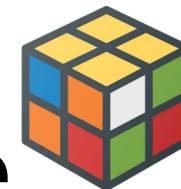


Fluxo de Lava:

Os sintomas de um Fluxo de Lava são:

- Baixa cobertura de código para testes desenvolvidos (se é que existem testes);
- Muitas ocorrências de código comentado sem motivo, ou pior ainda, código sem comentário;
- Interfaces obsoletas ou desenvolvedores tentando contornar código existente;

Antipadrão no Desenvolvimento de Software



CTRL + c / CTRL + v:

Este é o principal antipadrão.

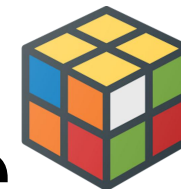
Desenvolvedores experientes disponibilizam trechos de códigos online, seja no Github, seja no Stack Overflow, que são soluções para alguns problemas que ocorrem comumente.

Com frequência, os desenvolvedores copiam estes códigos exatamente como estão e os usam em sua aplicação para avançar no desenvolvimento.

Neste caso não há nenhuma validação para saber se esse é o código mais otimizado ou se ele é realmente apropriado ao contexto.

Isso resulta em uma aplicação de software sem flexibilidade e de difícil manutenção.

Antipadrão no Desenvolvimento de Software

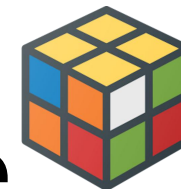


CTRL + c / CTRL + v:

As causas de uma programação do tipo “ctrl + c / ctrl + v” são:

- Desenvolvedores iniciantes não acostumados a escrever código ou que não saibam como desenvolver;
- Correção de bug ou avanços rápidos no desenvolvimento;
- Duplicação de código por necessidade de uma estrutura de código ou de uma padronização entre os módulos;
- Falta de pensamento em longo prazo ou planejamento antecipado;

Antipadrão no Desenvolvimento de Software

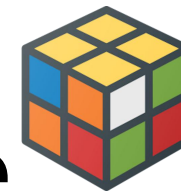


CTRL + c / CTRL + v:

As consequências de uma programação do tipo “ctrl + c / ctrl + v” são:

- Tipos de problemas semelhantes entre aplicações de software;
- Custos mais altos de manutenção e ciclo de vida mais longos para bugs;
- Base de código menos modular, com o mesmo código executado em várias linhas;
- Problemas de herança que já existiam antes.

Antipadrão no Desenvolvimento de Software



CTRL + c / CTRL + v:

As consequências de uma programação do tipo “ctrl + c / ctrl + v” são:

- Tipos de problemas semelhantes entre aplicações de software;
- Custos mais altos de manutenção e ciclo de vida mais longos para bugs;
- Base de código menos modular, com o mesmo código executado em várias linhas;
- Problemas de herança que já existiam antes.



Geek University

Evolua seu lado geek!

www.geekuniversity.com.br