



Geek University

Evolua seu lado geek!

www.geekuniversity.com.br

Antipadrões na arquitetura de software



pythonTM

Design Patterns

Antipadrões na arquitetura de software

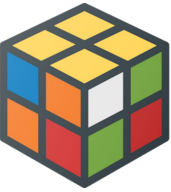


A arquitetura de software é uma parte importante da arquitetura de sistemas em geral.

Enquanto a arquitetura de sistemas tem como foco aspectos como o design, ferramentas e hardware, entre outros, a arquitetura de software está voltada para a modelagem do software, que deve ser bem compreendida pelas equipes de desenvolvimento e de testes, por gerentes de produtos e outros stakeholders (pessoas-chave).

Esta arquitetura desempenha um papel crucial para determinar o sucesso da implementação e no modo como o produto funcionará para os clientes.

Antipadrões na arquitetura de software



Reinventando a roda

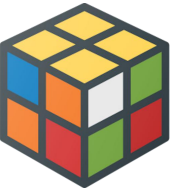
Com frequência, ouvimos os líderes de tecnologia falarem de não reinventar a roda.

O que isso quer dizer essencialmente?

Para alguns, pode significar a reutilização de arquitetura. Por exemplo, você resolveu um problema e concebeu uma solução no nível de arquitetura. Se você se deparar com um problema semelhante em qualquer outra aplicação, o processo de raciocínio (arquitetura ou design) desenvolvido anteriormente deve ser reutilizado.

Ou seja, não há motivos para rever o mesmo problema e conceber uma nova solução, o que, essencialmente, significa reinventar a roda.

Antipadrões na arquitetura de software



Causas que levam à reinvenção da roda:

- Ausência de uma documentação ou de um repositório central que discutam problemas no nível de arquitetura e soluções implementadas;
- Falta de comunicação entre líderes de tecnologia na comunidade ou na empresa;
- Construir do zero é o processo seguido pela empresa; basicamente, há processos imaturos, sem uma sólida implementação e aderência ao processo.

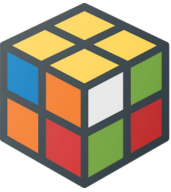
Antipadrões na arquitetura de software



As consequências deste antipadrão incluem:

- Soluções em demasia para resolver um problema-padrão, com muitas delas não tendo a atenção merecida;
- Mais tempo e utilização de recursos para a equipe de engenharia, resultando em estouros no orçamento e mais tempo para alcançar o mercado.
- Uma arquitetura de sistema fechada (a arquitetura é útil apenas para um produto), duplicação de esforços e gerenciamento de riscos precário.

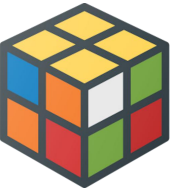
Antipadrões na arquitetura de software



As consequências deste antipadrão incluem:

- Soluções em demasia para resolver um problema-padrão, com muitas delas não tendo a atenção merecida;
- Mais tempo e utilização de recursos para a equipe de engenharia, resultando em estouros no orçamento e mais tempo para alcançar o mercado.
- Uma arquitetura de sistema fechada (a arquitetura é útil apenas para um produto), duplicação de esforços e gerenciamento de riscos precário.

Antipadrões na arquitetura de software

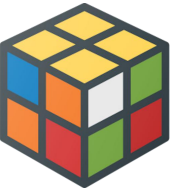


Vendor lock-in (Dependência de Fornecedor)

Como o nome deste antipadrão sugere, as empresas de produto tendem a ser dependentes de algumas tecnologias oferecidas pelos fornecedores.

Essas tecnologias estão muito amarradas a seus sistemas, a ponto de ser difícil afastar-se delas.

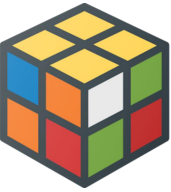
Antipadrões na arquitetura de software



Causas de um Vendor Lock-in:

- Familiaridade com as pessoas que têm autoridade na empresa fornecedora e possíveis descontos na compra da tecnologia;
- Tecnologia escolhida com base em ofertas de marketing de vendas, e não na avaliação da tecnologia;
- Uso de uma tecnologia comprovada, indicando que o retorno sobre os investimentos com essa tecnologia realmente foram elevados na experiência anterior, no projeto atual, mesmo quando ela não é apropriada para as necessidades ou os requisitos do projeto;
- Especialistas em tecnologias/desenvolvedores já estão treinados para usar essa tecnologia;

Antipadrões na arquitetura de software



Consequências de um Vendor Lock-in:

- Ciclos de lançamento e de manutenção do produto de uma empresa são diretamente dependentes do ciclo de lançamentos do fornecedor;
- O produto é desenvolvido em torno da tecnologia, e não com base nos requisitos do cliente;
- O tempo para o produto alcançar o mercado não é confiável e não atende às expectativas do cliente.

Antipadrões na arquitetura de software



Design by Committee (Design por Comitê):

As vezes com base no processo de uma organização, um grupo de pessoas se reúne e faz o design de um sistema em particular.

A arquitetura de software resultante muitas vezes é complexa ou fica abaixo do padrão, pois envolve muitos processos de raciocínio, e os especialistas em tecnologia, que podem não ter o conjunto de habilidades adequado nem a experiência para fazer o design dos produtos, impõem suas ideias.

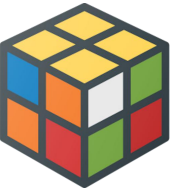
Antipadrões na arquitetura de software



Causas do Design por Comitê:

- O processo da empresa envolve a aprovação da arquitetura ou do design por muitos stakeholders;
- Não há nenhum ponto de contato ou arquiteto único responsável pelo design;
- As prioridades do design são definidas pelo marketing ou por especialistas em tecnologia, e não pelo feedback do cliente;

Antipadrões na arquitetura de software



Sintomas observados neste antipadrão:

- Pontos de vista conflitantes entre desenvolvedores e arquitetos, mesmo após o design ter sido concluído;
- Design excessivamente complexo, muito difícil de ser documentado;
- Qualquer mudança na especificação ou no design passa por revisão de muitas pessoas, resultando em atrasos na implementação;



Geek University

Evolua seu lado geek!

www.geekuniversity.com.br