

Projeto de Software

Definição

- O projeto é o processo criativo de **transformar** uma especificação de um problema em uma especificação de uma solução. No projeto de software utilizam-se a especificação e os modelos de requisitos gerados na fase de análise e especificação de requisitos. A partir dos requisitos, muitas soluções são possíveis e, portanto, muitos projetos diferentes podem ser produzidos. Uma solução é considerada adequada ao problema, se ela satisfizer a todos os requisitos especificados.

Princípios do Projeto

- considerar abordagens alternativas com base nos requisitos (funcionais e não funcionais) e conceitos de projeto de software;
 - o que importa é atender aos requisitos, adaptabilidade é fundamental
- estar relacionado aos modelos de análise e à especificação de requisitos e deve ser a eles rastreado;
 - parte-se na análise para o projeto e essa relação deve ser mantida a todo tempo para que não se perca o foco, para que os requisitos sejam atendidos
- não “reinventar a roda”, isto é, reutilizar componentes, frameworks, padrões e outras soluções que se mostraram eficazes em outros projetos, sobretudo aqueles similares ao sistema em desenvolvimento;
- exibir uniformidade (estilo) e integração (interfaces entre componentes);

Princípios do Projeto

- ser estruturado para acomodar mudanças (alterabilidade);
- ser passível de avaliação da qualidade;
- ser revisado para minimizar erros;
- minimizar a distância conceitual e semântica entre o software e o mundo real. Os modelos de projeto devem ser facilmente compreensíveis, tendo em vista que seu propósito é comunicar informações para profissionais responsáveis pela codificação, testes e manutenção;
- acomodar circunstâncias não usuais e, se necessário abortar o processamento, fazê-lo de modo elegante;
 - um bom tratamento de situações de exceção é fundamental;
- apresentar nível de abstração superior ao código fonte, afinal, projeto não é codificação.

Princípios da Modelagem Ágil

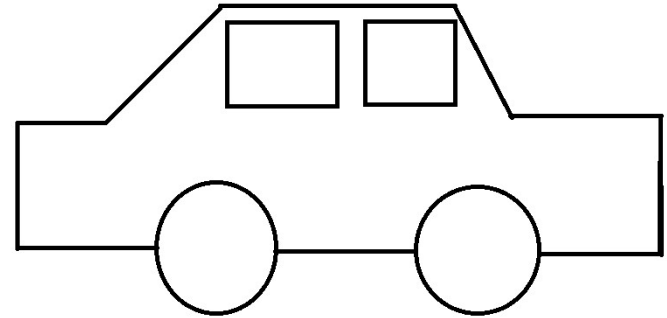
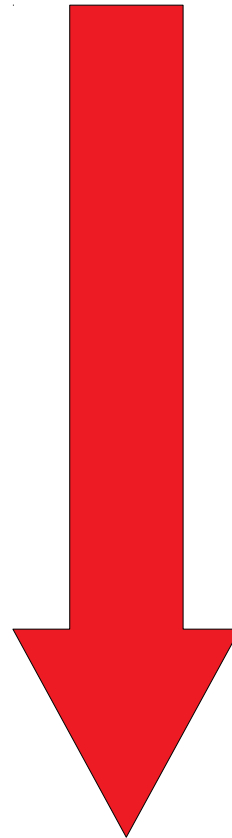
Uma vez que a fase de projeto é essencialmente uma atividade de modelagem, princípios da modelagem ágil também se aplicam:

- Seja econômico. Não crie mais modelos do que você precisa. Seja capaz de declarar um objetivo para cada modelo criado.
- Procure produzir modelos mais simples.
- Construa modelos de modo que sejam passíveis de mudanças.
- Obtenha feedback tão logo quanto possível.

Aspectos de Qualidade de Projeto de Software

- Níveis de Abstração

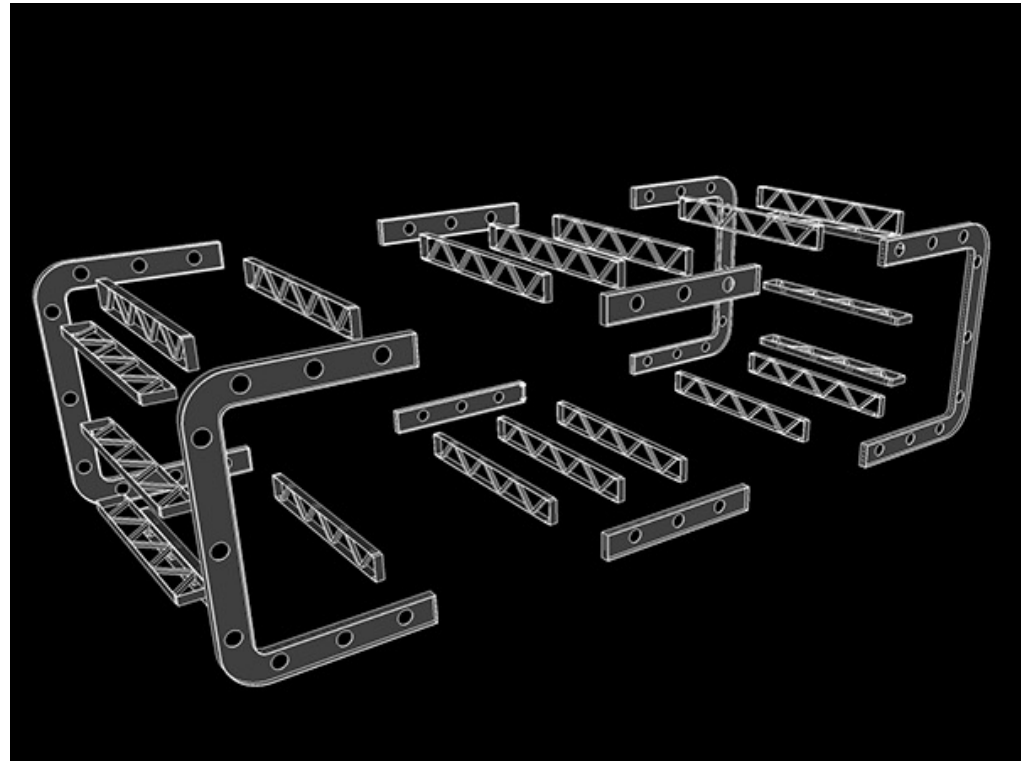
Maior abstração
(menos detalhes)
para menor
abstração (mais
concreto, mais
próximo do
resultado final)



Aspectos de Qualidade de Projeto de Software

- Modularidade

“Dividir para conquistar”



Aspectos de Qualidade de Projeto de Software

- Ocultação das Informações

“Módulos devem encapsular o máximo possível, mostrar apenas o necessário (interface)”



Aspectos de Qualidade de Projeto de Software

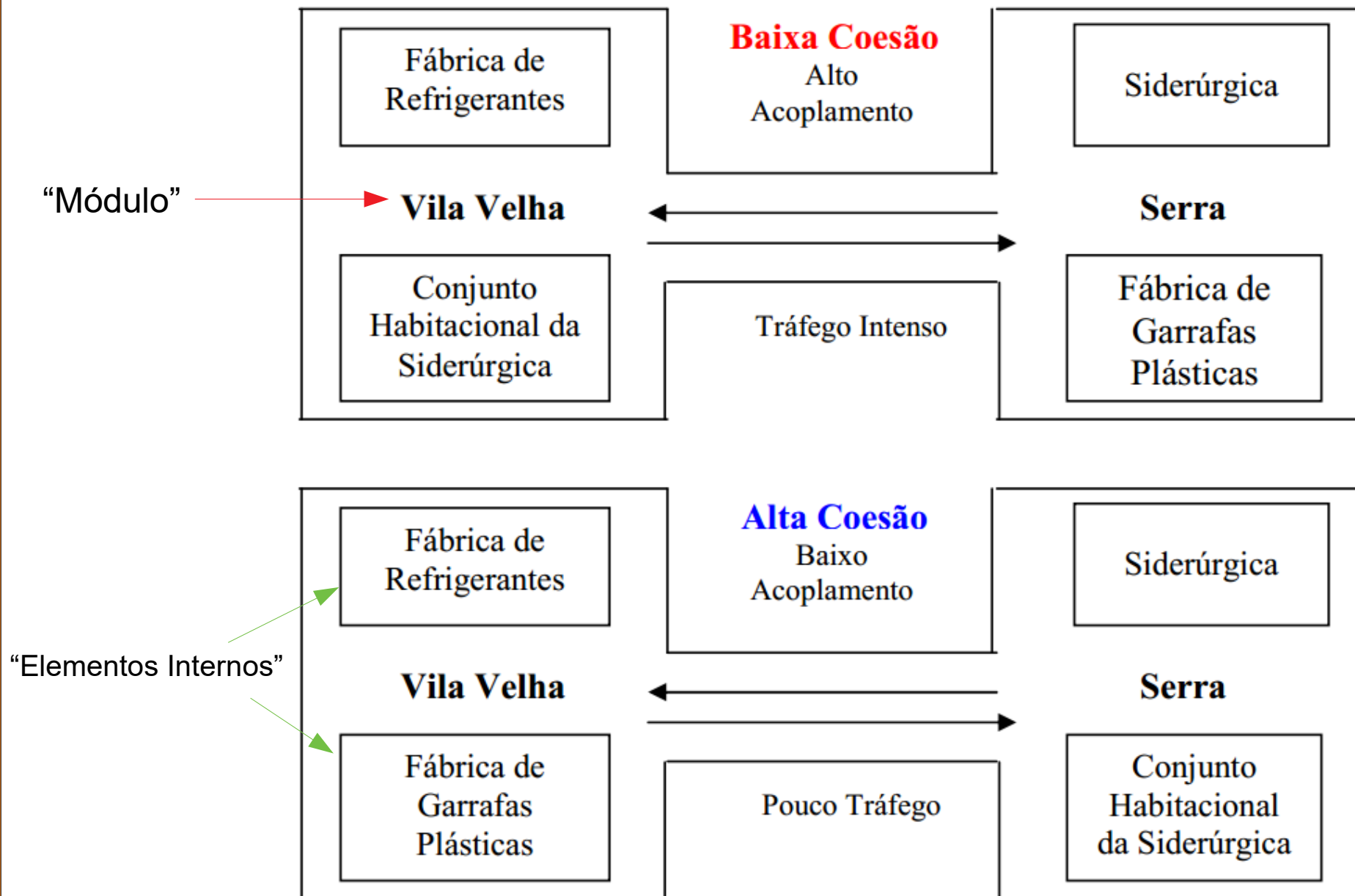
- Independência funcional

A independência funcional é uma decorrência direta da modularidade e dos conceitos de abstração e ocultação de informações. Ela é obtida pelo desenvolvimento de módulos com finalidade única e pequena interação com outros módulos. Isto é, módulos devem cumprir uma função bem estabelecida, minimizando interações com outros módulos. Módulos funcionalmente independentes são mais fáceis de entender, desenvolver, testar e alterar.

Aspectos de Qualidade de Projeto de Software

- Independência funcional

Um módulo possui uma alta independência funcional quando possui **alta coesão** (alta ligação entre os elementos internos do módulo) e **baixo acoplamento** (baixa relação entre módulos – independência tão maior quanto possível).



Projeto e Atributos da Qualidade

Atributo	Definição
Funcionalidade	Satisfaz às necessidades explícitas e implícitas do sistema.
Confiabilidade	Mantém o mesmo nível de desempenho, sob condições estabelecidas, por um período de tempo.
Usabilidade	Refere-se ao esforço necessário para se utilizar o software.
Eficiência	Relação entre a quantidade de recursos disponibilizados e o desempenho do software.
Manutenibilidade	Facilidade para modificar o software.
Portabilidade	Facilidade de levar o software de um ambiente para outro.

Projeto de Software e Padrões

- Reutilizar códigos é fundamental (“não reinventar a roda”). Nesse contexto usamos padrões.
- Padrão é uma solução testada e aprovada para um problema geral. Padrões possuem vantagens e desvantagens e sua escolha depende do software e contexto a ser implementado.
- O objetivo de um padrão é registrar a experiência no projeto de software, facilitando assim reaproveitamento com maior qualidade.

Dúvidas?

