

Reúso de Software

Paulo C. Masiero

Base: Sommerville, Cap. 16, Ed. 9ª.

Introdução

- Reúso de software é uma estratégia em que o desenvolvimento de software é baseado no reúso de software existente → eng. de soft. baseada em reúso.
- Busca maximizar o reúso de software existente.
- História
 - Linguagens de Programação: sub-rotinas
 - McIroy, M. D. – Mass-produced software components, Proc. of the NATO Conf. On Software Engineering, Garmisch, Germany, Springer, 1968.

Introdução

- Engenharia de software baseada em reúso:
busca:
 - Menores custos de produção e manutenção de software
 - Entregas mais rápidas
 - Software com melhor qualidade
 - Aumentar o retorno sobre o investimento em software

Introdução (Cont.)

- O reúso pode acontecer com unidades de tamanhos muito diferentes:
 - Reúso de aplicações (ou sistemas): a aplicação toda é reusada, ou é configurada para diferentes clientes ou pode-se desenvolver uma família de aplicações com base em uma arquitetura comum
 - Reúso de componentes: componentes de software de tamanho variado
 - Reúso de objetos e funções: componentes que implementam uma única função (ocorre há mais de 40 anos). Ex. funções matemáticas.

Introdução (Cont.)

- Outra forma de reúso é o reúso de conceitos.
- Conceito: é uma representação abstrata que inclui detalhes de implementação. Por exemplo: um modelo de classes
- O conceito pode ser instanciado e configurado para uma série de situações. Ex. padrões de projeto, padrões de arquitetura.

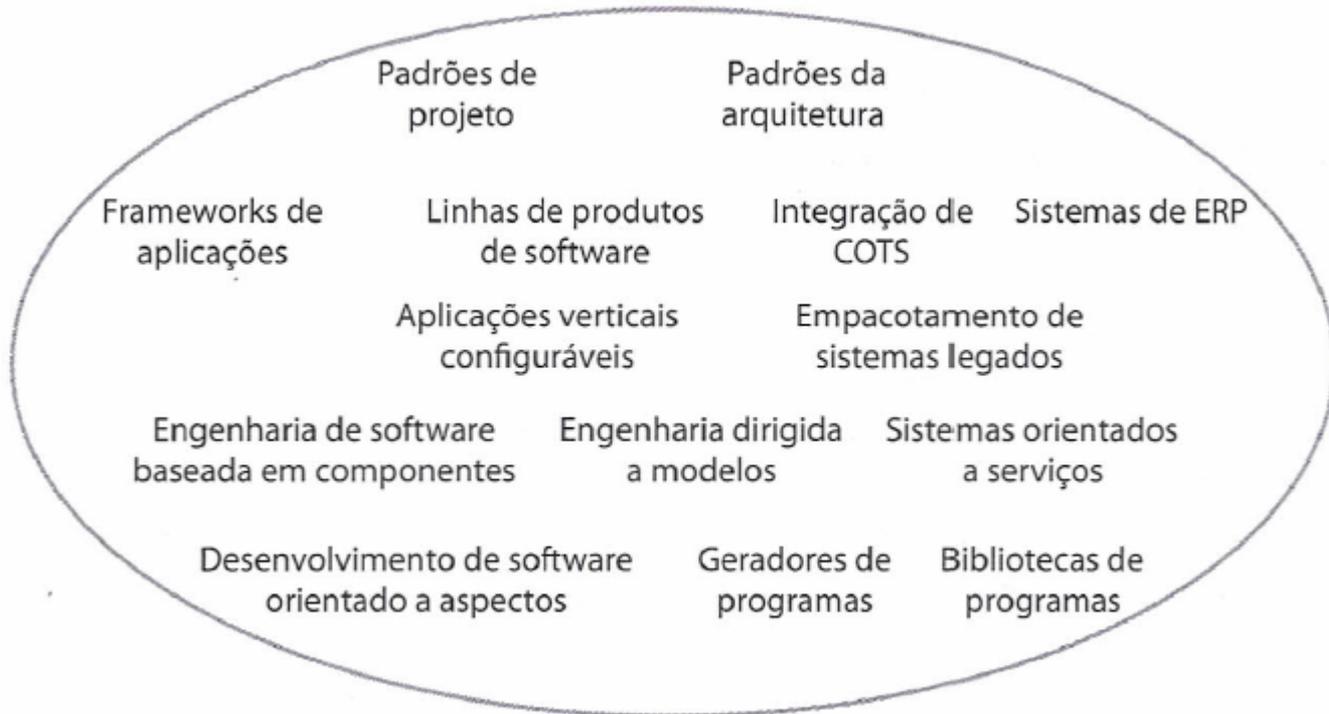
Reúso - Vantagens

- Redução dos custos de desenvolvimento.
- Confiança aumentada
- Redução do risco de processo
- Uso eficaz de especialistas
- Conformidade com padrões
- Desenvolvimento acelerado

Reúso: possíveis problemas

- Maiores custos de manutenção
- Falta de ferramentas de suporte
- Síndrome do “não-inventado-aqui”
- Criação, manutenção e uso de uma biblioteca de componentes
- Encontrar, compreender e adaptar componentes reusáveis

Panorama de reúso



Abordagem	Descrição
Padrões de arquitetura	Padrões de arquitetura de software que oferecem suporte a tipos comuns de sistemas de aplicação são usados como base de aplicações. São descritos nos capítulos 6, 13 e 20.
Padrões de projeto	Abstrações genéricas que ocorrem em todas as aplicações são representadas como padrões de projeto, mostrando os objetos abstratos e concretos e as interações. São descritos no Capítulo 7.
Desenvolvimento baseado em componentes	Sistemas são desenvolvidos através da integração de componentes (coleções de objetos) que atendem aos padrões de modelos e componentes. São descritos no Capítulo 17.
<i>Framework</i> de aplicações	Coleções de classes abstratas e concretas são adaptadas e estendidas para criar sistemas de aplicação.
Empacotamento de sistemas legados	Sistemas legados (veja o Capítulo 9) são 'empacotados' pela definição de um conjunto de interfaces e acesso a esses sistemas legados por meio dessas interfaces.
Sistemas orientados a serviços	Sistemas são desenvolvidos pela ligação de serviços compartilhados, que podem ser fornecidos externamente. São descritos no Capítulo 19.
Linhas de produtos de software	Um tipo de aplicação é generalizado em torno de uma arquitetura comum para que esta possa ser adaptada para diferentes clientes.
Reúso de produto COTS	Sistemas são desenvolvidos pela configuração e integração de sistemas de aplicação existentes.
Sistemas de ERP	Sistemas de grande porte que sintetizam a funcionalidade e as regras de negócios genéricos são configurados para uma organização.
Aplicações verticais configuráveis	Sistemas genéricos são projetados para poder ser configurados para as necessidades dos clientes de sistemas específicos.
Bibliotecas de programas	Bibliotecas de classe e funções que implementam abstrações comumente usadas são disponibilizadas para reúso.
Engenharia dirigida a modelos	O software é representado como modelos de domínio e modelos de implementação independentes. O código é gerado a partir desses modelos. São descritos no Capítulo 5.
Geradores de programas	Um sistema gerador incorpora o conhecimento de um tipo de aplicação, e é usado para gerar sistemas nesse domínio a partir de um modelo de sistema fornecido pelo usuário.
Desenvolvimento de software orientado a aspectos	Quando o programa é compilado, os componentes compartilhados são integrados em uma aplicação em diferentes locais. São descritos no Capítulo 21.

Questão fundamental?

- Qual é a técnica mais apropriada para usar em determinada situação?
- Isso depende....
 - Requisitos do sistema em desenvolvimento
 - Tecnologia disponível
 - Disponibilidade de ativos reusáveis
 - Conhecimento e experiência (*expertise*) da equipe de desenvolvimento

Fatores-chave a considerar para o reúso

- O cronograma de desenvolvimento do software
- A expectativa de duração do software
- O conhecimento, as habilidades e a experiência da equipe de desenvolvimento
- A importância do software e seus requisitos não funcionais
- O domínio da aplicação
- A plataforma em que o sistema será executado

Há uma grande gama de alternativas e situações

- Na maioria dos casos há a possibilidade de algum tipo de reúso (parcial)
- Muitas vezes o reúso não é atingido por problemas gerenciais e não técnicos. Ex. o gerente não quer comprometer os requisitos.

Restante do capítulo 16

- Frameworks de aplicações
- Linhas de produtos de software
- Reúso de produto COTS

Reúso de Produtos COTS (software pronto)

- Produto de prateleira (Commercial-of-the-shelf): é um produto que pode ser alterado para as necessidades de vários clientes sem alterar o código fonte do sistema.
- Como é projetado para uso geral, costuma incluir muitos recursos e funções.
- Baseado em “melhores práticas”.
- Muitas vezes, produtos de código aberto também são usados sem alterar o código fonte, como se fossem COTS.

Reúso de Produtos COTS (Cont.)

- Esta abordagem tem sido amplamente usada por empresas nas últimas duas décadas, para Sistemas de Informação.
- Vantagens:
 - Rapidez de implantação
 - Não precisam dedicar muitos recursos para TI
 - A evolução é simplificada
 - Alguns riscos de desenvolvimento são evitados
 - É possível escolher as funções que necessita

Reúso de Produtos COTS (Cont.)

- Desvantagens/Riscos
 - Os requisitos precisam ser adaptados ao COTS
 - Quando os requisitos não podem ser adaptados, a empresa precisa mudar seus processos de negócio.
 - A escolha do COTS que melhor se adequa à empresa pode ser difícil e custosa, principalmente se não houver especialistas locais e for preciso contratar consultores. Pode ser desastroso se errar.
 - O fornecedor do COTS controla a evolução do sistema

Reúso de Produtos COTS (Cont.)

- Há dois tipos de reúso de produtos COTS:
 - Sistemas de solução COTS
 - Solução genérica de um único fornecedor , configurada para os requisitos do cliente.
 - Ex. Um sistema de gestão de bibliotecas
 - Sistemas integrados COTS
 - Envolve a integração de dois ou mais sistemas COTs, talvez de diferentes fornecedores.
 - Ex. Um sistema ERP, como os da SAP, ORACLE, TOTVS etc.