

Software Reuse: Metrics and Models

Tarcísio Filó

Universidade Federal de Minas Gerais

tfilo@dcc.ufmg.br

29 de outubro de 2013

Overview

- 1 Introdução
- 2 Modelos e Métricas
 - Análise Custo-Benefício
 - Avaliação de Maturidade
 - Quantidade de Reúso
 - Modelo de Formas de Falha
 - Avaliação de Reusabilidade
 - Métricas para Bibliotecas de Reúso
- 3 Conclusões

Introdução

- **Reuso:** Uso de de artefatos pré-existentes de software para criar um novo software.
- É um ponto crucial para melhorar a **qualidade** de software e a **produtividade**.
- **Reusabilidade** é o potencial de reúso de um artefato de software.
- Para haver **retorno** com a reutilização de software, um programa de reuso deve ser sistemático.
- Organizações que implementam programas sistemáticos de reuso devem ser capazes de **medir** o progresso.

Introdução

- No trabalho são investigadas métricas e modelos de reuso e reusabilidade de software.
- O objetivo é traçar um panorama da área para auxiliar a medição de progresso nos programas sistemáticos de reuso, por meio de métricas e modelos.
- Uma **métrica** é um indicador quantitativo de algum atributo de alguma coisa.
- Um **modelo** especifica a essência de algum processo.

Introdução

O autor categoriza modelos e métricas de réuso em 6 tipos:

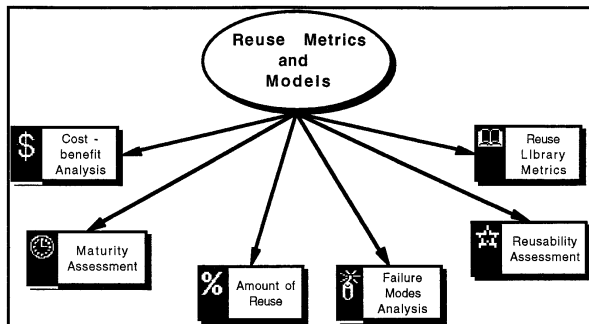


Figura: Categorização de modelos e métricas para réuso

Organizações frequentemente encontram a necessidade dessas métricas e modelos na presente ordem.

Análise Custo-Benefício

- Justificar o custo e tempo investido em reuso, estimando o potencial retorno financeiro.
- Gaffney and Durek [1989] propõem dois modelos: um simples, que mostra o custo-benefício de se reutilizar componentes de software, e um segundo mais complexo, que considera também o custo do desenvolvimento do componente reutilizável.

Análise Custo-Benefício - Simple Model

- C é o custo relativo ao código novo (por exemplo, $C = 1$).
- R é a proporção de código reutilizado no produto.
- b é o custo relativo a incorporar o código reutilizado no produto.
- A equação para isso é: $C = (1)(1 - R) + (b)(R) = [(b - 1)R] + 1$

Análise Custo-Benefício - Simple Model

- Gaffney and Durek estimaram $b = 0.85$ para réuso somente do código e $b = 0.08$ para réuso de outros artefatos no decorrer do projeto como requisitos e projeto.
- $C = [(b - 1) * R] + 1$
- $C = [(0.85 - 1) * 0.3] + 1 = 0.955$
- $C = [(0.08 - 1) * 0.3] + 1 = 0.724$

Análise Custo-Benefício - Cost-Of-Development Model

- Adicionar ao modelo simples o custo de desenvolvimento dos componentes reutilizáveis.
- Custo vai sendo amortizado nas utilizações.

| Variable | Definition |
|----------|---|
| R | % of code contributed by reusable components |
| b | integration cost of reusable component as opposed to development cost |
| RC | relative cost of overall development effort |
| RP | relative productivity |
| E | relative cost of making a component reusable |
| N_0 | payoff threshold value (number of reuses needed to recover all component development costs) |

| Formulas: | |
|-----------|--------------------------------|
| | $RC = (b + (E / N) - 1) R + 1$ |
| | $RP = 1/RC$ |
| | $N_0 = E / (1-b)$ |

Figura: Modelo para Avaliação de Custo-Benefício

Análise Custo-Benefício - Cost-Of-Development Model

- Favaro [1991] aplicou o modelo de Gaffney and Durek [1989] em um projeto Ada-based.

| | E | b | N_0 for simple implementations | N_0 for complex implementations |
|------------|-----|------|----------------------------------|-----------------------------------|
| Monolithic | 1.0 | 0.10 | 1.33 | 2.56 |
| Polythitic | 1.2 | 0.15 | 1.69 | 3.40 |
| Graph | 1.6 | 0.25 | 2.56 | 5.72 |
| Menu | 1.9 | 0.30 | 3.25 | 7.81 |
| Mask | 2.2 | 0.40 | 4.40 | 12.97 |

Figura: Threshold do trabalho de Favaro [1991]

Análise Custo-Benefício - Qualidade de Investimento

- Proposta por Barnes & Bollinger [1991].
- Relação Produtor-Consumidor.
- Atividades “Produtoras” são investimentos em reuso.
- Atividades “Consumidoras” são atividades que tiveram benefícios financeiros, dado os investimentos em reuso.

Análise Custo-Benefício - Qualidade de Investimento

- Q é a **Qualidade de Investimento**.
- B é o benefício propiciado às “Atividades Consumidoras”, por tudo aquilo produzido nas “Atividades Produtoras”.
- R é o investimento em reuso.
- A equação é: $Q = B/R$. Se a empresa investiu \$100 em um componente, que poupou \$1000 em projetos que sucederam ao investimento, a qualidade é $Q = 1000/100 = 10$.
- Se $Q > 1$, boa qualidade.

Avaliação de Maturidade

- Koltun and Hudson [1991] propõem um modelo de maturidade de reuso no qual avaliam o quão avançada a organização está na implementação de reuso de forma sistemática, apresentando uma escala para tal.
- O modelo é similar ao CMM.
- O modelo pode ser utilizado pela organização para guiar atividades que devem ser executadas para adquirir maior maturidade.
- A idéia é que o reuso se torne parte do negócio da organização.

Avaliação de Maturidade

| | 1 Initial/ Chaotic | 2 Monitored | 3 Coordinated | 4 Planned | 5 Ingrained |
|---|---|--|---|---|--|
| Motivation/ Culture | Reuse discouraged | Reuse encouraged | Reuse incentivized re-enforced rewarded | Reuse indoctrinated | Reuse is the way we do business |
| Planning for reuse | None | Grassroots activity | Targets of opportunity Department | Business imperative Division | Part of strategic plan Enterprise wide |
| Breadth of reuse | Individual | Work group | | | |
| Responsible for making reuse happen | Individual initiative | Shared initiative | Dedicated individual | Dedicated group | Corporate group with division liaisons |
| Process by which reuse is leveraged | Reuse process chaotic; unclear how reuse comes in. | Reuse questions raised at design reviews (after the fact) | Design emphasis placed on off the shelf parts | Focus on developing families of products | All software products are generalized for future reuse |
| Reuse assets | Salvage yard (no apparent structure to collection) | Catalog identifies language and platform specific parts | Catalog organized along application specific lines | Catalog includes generic data processing functions | Planned activity to acquire or develop missing pieces in catalog |
| Classification activity | Informal, individualized | Multiple independent schemes for classifying parts | Single scheme catalog published periodically | Some domain analyses done to determine categories | Formal, complete, consistent timely classification |
| Technology support | Personal tools, if any | Many tools, but not specialized for reuse | Classification aids and synthesis aids | Electronic library separate from development environment | Automated support integrated with development environment |
| Metrics | No metrics on reuse level, pay-off, or costs | Number of lines of code used in cost models | Manual tracking of reuse occurrences of catalog parts | Analyses done to identify expected payoffs from developing reusable parts | All system utilities, software tools and accounting mechanisms instrumented to track reuse |
| Legal, contractual, accounting considerations | Inhibitor to getting started | Internal accounting scheme for sharing costs and allocating benefits | Data rights and compensation issues resolved with customer | Royalty scheme for all suppliers and customers | Software treated as key capital asset |

Figura: Modelo de Maturidade de Reúso nas organizações

Quantidade de Reúso

- As métricas de quantidade de reutilização são usadas para avaliar e monitorar o esforço de reutilização, acompanhando porcentagens de reutilização de objetos durante seu ciclo de vida.

$$\frac{\text{amount of life cycle object reused}}{\text{total size of life cycle object}}$$

Figura: Formula Geral

$$\frac{\text{lines of reused code in system or module}}{\text{total lines of code in system or module}}$$

Figura: Formula Utilizando linhas de código

Quantidade de Reúso

- O autor cita o trabalho de Frakes [1990] e Frakes [1993], para programas desenvolvidos em C e outros.
- Um componente de software pode ser interno ou externo.
- Um componente interno é aquele desenvolvido para um determinado programa, criado para aquele fim.
- Um componente externo é aquele proveniente de outro fim.

Quantidade de Reúso

- L é o total de componentes no software.
- E é o número de componentes externos.
- I é o número de componentes internos.
- M é o número de componentes internos utilizados mais de uma vez.
- $ExternalReuse = E/L$, $InternalReuse = M/L$
- $TotalReuse = ExternalReuse + InternalReuse$
- Ex.: $E = 10$, $I = 50$, $M = 5$, $L = 60$
- $ExternalReuse = 10/60 = 0.17$, $InternalReuse = M/L = 5/60 = 0.08$
- $TotalReuse = 0.17 + 0.08 = 0.25$

Quantidade de Reúso

- Em softwares orientados por objetos, há métricas que quantificam o reúso, indiretamente, destacando o trabalho de Chidambler & Kemerer [1994].
- Nesse trabalho, foi proposto um conjunto de métricas, e o autor destaca a métrica DIT (Depth In Tree), ou Profundidade na Árvore de Herança, como indicador de reúso.
- DIT mede a distância máxima da classe avaliada até a classe raiz na herança.

Quantidade de Reúso

$DIT(C0) = 0$, $DIT(C0') = 0$, $DIT(C1) = 1$, $DIT(C2) = 2$, $DIT(C3) = 3$,
 $DIT(C4) = 4$

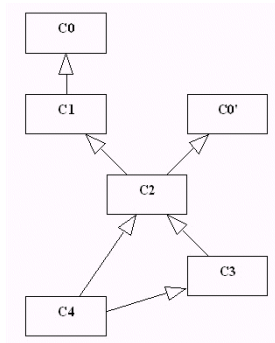


Figura: Exemplo DIT

Quantidade de Reúso

- Segundo o autor, a herança é uma forma simples de se reutilizar.
- Projetos que utilizam pouco o recurso da herança tem, conseqüentemente, menor reúso.
- Outras métricas do conjunto CK também compartilham a propriedade de avaliar o reúso, mesmo que de forma indireta.
- WMC (Weighted Method per Class), é uma métrica que mede o somatório da complexidade dos métodos da classe.
- Quanto maior for o número de métodos na classe, maior a tendência da classe ficar menos específica, limitando assim o potencial de reúso.

Modelo de Formas de Falha

- Implementação Sistemática de Reúso é difícil e envolve tanto fatores técnicos como fatores não técnicos
- Esse modelo de análise provê uma forma de melhorar o programa de reúso baseado nas formas em que pode haver falha.
- Proposto no trabalho de Frakes & Fox [1996], o modelo pode ser utilizado para avaliar a qualidade do programa de reúso, determinando os impedimentos estratégicos que atrapalham o programa.
- A organização pode atuar na melhoria do processo considerando as formas como o reúso pode falhar.

Modelo de Formas de Falha

- O modelo contém sete falhas correspondentes aos passos que um engenheiro de software precisa para completar, em ordem, um componente reutilizável:
- **No Attempt to Reuse**
- Part Does Not Exist
- Part Is Not Available
- Part Is Not Found
- Part Is Not Understood
- Part Is Not Valid
- **Part Can Not Be Integrated**

Avaliação de Reusabilidade

- Essa área consiste em estimar a reusabilidade de um determinado componente de software.
- Métricas que atuem nessa área são fundamentais em duas áreas de reúso: projeto e reengenharia para reúso.
- A pergunta é: Existem atributos mensuráveis que indicam o potencial de reúso ?
- Se sim, esses seriam fundamentais para os projetos e reengenharia de reúso.
- Uma das dificuldades é que os atributos de reúso são intimamente relacionados a tecnologia utilizada.
- O objetivo dessa seção é revisar o trabalho que tem sido feito nessa área.

Avaliação de Reusabilidade

- Num estudo na NASA, Selby [1989] identificou vários atributos que distinguem componentes reutilizáveis:
- Menor quantidade de chamadas a outros módulos por linha de código. (é uma forma de mensurar acoplamento)
- Maior taxa de comentários por linha de código.
- Basili et al. [1990] identificaram o acoplamento de módulos por meio de referências a variáveis e parâmetros de outros tipos. Módulos mais independentes são potencialmente mais reutilizáveis, ao contrário de módulos fortemente acoplados.

Métricas para Bibliotecas de Reúso

- Uma biblioteca para reúso é um repositório para armazenar componentes reutilizáveis, com uma interface para procurar nesse repositório.

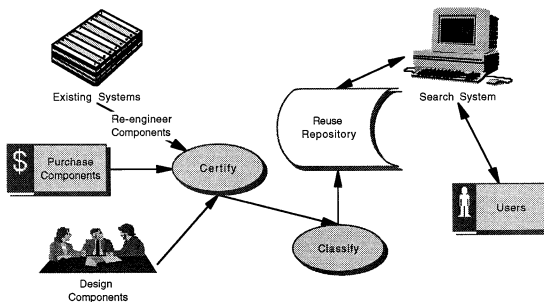


Figura: Biblioteca para Reúso

Métricas para Bibliotecas de Reúso

- Frakes & Gandel [1990] destacaram em seu trabalho a importância das ferramentas de busca nas bibliotecas de reúso, para real efetividade da sua utilização.
- Os critérios de avaliação são:
 - Custo de Indexação
 - Precisão + Recall
 - Eficiência (consumo de recursos físicos)

Métricas para Bibliotecas de Reúso

- Frakes & Nejme [1987] propuseram alguns indicadores para avaliar a qualidade dos componentes na biblioteca:
- **Time in Use:** utilizado em um ou mais sistemas (em produção) por um período de 3 meses.
- **Reuse Statistics:** à medida em que o módulo foi reutilizado com sucesso por outros, vai sendo atualizado.
- **Reuse Reviews:** opiniões favoráveis daqueles que já utilizaram o módulo.
- **Inspection:** módulo tem de ter passado por inspeção.
- **Testing:** 100% de testes unitários e 80% de cobertura em testes de integração.

Conclusões

- Um programa de reutilização deve ser planejado, deliberado e sistemático, se é para dar grandes retornos.
- Organizações que implementam programas sistemáticos de reutilização de software, para melhorar a produtividade e qualidade, devem ser capazes de medir seu progresso e identificar as estratégias mais eficazes de reutilização.
- No artigo foi apresentada uma pesquisa de métricas e modelos para réuso.

References



Frakes, Willian; Terry, Carol (1996)
Software Reuse: Metrics and Models

The End