

Avaliação de Projeto via Métricas de Classe no Ensino da Engenharia de Software

Sebastião Jesus, Mark Song

Pontifícia Universidade Católica de Minas Gerais
Belo Horizonte - MG - Brazil
`{sebast,song}@pucminas.br`

Abstract. *The software system development process applies different software engineering concepts and techniques to generate high quality products. Metrics are one among such techniques used to evaluate aspects as size, complexity, reuse and others. This work presents a tool which extracts quantitative metrics used to evaluate the quality of UML modeled in object-oriented software design given in software engineering courses.*

Resumo. *O processo de desenvolvimento de software utiliza diferentes conceitos e técnicas para construção de software com qualidade. Métricas têm sido usadas para a mensuração dos diferentes aspectos de projeto como tamanho, complexidade, reutilização de código, e outros. Este artigo apresenta uma ferramenta para a extração de métricas quantitativas utilizadas na avaliação da qualidade de projeto orientado a objeto, modelado com UML, em cursos de engenharia de software.*

1. Introdução

Para [Genero 2002] construir software com qualidade é um imperativo no mercado de produtos de software, fazendo com que a medição da qualidade seja uma questão central em qualquer sistema a ser desenvolvido. A disponibilidade de medidas que indiquem a boa qualidade do software de forma objetiva evita polêmicas sobre questões subjetivas ligadas ao processo de avaliação de qualidade. Além disto, como a tarefa de manutenção consome e continuará a consumir a maior parte dos recursos do ciclo de vida, a preocupação com a qualidade se torna ainda mais relevante.

Este artigo apresenta uma proposta para avaliação da qualidade de projeto de software desenvolvido por estudantes de engenharia de software apoiada em uma ferramenta de extração de métricas quantitativas diretamente do projeto orientado a objeto modelado com UML (Linguagem Unificada de Modelagem) e exportado no formato de intercâmbio de dados XMI (Intercâmbio de Metadado com Linguagem de Marcação Extensível). O objetivo é proporcionar ao aluno uma ferramenta que permita ao mesmo avaliar seu trabalho frente às diferentes soluções obtidas para um mesmo problema.

Para a validação da proposta, um estudo de caso com nove grupos de alunos e um mesmo tema foi avaliado. As métricas extraídas de nove projetos produzidos por eles foram usadas para uma avaliação empírica da qualidade dos mesmos através da atribuição de notas (escala de 0 a 10) com três diferentes critérios: calculando-se os valores médios, comparando-se com valores de um projeto de referência e comparando-os a valores catalogados na literatura.

Este texto está organizado em 5 seções. A seção 2 apresentada uma revisão da literatura acerca de métricas e respectivas aplicações na avaliação da qualidade de um produto. A seção 3 descreve a arquitetura da ferramenta, as bases para sua construção e o modelo de avaliação de projeto. A seção 4 apresenta o estudo de caso e a seção 5 conclui o artigo.

2. Revisão da Literatura

No ensino do desenvolvimento de software uma das primeiras atividades é a construção de modelos que orientam o projeto e a construção de artefatos. O processo inicia-se com o entendimento do problema real e emprega diferentes técnicas para alcançar o produto final.

Por modelo de software, entende-se a abstração construída pelo arquiteto de software visando identificar faceta específica do produto em desenvolvimento. Pode-se, por exemplo, abstrair tanto os aspectos funcionais do sistema como a arquitetura operacional do mesmo. Seu principal objetivo é uma simplificação que permita o entendimento do todo através da análise das partes.

Modelos de métricas permitem a avaliação da qualidade dos artefatos produzidos em cada fase do projeto - Concepção, Elaboração, Construção e Implantação - antecipando correções, quando necessárias, no decorrer do desenvolvimento e evitando desperdício de tempo e de custos desnecessários que poderiam ocorrer após o produto entrar em operação. Além disto, permitem mensurar as atividades de análise, projeto, codificação, documentação e teste [El-Wakil 2005], [Olague 2007].

2.1 Ferramentas para Modelagem com UML

Diversas ferramentas para a modelagem, usando UML, estão disponíveis, sejam estas de uso livre como ArgoUML, Umbrello, JUDE, OMONDO, ou proprietárias como Rose Enterprise e Artisan Real-Time Studio. O objetivo é possibilitar a especificação das características desejadas do modelo de forma rápida e interativa, com consequente melhoria da produtividade nas tarefas envolvidas.

2.2 Métricas

Nesta pesquisa utiliza-se a definição de métrica estabelecida pelo IEEE Standard Glossary of Software Engineering Terms:

"Medida quantitativa do grau em que um sistema, componente ou processo possui determinado atributo".

O uso de um conjunto de métricas permite a avaliação de diferentes características do software. Estas características possibilitam medir e quantificar atributos de qualidade como acoplamento, coesão, complexidade, herança, reutilização e tamanho que poderão fornecer indicativos da existência de falhas de projeto.

2.2.1 Modelos de Métricas

Os diversos modelos apresentados na literatura têm grande interseção, às vezes com variações de nome, mas conceitualmente iguais. Três dos principais modelos de métricas são descritos a seguir.

O modelo proposto por [Chidamber 1994] denominado MOOSE (Metrics for Object Oriented Software Engineering), é considerado precursor e serve como referência para estudos sobre métricas OO (Orientadas a Objeto). Validado empiricamente por [Basili 1996] é utilizado na obtenção de indicadores de qualidade e de falhas em projetos de classes. São exemplos destas métricas: RFC (Response For a Class) - o conjunto de métodos que podem ser potencialmente executados em resposta a uma mensagem recebida - e NOC (Number Of Children) - a contagem das classes diretamente subordinadas a uma classe ancestral.

O modelo MOOD (Metrics for Object Oriented Design) [Brito e Abreu 1996], [El-Wakil 2005] é composto por seis métricas: MHF (Method Hiding Factor) e AHF (Attribute

Hiding Factor) para avaliar encapsulamento, MIF (Method Inheritance Factor) e AIF (Attribute Inheritance Factor) para herança, CF (Coupling Factor) para acoplamento, PF (Polimorfism Factor) para polimorfismo.

O modelo de [Genero 2002] apresenta diversas métricas dividindo-se em dois grandes grupos: as obtidas no contexto de um diagrama de classes (aplicável ao diagrama como um todo) e as observadas no escopo da classe (aplicadas a uma única instância do modelo)

Para os autores a crescente demanda por qualidade de software tem colocado o tema qualidade como um diferencial entre produtos. Por esta razão desenvolvedores necessitam de medidas objetivas para utilizarem na avaliação e melhoria do produto a partir dos estágios iniciais do desenvolvimento. Na abordagem proposta, o diagrama de classes é de grande importância, pois serve de base para todo o trabalho de projeto e implementação. Além disto, focar na qualidade do diagrama de classes contribui significativamente para a melhoria dos sistemas [Genero 2005].

3. Extração de Métricas e Avaliação de Projeto

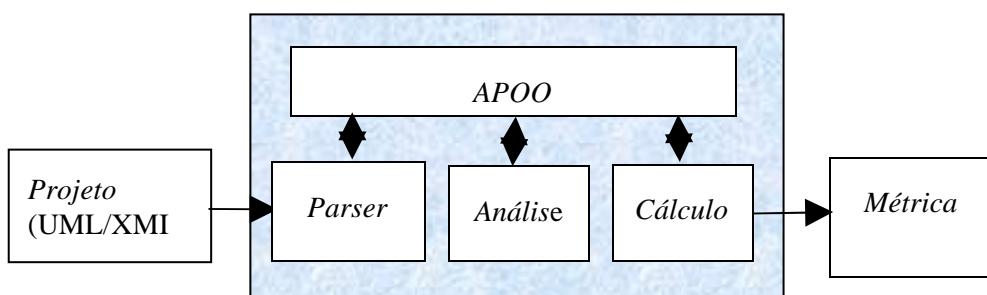


Figura 1 - Módulos de APOO.

Uma ferramenta para extração de métricas denominada APOO (Avaliação de Projetos Orientados a Objetos) desenvolvida na plataforma Java faz parte deste trabalho. Sua arquitetura e seus módulos são exibidos na Figura 1. Seu objetivo é processar projeto de software, descrito em UML e exportado em XMI, extraíndo métricas que permitam avaliar aspectos de qualidade.

A criação da ferramenta tem como justificativa a extração automática de métricas uma vez que se torna trabalhoso ou inviável, para um instrutor ou aluno, efetuar o cálculo de métricas a partir de modelos visuais – em geral a avaliação é centrada na documentação em papel. Outra vantagem é a utilização do padrão XMI, formato adotado por uma ampla gama de ferramentas de modelagem.

O uso de um ambiente de extração de métricas e avaliação dos projetos propicia tanto ao instrutor, quanto ao aluno, além do citado acima: incentivar a cultura de medição no processo de desenvolvimento de software, colaborar para desenvolvimento de projetos e construção de aplicações com qualidade e, contribuir para o desenvolvimento de novas ferramentas automatizadas que permitam a avaliação da qualidade do projeto em suas fases iniciais.

3.1 Métricas do APOO

Na atual versão de APOO são extraídas métricas derivadas do diagrama de classes modelado com UML. Dos diferentes modelos apresentados são consideradas as categorias: tamanho, herança, e encapsulamento [Brito e Abreu 1996], acoplamento [Chidamber 1994] e, relacionamento [Harrison 1998] englobando características como: número de classes, de subclasses, de métodos, e atributos, o número de descendentes de uma classe (subclasses), a

profundidade da árvore de herança, o acoplamento entre as classes e o ocultamento de métodos e de atributos. A Tabela 1 mostra algumas métricas para estas características.

Tabela 1 – Exemplos de Métricas de APOO.

Sigla	Nome	Sigla	Nome
NC	Nº de classes	TMP	NMPu/NM – Taxa de Métodos Públicos
NCPu	Nº de classes públicas	FOM	Fator de ocultamento de método
NCPv	Nº de classes privadas	FOA	Fator de Ocultamento de Atributo
NCPt	Nº de classes protegidas	MaxNF	Nº máximo de filhos de uma classe
NCPc	Nº de classes de pacote	CBOh	CBO por herança – acoplamento entre objetos
NCIn	Nº de classes de interface	CBOp	CBO por passagem de parâmetros
NCAb	Nº de classes abstratas	MaxPAH	Profundidade máxima da árvore de herança
NCAs	Classes de associação	MM	NM/NC - média de métodos por classe
NRAss	Relacionamentos de associação	MMPu	NMPu/NC – média de métodos públicos por classe
NR Agr	Relacionamentos de agregação	MMPV	NMPv/NC – média de métodos privados por classe

3.2 Formas de avaliação

Os valores de métricas extraídos por APOO permitem que o avaliador efetue comparações entre diferentes projetos, relativos a um mesmo tema, considerando:

- Comparações com medidas de tendência central. Calcula-se o valor médio de cada métrica considerando um grupo de projetos. Para cada métrica é definida uma faixa de tolerância em torno da média. A nota de cada projeto servirá para que o avaliador faça uma análise do aproveitamento de cada aluno quanto a aplicação de conceitos de engenharia de software no desenvolvimento do projeto;
- Comparação com um projeto de referência. Semelhante à forma anterior substituindo valores médios por valores de um projeto de referência submetido a APOO e fornecido pelo instrutor;
- Comparação com valores previamente catalogados na literatura. Os valores das métricas são comparados com valores catalogados na literatura admitindo-se uma faixa de tolerância definida previamente.

4. Estudo de Caso

Este estudo analisa nove projetos de software de estudantes do curso de Engenharia de Software, com prévio conhecimento dos conceitos de programação, análise e projeto orientado a objeto e o suposto domínio dos métodos, técnicas e ferramentas de modelagem de software usando UML.

O projeto consiste no desenvolvimento, em grupos, de um software de controle de manutenção de veículos em ambiente cliente-servidor para tratamento de informações sobre manutenção de veículos de diferentes tipos (passeio, utilitários e motos) em diferentes lojas de uma rede. A Figura 2 mostra um modelo de classes de um dos grupos. Observam-se claramente algumas falhas como: classes sem associação, classes vazias, métodos centrados no CRUD (Create, Read, Update, Delete) e influenciados por DER (Diagrama Entidade e Relacionamento) da área de Banco de Dados, comuns em versões iniciais do projeto, os quais afetarão as métricas.

Os projetos recebem uma nota do instrutor antes de serem submetidos à ferramenta - outros aspectos do projeto como documentação, casos de uso, diagrama de seqüência e de distribuição são considerados na avaliação. Um valor (nota) é atribuído para que se possa verificar a aderência da avaliação definida por APOO em relação à avaliação manual.

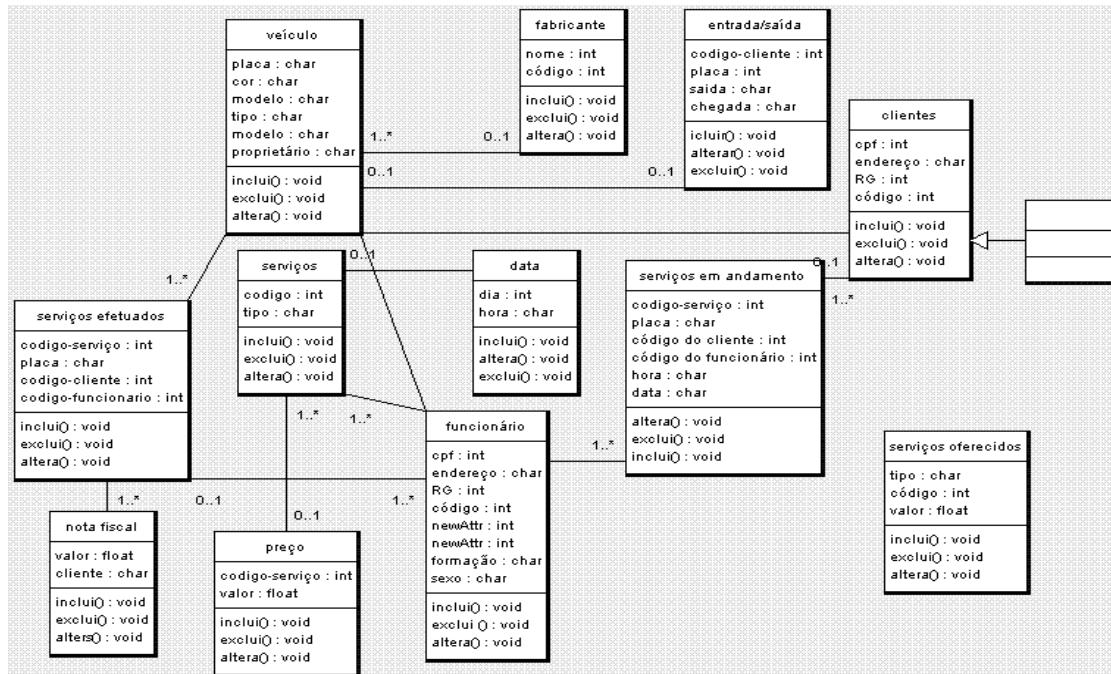


Figura 2 - Diagrama de Classes de um dos projetos.

4.1 Resultados

A Tabela 2 apresenta valores das métricas obtidos para todos os projetos. Exibe também os valores de um projeto de referência (PR) fornecido pelo avaliador. O projeto de referência é decorrente da especificação de requisitos fornecida pelo avaliador refletindo o resultado de discussões sobre alternativas de desenvolvimento - é um balizamento que procura esclarecer questões semânticas surgidas durante o desenvolvimento que se dá em 16 semanas. A seguir são apresentadas algumas análises sobre os resultados obtidos.

4.1.1 Métricas relativas a tamanho e complexidade

Estas métricas fornecem informações básicas do tamanho e complexidade do projeto. São denominadas primitivas, pois derivam diretamente da contagem dos elementos presentes e servem de base para o cálculo das métricas derivadas:

A. Número de classes (NC) - indica o tamanho do modelo identificado pelos estudantes e suas percepções relativas ao escopo do projeto e consequentemente o esforço de construção do software.

A variação nos valores encontrados para a métrica NC pode ser observada na Figura 3. Era de se esperar uma maior uniformidade, pois os estudantes possuem um mesmo nível de conhecimento. Note que distorções podem ser indicativas de dificuldades no entendimento do escopo do projeto. Conforme os valores obtidos, apenas três projetos se aproximam do padrão de referência (PR). Outro fato observado é a presença de classes sem identificação, o que eleva o número de classes indicando descuido de projeto.

Tabela 2 – Métricas Obtidas para o Sistema de Controle de Veículos.

Características	Métricas	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	P ₈	P ₉	PR
Classes do projeto	NC	15	13	32	11	23	7	8	25	17	8
Encapsulamento de classes	NCPu	15	13	32	11	23	7	8	25	17	8
Associações de classes	NCA _s	0	0	0	0	0	2	0	0	0	1
	NCI _n	0	1	0	0	0	0	0	0	0	0
	NCA _b	0	0	0	0	0	0	0	0	0	0
Relacionamentos de classes	NRAss	41	43	38	38	11	33	17	26	22	5
	NRAgre	0	0	0	0	0	1	0	0	0	0
	NRCCom	0	0	0	0	0	1	0	0	0	0
	NRDep	0	0	0	0	1	3	0	0	0	0
	NRHer	0	1	0	0	10	1	2	0	6	0
	NRImp	0	0	0	0	0	0	0	0	0	0
Atributos do projeto	NA	67	45	56	40	42	14	26	67	48	37
	NAPu	67	44	56	37	41	14	26	60	48	8
	NAPv	0	1	0	3	1	0	0	7	0	0
	NAPT	0	0	0	0	0	0	0	0	0	29
Métodos do projeto	NM	40	36	40	40	32	0	0	36	8	34
	NMPu	40	36	40	40	31	0	0	36	8	27
Herança	maxNF	0	1	0	0	3	0	2	0	3	0
	MaxPAH		1			1		1		1	0
Acoplamento	CBO _h	0	1	0	0	10	1	2	0	6	0
	CBO _p								54	0	3
Ocultamento	FOM	0%	0%	0%	0%	3%	0%	0%	0%	0	21%
	FOA	0%	2%	0%	8%	2%	0%	0%	10%	0	78%

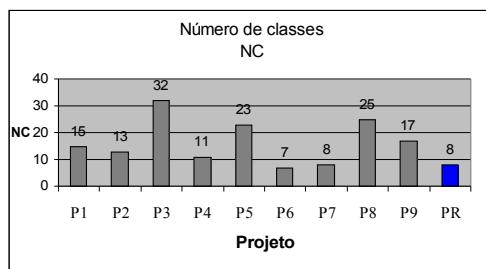


Figura 3. Métrica NC.

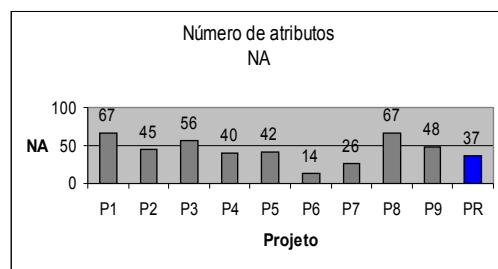


Figura 4. Métrica NA.

B. Número de atributos (NA) - esta métrica (Figura 4) é essencial para a verificação de fatoração das classes além de aspectos de ocultação de informação. Observa-se uma grande variação nos valores encontrados - mínimo de 14, máximo de 67, com amplitude 53. É de se esperar que o aumento no número de classes implique em um aumento no número de atributos.

Uma relação NA/NC muito alta é um indicativo que o desenvolvedor não fatora/particiona as classes do projeto visando reuso via herança ou associação. No caso de uma relação baixa é um indicativo de classes subutilizadas. Tal consideração é importante para se aperfeiçoar o entendimento do conceito de abstração.

C. Número de métodos (NM) - Não ocorre grande variação nos valores encontrados com exceção do projeto P9. Os projetos P6 e P7 são casos típicos de descuido na modelagem, uma vez que, nenhum método é disponibilizado nas classes de negócio. Outro aspecto a ser considerado é a relação NM/NC. Pode-se observar a existência de classes sem métodos quando a relação é menor que um (1), como no projeto P9.

4.1.2 Métricas relativas à herança

Tais medidas são utilizadas para indicar o grau de reuso verificado. Alguns valores são apresentados a seguir:

A. Número máximo de filhos (maxNF): indica um grau de reuso desejável. Define a divisão em subclasses. Um valor muito alto pode indicar excesso de classes especializadas. Além disto, pode indicar centralização de responsabilidades em poucas classes. Pode-se observar que apenas quatro projetos (P2, P5, P7 e P9) consideram a especialização para modelagem do problema, isto é, número de filhos maior que 0 (zero), tirando benefícios da reutilização de classes.

B. Profundidade máxima da árvore de herança (maxPAH): esta métrica permite verificar o grau de reutilização de classes no projeto. Quanto maior a profundidade da classe na hierarquia, maior a complexidade para a correta implementação e testes. Um maior esforço deve ser despendido para o entendimento dos componentes disponibilizados pelas classes ancestrais.

Observa-se que apenas quatro projetos estabeleceram maxPAH com valores aceitáveis. Os demais não consideraram ou não identificaram a necessidade de especializações.

4.1.3 Métricas relativas a encapsulamento

As métricas de encapsulamento indicam se o desenvolvedor se preocupa em definir atributos e métodos visíveis, exportados pela classe. Fornece, portanto uma medida do ocultamento de atributos e métodos [Brito e Abreu 1996], [Harrison 1998].

Recomenda-se que os fatores de ocultamento de atributos sejam altos, fato este que indica a necessidade de serem declarados privados ou protegidos. Em contra partida, como os métodos devem ser compartilhados precisam ser declarados públicos implicando em um fator de ocultamento baixo. Os valores obtidos são mostrados a seguir:

A. Fator de Ocultamento de atributo-FOA: indica o grau de ocultamento de atributos de todo o projeto. É a relação entre atributos ocultos e o total de atributos do projeto.

Observa-se que não houve, durante o projeto, uma análise mais profunda do aspecto de ocultamento de atributos (o maior valor é 10%) e cinco dos projetos consideraram todos os atributos públicos. Neste caso, tal fato indica uma percepção incorreta dos conceitos de abstração – não se ocultam detalhes de implementação – e o acesso aos dados permite a existência de objetos inconsistentes.

4.2 Avaliações dos Projetos

As subseções seguintes apresentam as avaliações efetuadas. O resultado final da avaliação é uma nota (zero a dez) que correspondente ao percentual dos pontos obtidos pelo projeto considerando as faixas de aceitação definidas para cada métrica.

4.2.1 Avaliação com Medidas de Tendência Central

Para esta forma de avaliação são realizados cálculos estatísticos para obtenção dos valores de referência tomando-se, como universo, todos os projetos apresentados. São obtidas medidas de tendência central e erro de estimativa (Tabela 3). Por exemplo, o projeto P₅ apresentou 13 métricas na faixa de um total de 16 métricas consideradas.

Tabela 3 – Avaliação com Medida de Tendência Central

Pontos:									
P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	P ₈	P ₉	
7	15	7	11	13	3	4	8	11	
Notas									
4,4	9,4	4,4	6,9	8,1	1,9	2,5	5	6,9	

Este modelo de avaliação está sujeito à ocorrência de valores extremos, muito altos ou muito baixos, que acabam por influenciar a média e o desvio padrão. Observa-se também que a mesma ponderação para todas as métricas pode não ser a ideal. Tal fato pode ser usado para avaliações parciais de modo a uniformizar o entendimento dos grupos de projeto acerca de conceitos ainda não consolidados. Por exemplo, de classe, de herança, de acoplamento, de associação e outros, podendo-se assim reduzir a influência de valores extremos obtendo-se um desvio menor em torno da média.

4.2.2 Avaliação usando o Padrão de Referência

Para esta forma de avaliação cada métrica é comparada com a de um projeto padrão de referência fornecido ao APOO pelo instrutor. Este define uma faixa de tolerância em torno do valor considerado padrão. São consideradas, para efeito de avaliação, apenas as que se situam dentro do intervalo de tolerância. A nota é o percentual das métricas no intervalo.

A Tabela 4 exibe nova avaliação dos projetos frente a um projeto de referência do avaliador. Neste caso os resultados demonstram que este tipo de avaliação torna o processo mais rígido em relação ao anterior. O uso de um padrão permite ao avaliador calibrar o intervalo de aceitação considerando diferentes aspectos relevantes ao projeto - amplia (ou reduz) o intervalo de tolerância de acordo com considerações obtidas pelo próprio grupo.

Tabela 4 – Avaliação com um Projeto de Referência

Pontos dos projetos usando o Padrão de Referência									
P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	P ₈	P ₉	PR
2	7	2	7	7	5	5	4	2	16
Nota APOO usando o padrão de referência									
3	6	3	6	6	4	4	4	3	10

A mesma ponderação para todas as métricas, aqui também, pode não ser a ideal. A diferenciação de pesos poderia ser introduzida em estágios mais avançados do projeto exigindo-se maior rigor na modelagem das classes.

As notas obtidas com o padrão de referência foram, em geral, menores que ao se usar a média dos projetos. Avaliação usando projeto de referência estabelece faixas de aceitação mais estreitas em relação à média. Entretanto, P₆ e P₇, quando avaliados com o projeto de referência obtêm notas maiores. A medida de tendência central não possibilita a avaliação de métricas cujos valores são destoantes da média, mas em um projeto de referência estas mesmas métricas podem estar dentro da faixa de aceitação.

4.2.3 Avaliação com Valores da Literatura

Nesta forma de avaliação o valor de cada métrica do projeto é comparado com valores

mínimo e máximo recomendados por diferentes pesquisadores. No caso de ocorrência de diferentes valores para a mesma métrica, utiliza-se o de maior amplitude por permitir flexibilidade na avaliação do projeto.

Os resultados são obtidos a partir do intervalo de aceitação. Como o limite inferior, para algumas métricas, não é determinado assume-se o valor zero. Os resultados são mostrados na Tabela 5.

Tabela 5 – Avaliação com Valores da Literatura

P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	P ₈	P ₉	PR
Pontos usando valores da literatura:									
8	14	9	11	15	12	15	8	16	15
Nota APOO com valores da literatura									
3	6	4	5	6	5	6	3	7	6

Os valores obtidos apresentam distribuição uniforme. Este fato certamente deriva da maior amplitude da faixa de aceitação obtida na literatura. A avaliação usando valores da literatura é uma tentativa de se verificar se um projeto, classificado por tamanho (classes, atributos e métodos), apresenta valores de métricas aceitáveis frente a diferentes projetos já avaliados por diferentes pesquisadores.

A avaliação é mais flexível e pode não ser a ideal se o grupo se mostra heterogêneo, pois há uma grande variação nos intervalos de tolerância. Entretanto, permite orientar o grupo quanto a distorções detectadas pela ferramenta. Este fato possibilita o avaliador reforçar conceitos ainda não consolidados.

Pode ser observado no gráfico da Figura 5 que os pontos de inflexão demonstram que, salvo as exceções, um projeto bem avaliado por determinado modelo tende a seguir o mesmo processo frente aos demais.

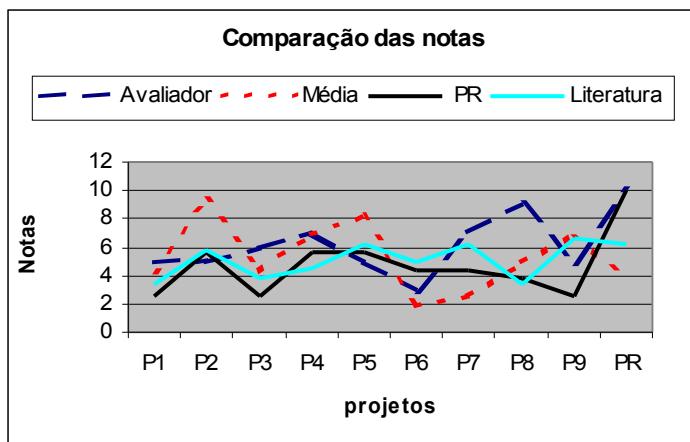


Figura 5 – Comparação dos Resultados.

5 Conclusão e Trabalhos Futuros

A qualidade dos Sistemas de Informação depende fortemente da qualidade do software que lhe dá suporte, que por sua vez, depende da qualidade de seu projeto. Daí a importância de se dedicar estudos visando melhorias na qualidade dos artefatos de projeto com metodologias centradas em métricas de produto e de processo. Além disso, uma boa formação e avaliação crítica por parte do próprio aluno é um fator de melhoria do aprendizado.

Este trabalho apresentou um estudo dos principais modelos de métricas de classes utilizados na avaliação da qualidade de software e a ferramenta APOO para avaliação de projetos modelados sob UML e exportados no formato XMI. Apresentou também estudo de caso envolvendo o uso da ferramenta na extração de métricas de projetos e um modelo de avaliação baseado em medidas de tendência central, projeto de referência e valores da literatura.

Os testes com APOO e a análise das métricas extraídas demonstram que a implementação de uma ferramenta como proposto neste trabalho é uma abordagem possível para o ensino. A solução é viável, por exemplo, como auxílio na avaliação do aprendizado, no meio acadêmico, em disciplinas relacionadas à engenharia de software ou mesmo no desenvolvimento de aplicativos em geral, após introdução de melhorias.

6 Referências

- Harrison, R., Counsell, S. (1998). An Evaluation of the MOOD Set of Object-Oriented Software Metrics, IEEE Transactions on Software Engineering, vol 24, NO 6, JUNE 1998 pp 491-496.
- Basili, V., Briand, L., Melo W. (1996), A Validation of Object-Oriented Design Quality Metrics as Quality Indicators, IEEE Transactions on Software Engineering, vol. 22, no. 10, pp. 751-761, 1996.
- Brito e Abreu, F., Melo, W. (1996) Evaluating the Impact of Object-Oriented Design on Software Quality, Proc. Third International Software Metrics Symposium, pp 90-96, 1996.
- Chidamber, S. R., Kemerer, C. F. (1994), A Metrics Suite for Object Oriented. IEEE Transactions on Software Engineering, vol. 20, No. 6, June 1994, pp. 476-493.
- El-Wakil, M., El-Bastawisi, A., Boshra, M., Fahmy, (2005). A., Object-Oriented Design Quality Models-A survey and Comparison, Faculty of Computers and Information, Cairo University, Cairo, Egito, 2005.
- Genero, M.; Piattini M.; Calero C. (2002). Empirical Validation of Class Diagram Metrics, Proceedings of the 2002 International Symposium on Empirical Software Engineering (ISESE'02)
- Genero, M.; Piattini M.; Calero C. (2005) A Survey of Metrics for UML Class Diagrams Journal of Object Technology, Vol. 4, No 9, November - December, 2005.
- Olague, H. M., Letha, H. E. (2007), Empirical Validation of Three Software Metrics Suites to Predict Fault-Proneness of Object-Oriented Classes Developed Using Highly Iterative or Agile Software Development Process, IEEE Transactions on Software Engineering Vol. 23, No. 6 June 2007 pp. 402-419.