

# Métricas para Medição e Melhoria de Processos de Software

Augusto Gomes, Káthia Marçal de Oliveira, Ana Regina Rocha

COPPE/UFRJ - Programa de Engenharia de Sistemas e Computação  
Caixa Postal 68 511 - Rio de Janeiro - RJ - Brasil - Cep 21945-970  
e-mail: {agomes, kathia, darocha}@cos.ufrj.br

## Abstract

A melhoria do processo de software é um objetivo fundamental para as organizações e deve estar baseado em medições. Entretanto, definir, coletar e analisar um conjunto de métricas não é uma tarefa trivial. Neste artigo descrevemos uma abordagem para definição de métricas, realização de medições e avaliação do processo de software baseada nos seguintes passos: (1) identificação dos objetivos da medição, definição das questões relacionadas a estes objetivos e seleção das métricas adequadas seguindo a abordagem GQM (Goal-Question-Metrics); (2) realização de medições como parte integrante do processo de desenvolvimento; (3) análise dos resultados apoiada em um sistema baseado em conhecimento. As métricas coletadas são utilizadas para: (i) caracterizar o projeto para comparação de resultados; (ii) sugerir melhorias no processo com base no resultado das métricas e (iii) realizar estudos empíricos comparando resultados obtidos em diversos projetos.

## 1. Introdução

Com o intuito de aperfeiçoar o desenvolvimento de software e obter produtos com os níveis desejáveis de qualidade, dentro do cronograma e orçamento propostos, a última década assistiu a uma mudança de enfoque com relação ao processo de software. Tem-se, então, uma nova abordagem na qual o foco principal das atenções está na garantia da qualidade do próprio processo produtivo, visto que este tem se mostrado o fator determinante para o alcance da qualidade do produto final.

A partir desta mudança de foco, intensificou-se a pesquisa sobre o processo de desenvolvimento e várias normas e padrões foram definidos a fim de auxiliar na definição e melhoria de processos de software. Com a intensificação dos estudos,

constatou-se que, para alcançar níveis cada vez mais altos de qualidade, era necessário melhorar cada passo do ciclo de vida de desenvolvimento [1]. Porém, para que isso se tornasse possível, dados quantitativos, que pudessem descrever a realidade do processo, precisavam ser obtidos e devidamente analisados.

Muitas métricas foram, então, propostas e aplicadas em casos práticos a fim de alcançar os seguintes objetivos: i) melhorar o entendimento sobre o processo, produto, recursos e ambiente de desenvolvimento e, assim, estabelecer bases para comparação entre medições; ii) avaliar o andamento do projeto comparando com dados planejados; iii) fazer previsões sobre o futuro andamento do projeto baseado em comportamentos passados; iv) promover melhorias identificando falhas, ineficiências e outras oportunidades para melhorar a qualidade do produto e o desempenho do processo [2].

Porém, ao contrário do que possa parecer, definir, coletar e analisar um conjunto de métricas é uma tarefa custosa que demanda grande conhecimento para evitar que o seu uso não aumente ainda mais os problemas enfrentados durante o desenvolvimento de software. Além disso, propor um conjunto de modificações para o processo a fim de melhorar os resultados obtidos é uma tarefa ainda mais desafiadora.

Este trabalho descreve uma abordagem para medição e melhoria do processo de desenvolvimento de software baseada nos seguintes passos: (1) identificação dos objetivos da medição, definição das questões relacionadas a estes objetivos e seleção das métricas adequadas seguindo a abordagem GQM (Goal-Question-Metrics); (2) realização de medições como parte integrante do processo de desenvolvimento; (3) análise dos resultados apoiada pela construção de um sistema baseado em conhecimento. A construção do sistema baseado em conhecimento foi realizada considerando as métricas definidas e

os possíveis problemas ou causas para resultados não satisfatórios para as mesmas.

O artigo está organizado da seguinte forma. A segunda seção apresenta o passo (1) da abordagem proposta. Na seção seguinte, é discutida a medição do processo, na quarta é apresentado como está sendo desenvolvido o sistema baseado em conhecimento para apoiar a análise dos resultados. Na quinta seção discutimos a utilização de diferentes métricas coletadas para processo de software com o objetivo de avaliar e poder definir melhorias sobre o mesmo. Finalmente, na última seção é apresentado as conclusões desse trabalho.

## 2. Definição dos Objetivos e Métricas

O primeiro passo para a realização deste trabalho foi a seleção de um conjunto de métricas que seriam utilizadas para a avaliação de processos de desenvolvimento de software. Porém, um processo de seleção de métricas não deve ser feito de forma aleatória, pois isto pode dificultar a análise dos resultados devido ao grande volume de dados coletados, além de provocar um aumento do esforço empregado no desenvolvimento com o levantamento destas informações.

Experiências em medições orientadas a objetivos mostraram a importância da definição prévia de metas para facilitar a escolha de métricas e a correta interpretação dos resultados obtidos nas medições. Assim, a abordagem GQM (Goal Question Metric) foi a solução que concluímos ser adequada para resolver este problema. Esta abordagem se baseia na suposição de que para uma organização medir de forma eficiente, é necessário, primeiro, especificar objetivos a serem alcançados, relacionar estes objetivos com dados reais obtidos através de medições e, finalmente, prover um framework para a interpretação destes dados de acordo com os objetivos propostos [3].

Seguindo esta abordagem, levantamos com especialistas os problemas mais comumente enfrentados no desenvolvimento de sistemas de forma a facilitar a definição dos objetivos. Vários foram os problemas citados, porém, três se destacaram como sendo os mais relevantes ao se iniciar uma abordagem de melhoria baseada em medições: (i) falta de precisão das estimativas de projetos; (ii) baixa qualidade dos produtos liberados para uso; e (iii) alto custo dos projetos. A partir destes problemas, definimos que a abordagem proposta teria três objetivos: (1) melhorar a precisão das estimativas de projeto; (2) melhorar a qualidade dos produtos liberados para uso; e, (3) diminuir o custo final dos projetos.

A partir da definição dos objetivos, passamos para a elaboração de questões a serem investigadas a fim de analisar se as metas definidas foram devidamente alcançadas. Para cada uma destas questões, deveria ser viável a coleta de valores quantitativos que representassem corretamente a realidade enfrentada no decorrer do desenvolvimento. Foram, então, selecionadas algumas questões e suas respectivas métricas, como pode ser visto nas figuras 1, 2 e 3. Na figura 2, consideramos Erro qualquer problema encontrado em uma revisão na fase em que este foi gerado, Modificação é um problema encontrado em um documento em fases posteriores a sua aceitação e o Tamanho do Sistema é medido em número de

Objetivo 1: Melhorar a precisão das estimativas de projeto.

Questão 1.1: Qual é a precisão das estimativas de cronograma?

Métrica 1.1a) Precisão da estimativa de cronograma de todo o projeto  
$$\text{Precisão} = \frac{\text{Tempo real de todo o projeto}}{\text{Tempo estimado do projeto}}$$

Métrica 1.1b) Precisão da estimativa de cronograma por fase do projeto  
$$\text{Precisão} = \frac{\text{Tempo real de cada fase do projeto}}{\text{Tempo estimado para a fase}}$$

Questão 1.2: Qual é a precisão das estimativas de esforço?

Métrica 1.2a) Precisão da estimativa de esforço de todo o projeto  
$$\text{Precisão} = \frac{\text{Esforço real de todo o projeto}}{\text{Esforço estimado para o projeto}}$$

Métrica 1.2b) Precisão da estimativa de esforço por fase do projeto  
$$\text{Precisão} = \frac{\text{Esforço real de cada fase do projeto}}{\text{Esforço estimado para a fase}}$$

Figura 1 – Métricas para o objetivo melhorar precisão de estimativas

Objetivo 2: Melhorar a qualidade dos produtos liberados para uso.

Questão 2.1: Qual a densidade de defeitos encontrados nos produtos antes da liberação para uso?

Métrica 2.1) Densidade de Defeitos.  
$$\text{Densidade} = \frac{\text{Número de erros e modificações}}{\text{Tamanho do sistema}}$$

Questão 2.2: Qual o percentual de deterioração do software?

Métrica 2.2) Percentual de deterioração do software.

Figura 2 – Métricas para o objetivo qualidade de produtos

linhas de código (exceto as em branco e as contendo somente comentários).

Além destas, outras métricas foram selecionadas para serem utilizadas na análise e melhoria do processo como será visto nas próximas seções deste artigo. Vale ressaltar que todas as métricas utilizadas aqui e no decorrer deste trabalho foram extraídas ou adaptadas da literatura [4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. Sabemos, no entanto, que este está longe de ser o melhor conjunto de métricas para a avaliação de qualquer projeto, nem é este nosso objetivo, porém, é um conjunto que acreditamos ser válido para colocarmos esta idéia em prática e, a partir dos resultados obtidos, poder melhorá-lo.

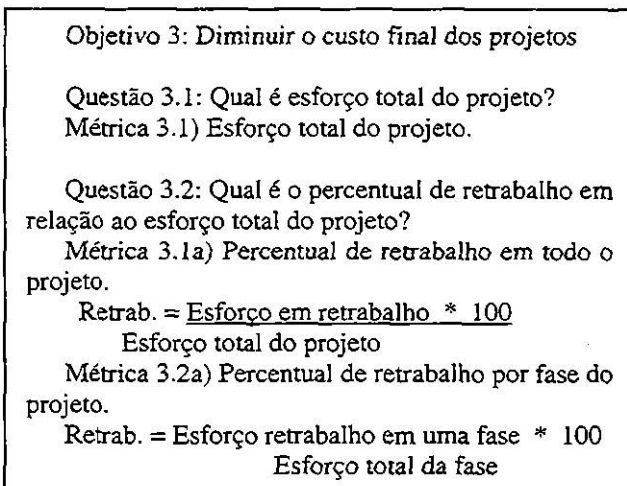


Figura 3 – Métricas para o objetivo diminuir custos dos projetos

### 3. Medição do Processo

Para tornar possível a medição e a melhoria de um processo de software, este deve ser definido de forma clara e precisa a fim de evitar problemas na sua interpretação e deve ser devidamente executado pela equipe de desenvolvimento para que os valores coletados sejam válidos para a sua avaliação.

A tarefa de definição do processo de software se inicia pela elaboração de um processo bastante genérico de forma a tornar possível sua especialização para o desenvolvimento de diferentes tipos de software (como, por exemplo, software para Web ou com orientação a objetos) e para diferentes culturas organizacionais. A definição deste processo (chamado de processo padrão) permite, também, sua instanciação para projetos específicos, considerando-se as características particulares de cada projeto de desenvolvimento. Esta instanciação pode ser feita diretamente a partir do processo padrão ou a partir de especializações já existentes para tipos de software e empresas específicas. O modelo para definição de processos utilizado está ilustrado na figura 4. Mais detalhes sobre esta abordagem para definição de processos pode ser encontrada em [14].

Assim, a partir da definição do processo, adaptamos alguns dos documentos já previsto de serem elaborados durante o desenvolvimento do projeto de forma que todos os dados necessários para a análise do processo fossem facilmente obtidos destes documentos. Desta forma, a atividade de medição passou a ser parte integrante do próprio processo de desenvolvimento o que minimiza o tempo gasto na coleta de dados e nos permite separar as atividades de desenvolvimento das atividades de análise e melhoria do processo. Esta é uma característica muito importante, pois permite que a equipe de desenvolvimento possa concentrar seus esforços na realização das atividades de produção enquanto as atividades de análise da qualidade podem ser feitas por uma equipe especializada. Podemos citar como exemplos: o registro das datas dos marcos importantes no desenvolvimento em documentos de acompanhamento de projetos; a contagem do número de erros e modificações nos laudos de reuniões de inspeção; e o controle de tempo gasto em atividades de desenvolvimento.

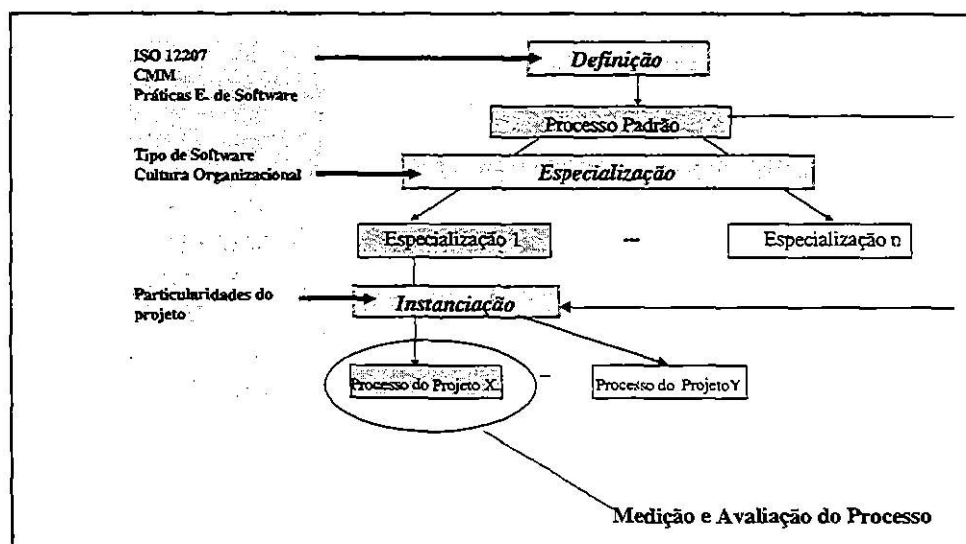


Figura 4: Modelo para Definição de Processos de Software

#### 4. Análise dos Resultados das Medições

Neste ponto da abordagem temos os objetivos definidos e o processo de coleta de dados integrado ao próprio processo produtivo. Porém, ainda nos falta definir como analisar o resultado final das medições e, a partir desta análise, sugerir modificações e melhorias para o processo utilizado. É importante enfatizar que neste trabalho o objetivo principal não é medir o processo, mas sim encontrar possibilidades para sua melhoria sendo utilizado, para isso, medições apenas como uma ferramenta de orientação [6]. Assim, quando o custo necessário para a coleta de uma métrica suplantar seu benefício para o processo de melhoria, esta métrica será excluída do conjunto a ser coletado, pois sua finalidade básica foi violada.

Porém, a análise dos resultados obtidos através de medições é um processo que demanda tempo e um conhecimento profundo sobre processo, métricas e melhoria. Além disso, sabe-se que boa parte das empresas não possui em sua equipe pessoal capacitado para desempenhar tal tarefa. Decidimos, portanto, pela construção de um sistema capaz de apoiar a análise e julgamento dos resultados das medições, identificar aspectos do processo que necessitem de melhoria e, quando possível, sugerir modificações para corrigir seus problemas e deficiências. Esse sistema utiliza como entrada os valores das métricas definidas na seção 2 coletadas de um projeto específico. A partir desses valores é feita uma análise dos possíveis problemas que levaram a resultados não satisfatórios para um dos objetivos definidos (ver seção 2).

A construção deste sistema especialista está sendo feita a partir do conhecimento obtido de especialistas em relação à interpretação dos resultados obtidos nas medições e da relação entre estes resultados e aspectos do processo utilizado no desenvolvimento. O resultado é um conjunto de recomendações para melhoria do processo.

Com este objetivo, procuramos obter o conhecimento sobre os possíveis problemas existentes no processo quando resultados ruins para as métricas são detectados. Sempre que um valor não aceitável para o resultado de uma métrica for obtido, alguma inferência deverá ser feita em relação às características do processo.

Para validar esta idéia, tomamos o primeiro objetivo definido no primeiro passo desta abordagem (melhorar a precisão das estimativas) e questionamos os especialistas quais as possíveis causas para a obtenção de um valor não aceitável

para a métrica de precisão de cronograma, ou seja, o que pode ter levado o projeto a atrasar (métrica definida para o objetivo 1 na seção 2). Após definir as possíveis causas diretas deste problema, relacionamos estas causas com métricas que pudessem descrevê-las de forma quantitativa. Por exemplo, podemos citar como causa de atraso no cronograma um alto índice de retrabalho que é avaliado pela métrica percentagem de retrabalho. Esta metodologia segue a abordagem GQM procurando encontrar métricas para analisar cada um dos sub-objetivos que possam auxiliar na análise dos objetivos principais. Porém, nesta proposta estamos encadeando os objetivos de maneira hierárquica onde os níveis inferiores definem causas para problemas encontrados nas métricas dos níveis superiores [15]. Para cada um dos sub-objetivos, repetimos esta pesquisa até encontrarmos causas que não pudessem ser mais bem detalhadas (ou que não era de nosso interesse detalhar neste estágio em que se encontram os trabalhos). Na figura 5, pode ser visto, a título de exemplo, um grafo gerado seguindo esta metodologia. Nesse grafo, apresentamos que possíveis causas de problemas na precisão da estimativas (elipse na parte superior da figura) podem ser: problema com a forma das estimativas, problemas de baixo desempenho de pessoal ou muito retrabalho nas fases de desenvolvimento. Cada um desses problemas por sua vez pode ter sido gerado por várias razões. A causa do baixo desempenho de pessoal, por exemplo, pode ser pelo fato do pessoal ser mal preparado ou porque houve alta rotatividade de pessoal. Finalmente a causa do pessoal ser mal preparado pode ser pela falta de experiência no domínio da aplicação, na ferramenta, no método, no processo ou mesmo na linguagem de programação. Quando desejarmos fazer a análise dos resultados de métricas de um determinado projeto vamos avaliar os valores de métricas de cada uma dessas causas e assim identificar qual ou quais foram os problemas para os resultados não satisfatórios para um determinado objetivo.

É importante observar que, para cada nova métrica introduzida na construção da base de conhecimento do sistema especialista, é necessária a readaptação dos documentos do processo como explicado anteriormente na seção 3. Ou seja, todos as métricas utilizadas no sistema especialista estão sendo coletados dos próprios documentos do processo. A inclusão de uma nova métrica para análise no sistema implica em definirmos de onde vamos coletar os valores a serem avaliados.

Para que seja possível a construção do sistema baseado em conhecimento utilizando um grafo

como o descrito anteriormente, ainda é necessário definir para cada nó do grafo que foi representado com uma métrica, o que deve ser considerado um valor ruim ou não aceitável para o resultado da medição. Este procedimento deve, de alguma forma, permitir que empresas com realidades diferentes possam utilizar valores distintos que melhor se adequem às suas condições e características, além das próprias características do projeto (ex: sistemas médicos necessitam uma confiabilidade maior que jogos ou aplicativos simples).

Estamos resolvendo este problema utilizando a parametrização destes valores de forma a permitir que os usuários do sistema possam defini-los no início de cada projeto. Essa é uma alternativa interessante, pois valores cada vez melhores podem ser utilizados de maneira a permitir uma constante melhoria no processo de desenvolvimento e

Com os possíveis problemas definidos, passamos para a etapa final da elicitação de conhecimento com os especialistas que, neste ponto, deveriam sugerir, para cada um dos problemas encontrados, modificações para corrigir e melhorar o processo utilizado.

Como pôde ser visto, a abordagem proposta abrange desde a definição das métricas para a análise da melhoria do processo, integração destas ao próprio processo produtivo até a análise dos resultados das medições e sugestão de melhoria para o processo.

## 5. Ampliando a Avaliação do Processo de Software

O conhecimento utilizado na construção da base do sistema especialista se baseia somente em

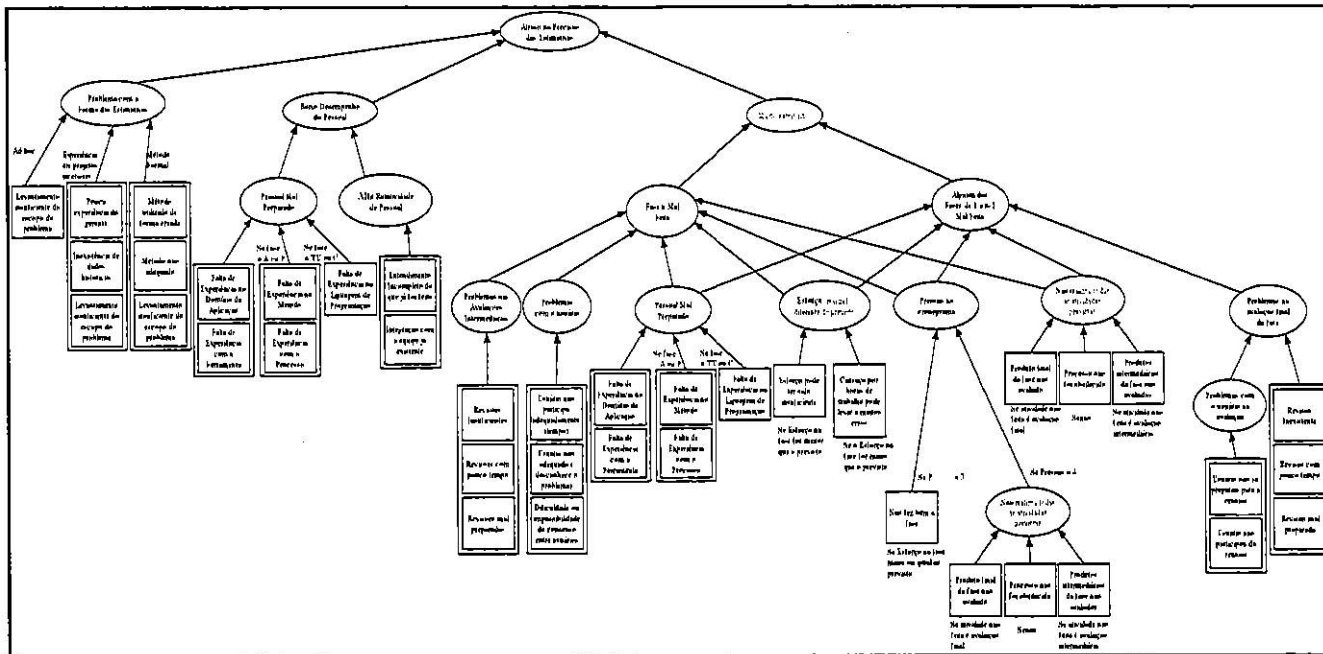


Figura 5: Exemplo do grafo obtido com o conhecimento elicitado de especialistas.

possibilitando o estabelecimento de metas a serem alcançadas.

Quando possível, é interessante que um primeiro projeto seja medido para avaliar qual o nível em que se encontram estes valores de forma a facilitar a definição dos parâmetros. Outra orientação que é passada para os usuários é a de valores extraídos da literatura, como exemplo, podemos citar a proposta de [16] que define que um valor não aceitável para a métrica rotatividade de pessoal seria acima de 5%.

entrevistas feitas com especialistas no assunto. No entanto, tomou-se o cuidado de selecionar as informações que pudessem ser utilizadas por uma variedade maior de projetos. Desta forma, na construção do sistema especialista não foram incluídas, por exemplo, regras que pudessem ser específicas de um determinado domínio ou empresa. Contudo, este ganho em generalidade implicou em uma sensível perda de poder de análise, pois regras que não pudessem ser utilizadas em projetos com determinadas características,

mesmo que pertinentes para a maioria, não poderiam compor a base de conhecimento.

Com a finalidade de contornar mais este problema, foi selecionado um segundo grupo de métricas que seriam utilizadas para definir "tipos" de projetos contendo características similares. Este grupo de métricas atualmente é composto por: tamanho do sistema (em número de linhas de código, exceto as em branco e as contendo somente comentários), tempo total de desenvolvimento e experiência da equipe (obtida através de um questionário). Acredita-se que projetos de um mesmo porte em tempo e tamanho e com equipes em um mesmo nível técnico, normalmente enfrentam problemas similares, daí utilizarmos estas três características para classificar diferentes projetos. Porém, sabemos que pode ser necessária a inclusão de outras métricas ou características para melhorar a classificação dos projetos e, desta forma, será permitida a modificação deste conjunto quando necessário.

Além deste segundo grupo utilizado para identificar o que consideramos "tipos" de projetos, foi ainda definido um terceiro para o qual serão selecionadas métricas que se pretende estudar melhor seu comportamento e relacionamento com as demais. Assim, será construída uma base de dados de métricas contendo dados obtidos em diferentes projetos de forma a tornar possível a realização de estudos empíricos neste banco de dados comparando os resultados obtidos nos diferentes projetos de um mesmo "tipo". À medida que esta base aumentar, relações entre métricas poderão se mostrar pertinentes e poderão orientar no estabelecimento de novas regras a serem inseridas na base de conhecimento do sistema especialista. Assim, conhecimentos específicos de determinadas áreas poderão compor a base do sistema especialista melhorando a qualidade das inferências e permitindo que respostas cada vez mais precisas possam ser obtidas e, desta forma, será possível orientar melhor os desenvolvedores de sistemas. Somente para exemplificar este processo, podemos citar como exemplo a análise da existência de relacionamento entre o conhecimento do domínio e a qualidade da especificação de requisitos gerada na fase de análise do problema.

## 6. Conclusão

Com o aumento da competitividade observado no mercado mundial, a melhoria da qualidade dos produtos de software passou a ser não mais um diferencial para a empresa desenvolvedora, mas sim, um fator crítico para sua sobrevivência. Desta forma, a tentativa de encontrar abordagens que

possibilitem a melhoria do processo de desenvolvimento tem sido uma constante tanto na academia quanto na indústria.

Neste artigo apresentamos uma abordagem que inclui uma forma de escolha adequada de métricas, sua coleta de forma integrada ao próprio processo produtivo, análise dos resultados e sugestões para a melhoria do processo de software baseado no conhecimento extraído de especialistas.

Atualmente, os trabalhos estão concentrados na implementação em prolog do sistema baseado em conhecimento que será utilizado para apoiar a etapa de análise proposta nesta abordagem. O sistema final deverá ser integrado ao ambiente de definição de processos, o TABA [17], de forma a compor a análise da qualidade dos processos definidos nesta ferramenta.

## Referências

- [1] Oman P., Pflieger S.L., *Applying Software Metrics*. Los Alamitos, CA, IEEE Computer Society.
- [2] Park R.E., Goethert W.B., Florac W.A., *Goal-Driven Software Measurement - A Guidebook*, CMU/SEI-96-HB-002, Pittsburgh, PA, Software Engineering Institute, Carnegie Mellon University, August 1996.
- [3] Basili V.R., Caldiera G., Rombach H.D., *Goal Question Metric Paradigm*, Encyclopedia of Software Engineering, 2 Volume Set, John Wiley & Sons, Inc, 1994.
- [4] Carleton A.D., Park R.E., Goethert W.B., Florac W.A., Bailey E.K., Pflieger S.L., *Software Measurement for DoD Systems: Recommendation for Initial Core Measures*, CMU/SEI-92-TR-19, ESC-TR-92-19, Pittsburgh, PA, Software Engineering Institute, Carnegie Mellon University, September 1992.
- [5] Conte, S.D., Dunsmore, H.D., Shen, V.Y., *Software Engineering Metrics and Models*, Benjamin-Cummings, Menlo Park, CA, 1986 in [FENTON 97]
- [6] Daskalantonakis M.K., *A Practical View of Software Measurement and Implementation Experiences Within Motorola*, IEEE Transaction Software Engineering, Vol 18 No. 11, November 1992 in [OMAN 97].
- [7] Fenton N.E., Pflieger S.L., *Software Metrics - A Rigorous & Practical Approach*, Second Edition. Boston, MA, PWS Publishing Company, 1997
- [8] Florac W.A., *Software Quality Measurement: A Framework for Counting Problems, Failures and Faults*, CMU/SEI-92-TR-22, ESC-TR-92-22, Pittsburgh, PA, Software Engineering Institute, Carnegie Mellon University, September 1992.
- [9] Florac W.A., Park R.E., *Practical Software Measurement: Measuring for Process Management and Improvement*, CMU/SEI-97-HB-003, Pittsburgh, PA, Software Engineering Institute, Carnegie Mellon University, April 1997.
- [10] Goethert W.B., Bailey E.K., Busby M.B., *Software Effort & Schedule Measurement: A Framework for Counting Staff-hours and Reporting Schedule*

*Information*, CMU/SEI-92-TR-21, ESC-TR-92-21, Pittsburgh, PA, Software Engineering Institute, Carnegie Mellon University, September 1992.

[11] Park R.E., *Software Size Measurement: A Framework for Counting Source Statements*, CMU/SEI-92-TR-20, ESC-TR-92-20, Pittsburgh, PA, Software Engineering Institute, Carnegie Mellon University, September 1992.

[12] Roberts M.A., *Experiences in Analyzing Software Inspection Data*, Software Engineering Process Group, McDonnell Douglas Aerospace, McDonnell Douglas Corporation, St. Louis, MO, May 1996 in [FLORAC 97]

[13] Tajima D., Matsubara T., The Computer Software Industry in Japan, IEEE Computer, 14(5), 1981 in [FENTON 97].

[14] Machado, L.F.C. et al DEF-PRO: Uma Ferramenta para Apoiar a Definição do Processo Padrão, XI CITS, Curitiba, Junho 2000

[15] Fenton N.E., Neil, M., *Software Metrics: Roadmap*, Computer Science Department, Queen Mary and Westfield College, London, UK, 2000.

[16] *Software Program Managers Network*, 207, Department of the Navy (US), January 1995 in [7]

[17] Oliveira K.M., Rocha A.R., Travassos G., Menezes C.; Using Domain-Knowledge in Software Development Environments, Software Engineering and Knowledge Engineering - SEKE 99; Kaiserslautern, Germany, July 1999.