

Avaliação de Processos de Software na Estação Taba

Augusto Gomes, Sômulo Mafra, Kathia Oliveira, Ana Regina Rocha
COPPE/UFRJ - Programa de Engenharia de Sistemas e Computação
agomes@cos.ufrj.br

Resumo

A melhoria do processo de software é um objetivo fundamental para as organizações e deve estar baseada em medições. Entretanto, definir, coletar e analisar um conjunto de métricas não é uma tarefa trivial. Neste artigo, descrevemos uma ferramenta capaz de apoiar a avaliação de processos de software. Esta ferramenta abrange desde a coleta dos dados para cada uma das métricas que serão utilizadas na avaliação do processo até a sugestão dos possíveis problemas que a equipe de desenvolvimento pode ter enfrentado no decorrer dos trabalhos.

Abstract

Software process improvement is a common goal and it should be based on measurement. However, the definition, collection and analysis of a group of metrics is a no trivial task. This paper describes a tool for evaluating the software process. It includes since the collection of data for each metric used until the suggestion of possible problems that the development team could have faced during the project.

1 - Introdução

Com o intuito de aperfeiçoar o desenvolvimento de software e obter produtos com os níveis desejáveis de qualidade a última década assistiu a uma mudança de enfoque com relação à garantia da qualidade. 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 um dos fatores determinantes 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 estabelecidos 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, mas poucos foram os casos onde o seu uso não foi considerado ineficiente ou, até mesmo, desaconselhável. Fenton [2] defende a idéia de que a razão para esta falta de sucesso pode ser atribuída ao fato de que as atividades de medição não tiveram como objetivo o principal requisito, que é prover as informações necessárias para apoiar as decisões gerenciais durante o ciclo de vida do software. Além disso, também argumenta que as metodologias tradicionais geralmente são orientadas por modelos de regressão para estimativa de custo e de defeitos o que provê pouca informação de caráter decisório para os gerentes.

Este trabalho descreve uma ferramenta capaz de apoiar os desenvolvedores na avaliação de processos de software. Esta ferramenta abrange desde a coleta dos dados para cada uma das métricas utilizadas até a análise dos resultados com a sugestão dos possíveis problemas que a equipe de desenvolvimento pode ter enfrentado durante o decorrer dos trabalhos. Esta ferra-

menta foi desenvolvida a partir da infra-estrutura já existente da Estação Taba, projeto no qual este trabalho está inserido.

2 - Medição de Processo

A Estação TABA é um meta-ambiente capaz de gerar, a partir da definição de um processo, ambientes de desenvolvimento de software (ADS) adequados às particularidades de projetos específicos. Com o intuito de garantir que o procedimento de medição seja feito de forma consistente, o processo definido na Estação Taba foi adaptado de forma que todos os dados necessários para a sua análise fossem facilmente obtidos de documentos que deveriam ser preenchidos pela equipe durante o decorrer dos trabalhos. 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, ou, como será visto mais à frente neste trabalho, por um sistema baseado no conhecimento extraídos de especialistas no assunto.

Porém, por mais precisas que sejam as definições das métricas utilizadas na avaliação do processo, ambigüidades podem aparecer tanto no momento da coleta dos dados quanto na avaliação dos resultados. Desta forma, um conjunto de funcionalidades capaz de apoiar e orientar o levantamento e a análise de dados foi incorporado aos ambientes de software instanciados pela Estação Taba. Este projeto está sendo desenvolvido utilizando o Microsoft Visual C++ para Windows.

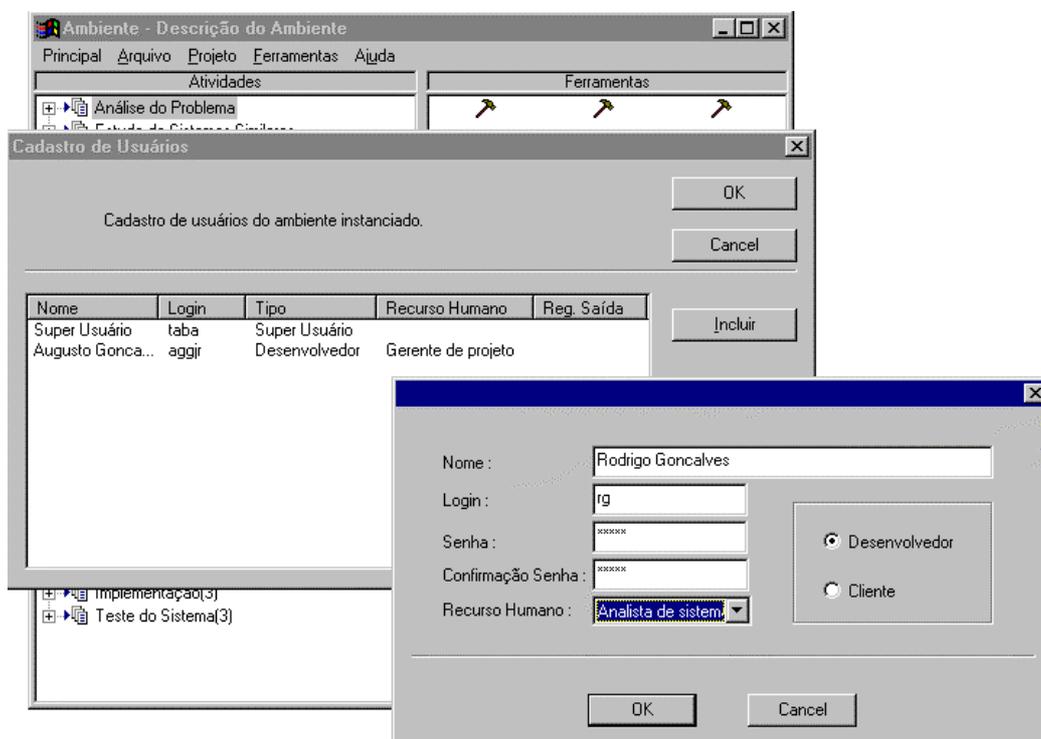


Figura 1: Cadastro dos usuários do ambiente instanciado pela Estação Taba.

Cada um dos usuários do sistema deve ser caracterizado por um dos tipos de recursos humanos já definidos na Estação Taba, como, por exemplo, Analista de sistemas, Gerente ou Programador. Este tipo de informação é importante para que o ambiente possa ser configura-

do de forma coerente ao papel desempenhado por cada membro da equipe. Somente a título de exemplo, fica a cargo do gerente do projeto a definição das estimativas de tempo e esforço para cada uma das macro-atividades do processo e, desta forma, somente este terá acesso a tela de cadastro de estimativas do projeto (figura 2).

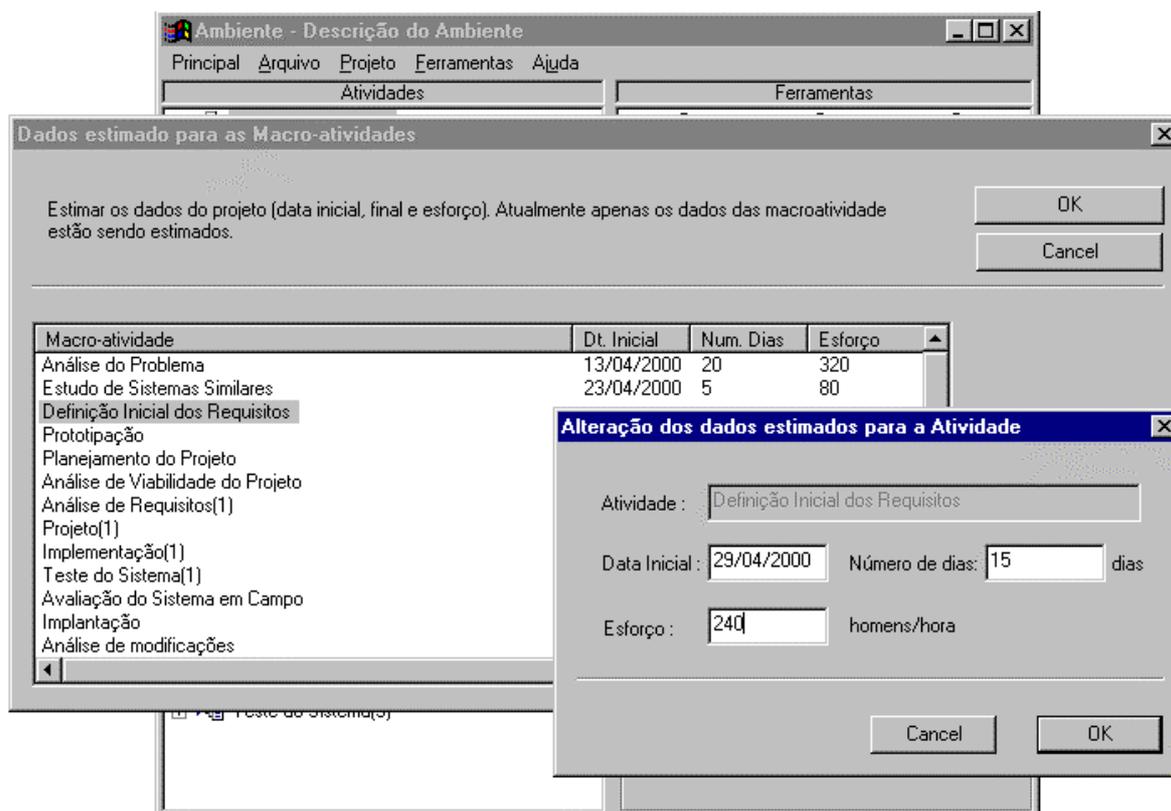


Figura 2 – Cadastro das estimativas de tempo e esforço para cada macro-atividade do processo.

Depois de cadastrados, todos os usuários deverão documentar as atividades executadas durante o decorrer dos trabalhos. Para isso, novas funcionalidades foram incorporadas ao ambiente de forma a tornar possível a coleta de inúmeras informações, como, por exemplo, número de erros e modificações, tempo e esforço empregados por cada membro da equipe, datas de aprovação das macro-atividade, etc.

Para exemplificar, na figura 3 pode ser vista uma planilha de atividades parcialmente preenchida por um membro da equipe. Esta planilha deve ser preenchida por cada participante do projeto para que os dados referentes às atividades executadas possam ser corretamente armazenados. Com estas informações, o sistema será capaz de calcular as métricas de esforço, tempo e re-trabalho em todo o projeto e em cada uma de suas macro-atividades.

3 - Avaliação de Processo

Ao término do projeto, os dados referentes a cada uma das métricas poderão ser reunidos e os resultados das medições poderão ser calculados. A partir disso, foi utilizado o conhecimento de especialistas para criar uma estrutura de decisão capaz de encontrar os possíveis problemas que a equipe de desenvolvimento pode ter enfrentado durante o decorrer dos trabalhos. Esta estrutura foi construída seguindo a abordagem *GQM (Goal/Question/Metric)* proposta por Basili [3] a qual procura detalhar objetivos definidos para a avaliação do processo em sub-objetivos e estes, por sua vez, em novos sub-objetivos até não ser mais possível ou

interessante o detalhamento. Desta forma, esta é uma estrutura hierárquica na qual os níveis inferiores determinam causas para problemas encontrados em níveis superiores. Os problemas para cada um dos nós da estrutura são determinados pelos resultados das métricas selecionadas.

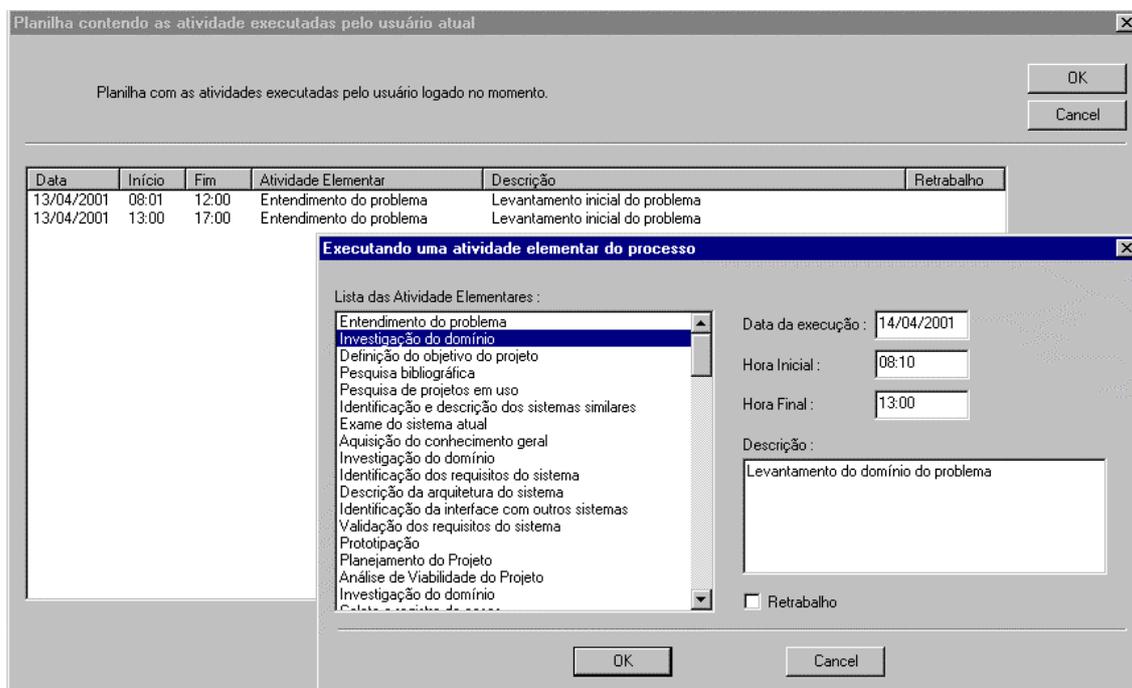


Figura 3: Planilha de Atividades parcialmente preenchida por um membro da equipe.

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 utilizado. Para definir o que é um valor não aceitável para o resultado de uma métrica, várias técnicas foram analisadas e o uso de tolerâncias definidas pelo usuário do sistema foi escolhido. Esta é uma alternativa interessante, pois valores mais próximos da realidade da empresa podem ser utilizados e parâmetros cada vez menores permitem uma contínua melhoria do processo.

A partir dos conhecimentos obtidos com os especialistas, foram estabelecidos, então, três objetivos para o procedimento de avaliação do processo: (i) melhorar a precisão das estimativas de projeto; (ii) aumentar a qualidade dos produtos liberados para uso; (iii) diminuir o custo final dos projetos. Por questões de tempo, foi construída somente a estrutura para a análise do processo segundo o primeiro objetivo definido para a avaliação.

Assim, cabe ao usuário do sistema, apenas, determinar qual objetivo será utilizado para a avaliação (atualmente somente o primeiro foi implementado) e quais valores devem ser utilizados para as tolerâncias das métricas que necessitarem deste tipo de informação (figura 4).

Definido o objetivo, o ambiente instanciado passa os resultados referentes a cada uma das métricas utilizadas na estrutura de decisão daquele objetivo para uma ferramenta construída em Prolog que, baseada na estrutura de decisão definida anteriormente, analisa os dados e informa as causas mais prováveis para os problemas ocorridos no alcance deste objetivo. Como pode ser visto no exemplo da figura 5, a resposta do sistema é dada em uma forma de árvore de diretórios onde os nós mais internos procuram definir os problemas encontrados nos nós mais externos. Neste exemplo gerado para a análise do primeiro objetivo, o problema encontrado na "Precisão das Estimativas de Projeto" pode ser explicado pelo problema encon-

trado na “Precisão da Estimativa de Cronograma” o qual, por sua vez, pode ser explicado pelos problemas ocorridos nas estimativas de cronograma na macro-atividade de “Projeto” e assim sucessivamente.

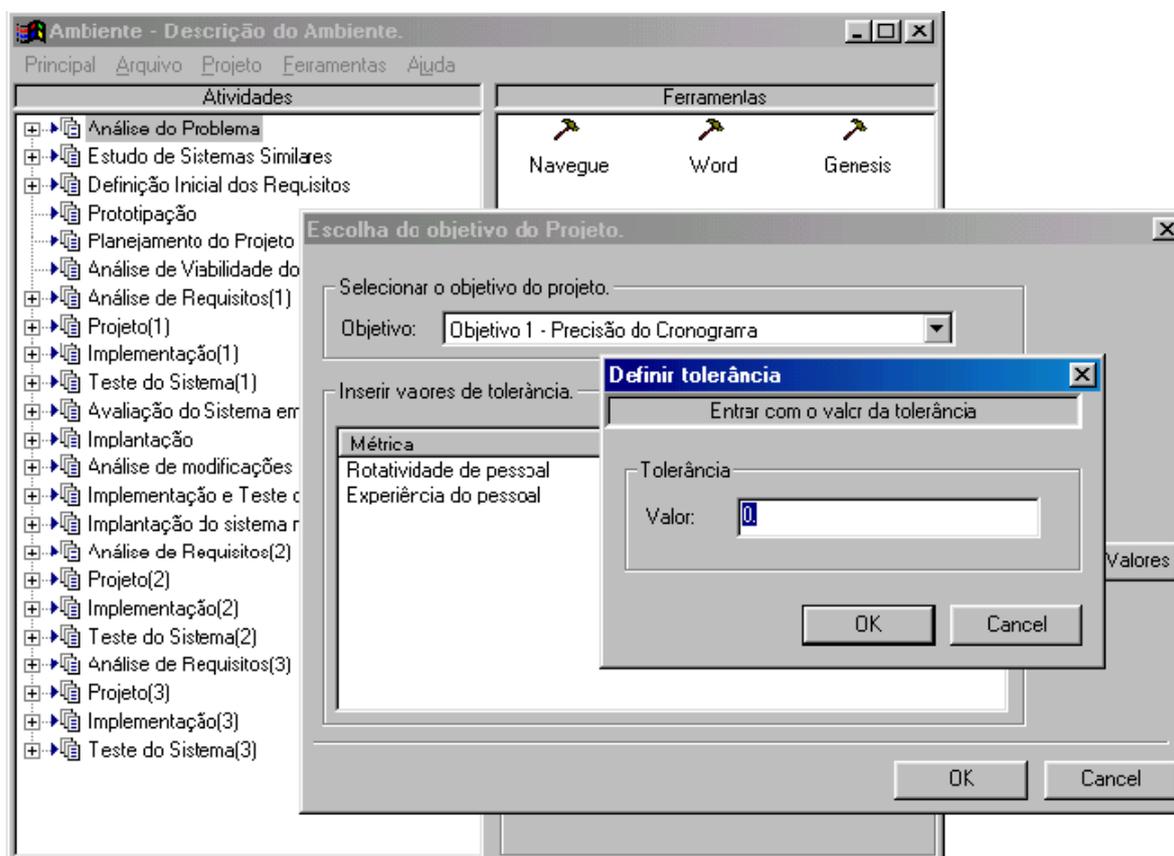


Figura 4: Definição do Objetivo da Avaliação e dos Parâmetros que serão utilizados na avaliação do processo de software.

Desta forma, o sistema tem, como entrada, os dados referentes a cada uma das métricas necessárias para a análise do objetivo selecionado (coletados durante todo o decorrer do projeto) e informa, como saída, uma lista dos possíveis problemas encontrados no processo e suas respectivas causas.

A partir dos principais problemas encontrados pela ferramenta, gerentes e engenheiros de software saberão onde devem concentrar seus esforços para corrigir problemas e, quando se mostrar pertinente, modificar o processo de desenvolvimento utilizado durante este projeto específico para que estes problemas não sejam novamente enfrentados em trabalhos futuros.

Com isto, acreditamos que as organizações terão uma ferramenta capaz de orientar de forma semi-automatizada o procedimento de avaliação dos processos de software.

4 - Conclusão

O uso de métricas de software é uma poderosa ferramenta explorada em inúmeros trabalhos, mas a maioria das abordagens não obteve o êxito esperado. Alguns pesquisadores argumentam que o problema não está com o uso das métricas, mas sim com a forma como estas têm sido utilizadas. O seu uso combinado pode ser uma poderosa ferramenta, ainda pouco explorada atualmente.

Neste artigo, apresentamos uma ferramenta capaz de apoiar de forma semi-automática os gerentes e engenheiros de software em todo o procedimento de avaliação de processos de software baseada no uso de medições, abrangendo desde a coleta de dados até a análise dos resultados apoiada por um sistema baseado em conhecimento.

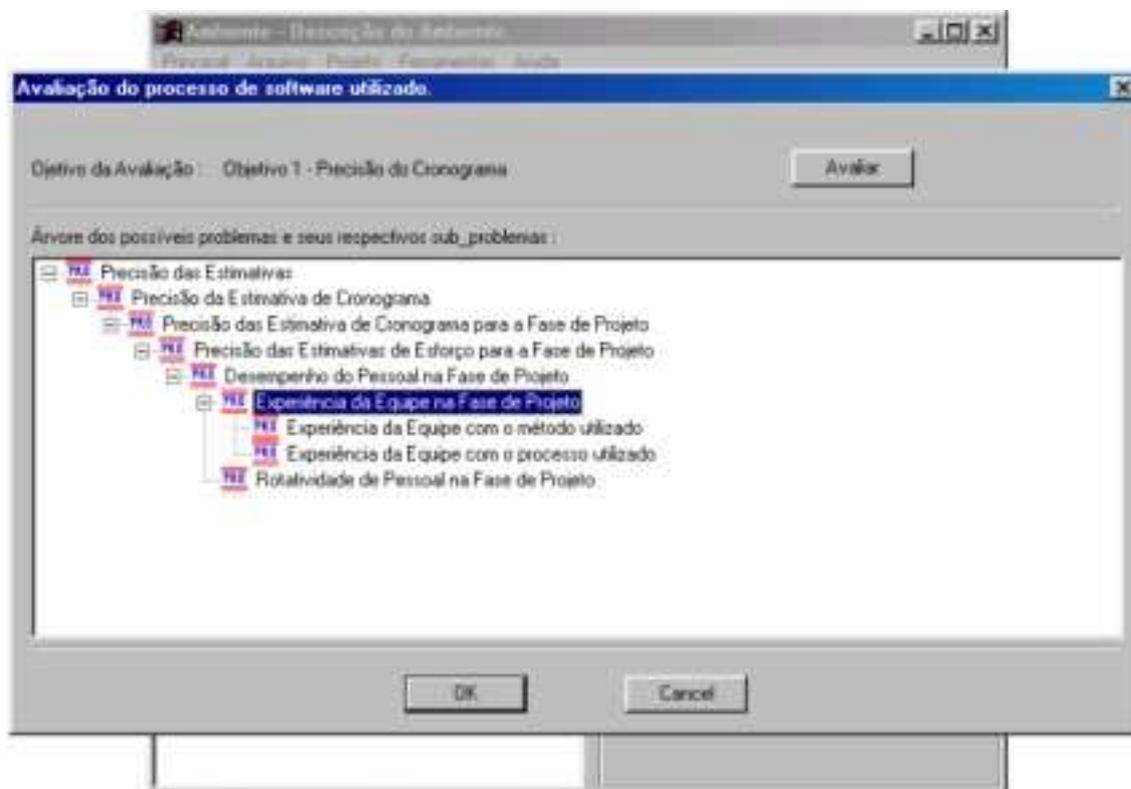


Figura 5: Resposta final do sistema baseado em conhecimento gerada para a avaliação do processo segundo o primeiro objetivo definido.

Referências Bibliográficas

- [1] Oman P., Pfleger S.L., *Applying Software Metrics*, Los Alamitos, CA, IEEE Computer Society.
- [2] Fenton N.E., Neil, M., *Software Metrics: Roadmap*, Computer Science Department, Queen Mary and Westfield College, London, UK, 2000.
- [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] Machado, L.F.C. et al DEF-PRO: Uma Ferramenta para Apoiar a Definição do Processo Padrão, XI CITS, Curitiba, Junho 2000
- [5] 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.