

Avaliação de Processos de Software Baseada em Medições

Augusto Gomes, 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 abordagem para avaliação de processos de software que define como selecionar métricas adequadas seguindo a abordagem GQM (Goal-Question-Metrics), estabelece a realização de medições como parte integrante do processo de desenvolvimento e propõe a análise dos resultados apoiada em um sistema baseado em conhecimento.

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 software process evaluation approach that defines an adequate choice of metrics based on GQM (Goal-Question-Metric), incorporate measurement as an integral part of the software development process and define a knowledge based system to analyze the measurements results.

1. Introdução

Ao longo das últimas décadas, os produtos de software têm sofrido um considerável crescimento de tamanho e complexidade e, à medida que isto ocorre, o número de problemas enfrentados durante o desenvolvimento também aumenta. Com o intuito de 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 no próprio processo produtivo, visto que este tem se mostrado o fator determinante para o alcance da qualidade do produto final.

Para alcançar níveis cada vez mais altos de qualidade torna-se necessário melhorar cada etapa do ciclo de vida do software [1] e, para tornar isto possível, dados quantitativos que descrevam a realidade do processo precisam ser obtidos e devidamente analisados. Neste contexto, medições de software têm se mostrado o fator chave para o aumento da qualidade dos processos, pois são a base para a identificação de suas forças e fraquezas facilitando a visualização das oportunidades de melhoria.

Entretanto, o uso de métricas apenas provê dados referentes às entidades do mundo real envolvidas na execução dos projetos. Serão as ações tomadas, a partir do resultado obtido que farão com que a qualidade dos produtos e o desempenho do processo aumentem. Porém, para que uma abordagem de melhoria de processos baseada no uso de medições obtenha êxito, a escolha das métricas, sua definição consistente, o correto levantamento dos dados e a elaboração de um mecanismo de análise dos resultados, devem ser devidamente estudados.

Neste artigo apresentamos uma abordagem que envolve desde a seleção e definição de métricas até a análise dos resultados obtidos com a indicação dos possíveis problemas enfrentados pela equipe de desenvolvimento durante o decorrer dos trabalhos e que podem ter afetado o desempenho do processo ou sido afetados pelo mesmo. Os problemas identificados servirão de base para que os desenvolvedores possam encontrar falhas na execução do processo e propor melhorias para otimizar o seu desempenho.

Nas seções seguintes faremos uma breve descrição sobre métricas na avaliação de processos de desenvolvimento de software (seção 2), apresentaremos uma abordagem para avaliação de processos (seção 3) e uma ferramenta definida para apoiar essa avaliação (seção 4). Na seção 5, será apresentado o uso desta abordagem em um projeto real e na última seção serão apresentadas algumas conclusões (seção 6).

2. Medição de Processo de Software

Assim como qualquer outra disciplina de engenharia, o desenvolvimento de software requer um mecanismo de medição para obter as informações sobre a execução do processo necessárias para o seu correto entendimento e avaliação [2]. Medir é o processo através do qual números ou símbolos são atribuídos a características das entidades do mundo real de forma a tornar possível caracterizar cada entidade através de regras claramente definidas [3]. O uso de métricas nos ajuda a entender e a interagir com o mundo para, então, poder melhorá-lo.

Muitas métricas foram 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 [4].

Porém, ao contrário do que possa parecer, definir, coletar e analisar um conjunto de métricas não é uma tarefa trivial. Na realidade, esta 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 acordo com Park [4], este problema ocorre devido ao grande número de atributos que podem ser medidos durante um projeto. Uma escolha incorreta das métricas pode levar, além do próprio aumento desnecessário de esforço, a uma visão distorcida do processo o que dificulta a sua análise e, muitas vezes, termina por orientar decisões equivocadas.

Experiências com medições orientadas a objetivos mostraram a importância da definição prévia de metas para facilitar a escolha de métricas e a correta interpretação dos resultados. Neste contexto se destaca a abordagem Goal/Question/Metric [2] que se baseia na convicção 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. Dessa forma, a definição de métricas deve ser baseada nos objetivos definidos para o processo de avaliação. Estes objetivos, por sua vez, devem estar diretamente relacionados às metas organizacionais que levaram a implantação do programa de melhoria, pois este é um fator de grande influência para a obtenção do apoio da alta gerência.

Todas estas dificuldades fizeram com que a maioria das propostas de implantação de métricas para avaliação e melhoria de processos não tenham tido muito sucesso até os dias de hoje. Poucos foram os casos onde o seu uso não foi considerado ineficiente ou, até mesmo, desaconselhável. Fenton [5] 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. Na sua opinião, o futuro das métricas de software está no uso de métricas relativamente simples combinando diferentes aspectos do desenvolvimento de forma a permitir vários tipos de estimativas e avaliações. Assim, torna-se necessário analisar os fatores chave esquecidos pelas abordagens tradicionais como, por exemplo, a relação de causa e efeito e a combinação de evidências.

3. Avaliação de Processos de Software

Apesar de existir a possibilidade da implantação de métricas independentemente do uso de um processo [6], a abordagem proposta neste trabalho não tem por objetivo somente medir, mas sim levantar dados sobre as características do processo a fim de poder avaliá-lo. Dessa forma, nossa abordagem abrange (i) a definição das métricas a serem utilizadas, (ii) a definição de uma estrutura de decisão para avaliação das métricas, e, (iii) a definição de como realizar a medição do processo. A seguir descrevemos como atendemos cada um desses aspectos.

3.1 Seleção e Definição das Métricas para Avaliação do Processo

O primeiro passo para a implantação de um programa para a avaliação de processos de software baseado em medição é a seleção do conjunto de métricas que deverão ser utilizadas na análise do processo. Todas as métricas utilizadas neste trabalho foram extraídas ou adaptadas da literatura. É importante, entretanto, ressaltar que não pretendemos que este seja o melhor conjunto de métricas para a avaliação de um processo de software, mesmo porque, não foi este o objetivo do trabalho. Este é apenas um conjunto válido com o qual será possível apresentar as principais idéias da abordagem proposta.

Seguindo a abordagem GQM, foram inicialmente estabelecidos os objetivos a serem alcançados no programa de avaliação. Para estabelecer estes objetivos, especialistas nas áreas de gerência de projetos, qualidade de software, processo e melhoria de processo foram consultados a respeito dos principais problemas comumente enfrentados pelas empresas desenvolvedoras de software no contexto brasileiro. Nas entrevistas realizadas, vários foram os problemas citados, porém três se destacaram como sendo os mais relevantes para iniciar uma abordagem de melhoria de processo baseada em medições: a falta de precisão das estimativas de projeto, a baixa qualidade observada nos produtos liberados para uso, e, o alto custo envolvido no desenvolvimento de software.

Segundo os especialistas, a solução ou o melhor entendimento dos fatores que interferem nestes três itens aumentaria consideravelmente a qualidade e o desempenho das empresas que desenvolvem software. A partir da identificação destes problemas foram, então, definidos três objetivos para o programa de medição: (i) melhorar a precisão das estimativas de projeto; (ii) aumentar a qualidade dos produtos liberados para uso; (iii) diminuir o custo final dos projetos. Uma vez definidos os objetivos, estes devem ser utilizados para orientar a definição de questões a serem respondidas a fim de analisar se as metas estipuladas foram devidamente alcançadas. A seguir descrevemos as questões definidas para cada um dos objetivos

⇒ **Objetivo 1: Melhorar a precisão das estimativas de projeto**

Analisando esse objetivo definimos duas questões para melhor caracterizar o objeto em questão: (i) “qual a precisão da estimativa de cronograma?”, e (ii) “qual a precisão da estimativa de esforço?”. Com estas perguntas, é possível estudar os dois principais aspectos de uma estimativa de desenvolvimento e, desta forma, pode-se avaliar se o projeto atende à

primeira meta estipulada. A partir da definição destas questões, foram definidas métricas que comparam os dados estimados para o projeto com os dados reais medidos durante o decorrer dos trabalhos. Foram selecionadas duas métricas para cada uma das questões de forma a considerar tanto os valores globais (todo o projeto) quanto os valores detalhados por macro-atividade¹, como pode ser visto na Figura 1.

Objetivo 1 :	
• Propósito:	Melhorar
• Questão:	precisão
• Objeto:	estimativas de projeto
• Ponto de Vista:	analisado pelo ponto de vista dos gerentes de projeto
<hr/>	
Questão 1.1 :	
Qual a precisão das estimativas de cronograma do projeto?	
Métrica 1.1 a)	Precisão Total do Cronograma = $\frac{\text{Tempo real de todo o projeto.}}{\text{Tempo estimado do projeto}}$
Métrica 1.1 b)	Pr. Cron. por Macro-Atividade = $\frac{\text{Tempo real por macro-atividade}}{\text{Tempo estimado para a macro-ativ.}}$
Questão 1.2 :	
Qual a precisão das estimativas de esforço do projeto?	
Métrica 1.2 a)	Precisão Total do Esforço = $\frac{\text{Esforço real de todo o projeto}}{\text{Esforço estimado para o projeto}}$
Métrica 1.2 b)	Pr. Esforço por Macro-Atividade = $\frac{\text{Esforço real por macro-atividade}}{\text{Esforço estimado para a macro-ativ.}}$

Figura 1 – Definição das métricas para o primeiro objetivo de avaliação do processo segundo a abordagem GQM.

Para facilitar a coleta, o entendimento e a análise dos dados, cada métrica selecionada para um objetivo deve ser claramente definida. Baseados em [7], definimos que para medir “tempo” em um projeto seria utilizado o número de dias decorridos entre a data de início dos trabalhos (de uma atividade ou de todo o projeto) e a data na qual todas as atividades desta etapa são consideradas como encerradas. Ainda segundo o mesmo trabalho, definimos que a métrica de “esforço” seria medida em homens-hora sendo que um homem-hora é o tempo de uma hora empregado por um membro da equipe do projeto. Assim, “esforço” é o somatório do número total de horas empregadas por todos os membros da equipe na execução de uma tarefa (macro-atividade ou todo o projeto).

Definidos desta forma, medidas de “tempo” e “esforço” podem ser facilmente obtidas, sendo necessário para isso o armazenamento das datas, das atividades executadas por cada membro da equipe e do intervalo de tempo gasto para a sua execução. Para a coleta destes dados basta utilizar uma planilha eletrônica contendo estas informações.

⇒ **Objetivo 2: Aumentar a qualidade dos produtos liberados para uso**

De forma semelhante duas questões foram definidas para o objetivo 2: “qual a qualidade dos produtos antes da sua liberação para uso?” e “qual a qualidade dos produtos após a sua liberação para uso?”. Entretanto, medir a qualidade de um produto não é uma tarefa tão fácil e direta como no caso anterior devido ao grande número de componentes que podem interferir direta ou indiretamente na qualidade final do produto. Como um requisito primordial da nossa abordagem é utilizar métricas simples e fáceis de serem aplicadas, consideramos duas

¹ Definimos macro-atividade como uma atividade que é composta de várias sub-atividades (ou tarefas) elementares.

métricas básicas na avaliação de processos. Na Figura 2, podem ser vistas as questões definidas para a análise deste segundo objetivo com suas respectivas métricas.

Visando melhor estudar a qualidade dos produtos antes da sua liberação para uso, a densidade de defeitos no sistema está sendo utilizada como o fator de análise. Estabelecemos que “erro” é qualquer problema novo encontrado em um documento/produto no momento da sua aprovação. Para que seja possível coletar estes dados, todas as reuniões de revisão devem gerar um relatório contendo uma listagem dos erros encontrados. Quando um novo problema é encontrado em um produto de software que já tenha sido aprovado anteriormente, é necessário modificá-lo para que este atenda corretamente aos requisitos estipulados. Sempre que isto for necessário, deve-se documentar as modificações realizadas de forma a tornar possível sua posterior contagem de forma semelhante à feita com os erros.

Objetivo 2 :	
• Propósito:	Aumentar
• Questão:	qualidade
• Objeto:	produtos liberados para uso
• Ponto de Vista:	analisado pelo ponto de vista dos gerentes de projeto
<hr/>	
Questão 2.1 :	
Qual a qualidade dos produtos antes da sua liberação para uso?	
Métrica 2.1 a) Densidade de Defeitos = $\frac{\text{Número de erros e modificações}}{\text{Tamanho do sistema}}$	
<hr/>	
Questão 2.2 :	
Qual a qualidade dos produtos após sua liberação para uso?	
Métrica 2.2 a) Deterioração do Software = $\frac{\text{Esforço após a liberação para uso}}{\text{Esforço antes da liberação}}$	

Figura 2 – Definição das métricas para o segundo objetivo para avaliação do processo segundo a abordagem GQM.

Finalmente consideramos o tamanho do sistema como o número de linhas de código, excluindo linhas em branco e linhas contendo apenas comentários, como sugerido por Carleton [7]. Para garantir a consistência entre diferentes medições e evitar que divergências apareçam devido a formas distintas de contagem, deve-se construir uma ferramenta simples capaz de contar o número de linhas de código produzidas. É importante observar que para padronizar o número de defeitos encontrados durante os projetos, outras métricas de tamanho podem ser utilizadas, como é o caso, por exemplo, de pontos por função. Para isto, basta que a maturidade da empresa permita sua fácil implantação.

Além dos problemas internos que estão sendo utilizados para responder a primeira questão, defeitos não encontrados durante a produção do software, sendo apenas detectados pelo usuário final, reduzem sua satisfação com o produto. Determinar o que realmente representa qualidade do ponto de vista dos consumidores (usuários) é uma tarefa bastante complicada. Porém, é bastante claro que o número de problemas associados ao software varia inversamente à percepção de qualidade [8]. Seguindo a abordagem proposta por [9], a segunda questão levantada objetiva analisar a qualidade dos produtos liberados para uso através da comparação do esforço necessário para a realização de modificações para a correção de problemas encontrados após a liberação com o esforço total empregado antes da liberação. Com o uso desta métrica, vários fatores são implicitamente considerados na contagem do esforço empregado após a liberação para uso com a correção de defeitos. Somente a título de exemplo, podem ser citadas a quantidade de modificações realizadas e a complexidade de cada uma delas.

⇒ **Objetivo 3: Diminuir o custo final dos projetos**

Assim como no objetivo anterior, analisar genericamente os fatores que compõem o custo final de um projeto de software não é uma tarefa simples e, desta forma, foi decidido que somente o componente humano seria analisado visto que, na grande maioria dos casos, este é o fator determinante para o custo final dos projetos. Como pode ser visto na Figura 3, foram estabelecidas duas questões de forma a definir melhor o objeto de estudo. A primeira pergunta procura analisar o componente final do esforço empregado pela equipe, tanto para o projeto como um todo como para cada uma das macro-atividades que o compõem. Desta forma, pode-se visualizar qual a ordem de grandeza do esforço do projeto e analisar qual macro-atividade ou quais macro-atividades possuem o maior impacto no valor global. Estas são informações importantes que nos permitem direcionar os trabalhos de redução de custo e comparar projetos de mesma ordem de grandeza a fim de entender melhor os fatores que influenciam o valor final.

Objetivo 3 :	
• Propósito:	Diminuir
• Questão:	custo final
• Objeto:	dos projetos
• Ponto de Vista:	analisado pelo ponto de vista dos gerentes de projeto
<hr/>	
Questão 3.1 :	
Qual é o esforço total do projeto?	
Métrica 3.1 a) Esforço total do projeto.	
Métrica 3.1 b) Esforço por macro-atividades do projeto.	
<hr/>	
Questão 3.2 :	
Qual o percentual de re-trabalho em relação ao esforço do projeto?	
Métrica 3.2 a) Percentual de Re-trabalho Total = $\frac{\text{Esforço em re-trabalho} * 100}{\text{Esforço total do projeto}}$	
Métrica 3.2 b) Pe. Re-trab. Macro-Ativ. = $\frac{\text{Esforço em re-trabalho na macro-ativ} * 100}{\text{Esforço total na macro-atividade}}$	

Figura 3 – Definição das métricas para o terceiro objetivo para avaliação do processo segundo a abordagem GQM.

Além disso, um fator chave que contribui para o aumento desnecessário do valor final é o esforço empregado com análise, correção, testes e implantação ocorridos devido a problemas detectados durante e após o desenvolvimento. Definimos, re-trabalho como qualquer atividade de correção, modificação ou melhoria realizada sobre um produto/sub-produto de software que já tenha sido finalizado. Assim, não é considerado re-trabalho, por exemplo, os problemas encontrados por um programador durante o desenvolvimento de um módulo ou função, pois este ainda não foi validado e considerado finalizado.

Dessa forma, o “esforço em re-trabalho” empregado durante o projeto está sendo comparado com o valor do “esforço total”, tanto para o valor global como para os valores detalhados por macro-atividade. Com este tipo de informação, pode-se direcionar os trabalhos de redução de custo às macro-atividades que possuem um nível de re-trabalho considerado acima do normal.

3.2 Estrutura para Avaliação de Métricas

Pode-se observar, através da análise dos objetivos definidos, que nem todas as informações necessárias para a melhoria do processo podem ser obtidas aplicando somente estas medidas. Elas ajudam a determinar se as metas estabelecidas foram realmente alcançadas, porém, quando isto não acontece, pouca informação é fornecida a respeito da origem dos desvios. Segundo Fenton [5], este problema ocorre devido aos modelos atuais não conseguirem fornecer todas as informações necessárias para a correta análise e tomada de decisão. O uso de métricas tem sido dominado por modelos estatísticos, enquanto o que é realmente necessário é um modelo causal, pois este provê uma estrutura lógica para explicar eventos que, então, podem ser quantificados.

Seguindo esta abordagem, definimos uma estrutura capaz de inferir, a partir dos resultados obtidos nas medições, quais os possíveis problemas que a equipe de desenvolvimento pode ter enfrentado quando os objetivos definidos não são alcançados. Sempre que um valor não aceitável para uma métrica for obtido, alguma inferência pode ser feita a respeito das características do processo utilizado. Para tornar possível a construção desta estrutura, buscamos, novamente, a ajuda de especialistas e foi estabelecido o seguinte procedimento:

I - Com um objetivo selecionado, deve-se questionar os especialistas a respeito de quais os possíveis problemas diretos que a equipe de desenvolvimento pode ter enfrentado para não ser possível alcançar este objetivo. Neste passo o objetivo deve ser dividido nos sub-objetivos diretos que o compõem. Quando não for possível ou interessante detalhá-lo melhor, marque este como um nó terminal da estrutura e passe para o passo seguinte;

II - A partir disto, os especialistas devem informar o que pode ser considerado um problema para este objetivo levando em consideração o uso ou a definição de uma métrica para sua análise e/ou o uso dos sub-objetivos que o compõem. Ou seja, deve ser definido como decidir se o objetivo selecionado foi ou não alcançado. Deve-se definir uma única métrica que possa descrever o objetivo selecionado de forma mais precisa e, assim, permitir a sua análise. Isto nem sempre será possível, pois alguns objetivos são, na verdade, um conjunto de fatores e somente quando detalhados poderão ser analisados por métricas específicas;

III - Enquanto existir sub-objetivo não analisado, reiniciar o processo tomando como objetivo um destes sub-objetivos.

Este procedimento procura definir métricas capazes de analisar cada um dos sub-objetivos que compõem o objetivo principal de forma que sejam encadeados de maneira hierárquica na qual os níveis inferiores definem causas para problemas encontrados nos níveis superiores. Uma decisão difícil neste procedimento está em definir quais são causas diretas e quais são causas indiretas para o correto encadeamento dos nós na estrutura. Seguindo este procedimento, será possível decidir quais as possíveis causas para o desvio quando uma meta não for alcançada. É importante observar que a construção desta estrutura pode ser tão mais detalhada quanto mais maduros forem as métricas e o processo utilizados.

Esta estrutura pode ser vista como um framework capaz de integrar diversas métricas com características distintas (como, por exemplo, diferentes tipos de escala). Além disso, informações a respeito do contexto do projeto ou, até mesmo, das características do ambiente de desenvolvimento podem ser introduzidas e relacionadas como qualquer outra métrica. Isto é muito interessante, pois este tipo de informação é de suma importância para a correta

interpretação dos resultados das medições. Todos estes dados, vistos em conjunto, formam o contexto geral do projeto e permitem um melhor apoio à tomada de decisão.

⇒ **Estrutura de Avaliação para o Primeiro Objetivo**

A fim de exemplificar a construção desta estrutura causal, foi escolhido o primeiro objetivo proposto neste trabalho (melhorar a precisão das estimativas) para validar esta idéia. A partir de então, foi seguido o procedimento anteriormente definido como será descrito abaixo.

O objetivo principal, que deve ocupar o topo da estrutura, é a “Precisão das Estimativas” e, de acordo com o passo I, este deve ser dividido em dois sub-objetivos diretos que são: “Precisão Total do Cronograma” e “Precisão Total de Esforço”. Como não existe uma métrica única que possa descrever seu comportamento, deve ser considerado que o objetivo principal não foi alcançado quando pelo menos um dos sub-objetivos não for atendido (passo II). Abaixo pode ser vista a estrutura definida até o momento.

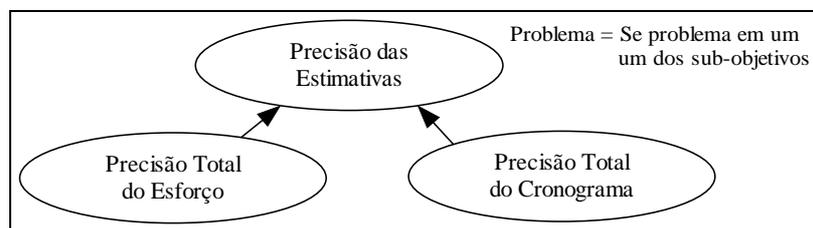


Figura 4 – Estrutura de decisão para o objetivo 1 após uma execução dos passos I e II do procedimento.

Seguindo o procedimento definido (passo III), foi selecionado para análise o objetivo “Precisão Total do Cronograma” o qual pode ser detalhado na “Precisão do Cronograma por Macro-atividade” (passo I). Como pôde ser visto na Figura 1, a métrica definida para este objetivo é “o tempo total sobre o tempo estimado para todo o projeto” e para definir o que é um problema para ela é necessário levar em consideração um número muito grande de fatores, como, por exemplo, as características da empresa e o nível de maturidade do processo. Desta forma, chegou-se a conclusão de que o valor limite para o atraso do cronograma deveria ser de alguma forma definido pelo próprio gerente do projeto. Esta é uma solução interessante, pois permite a utilização de valores mais próximos da realidade do projeto que esta sendo medido e tolerâncias cada vez menores podem ser utilizadas permitindo uma melhoria contínua do processo. Uma análise semelhante a esta foi feita para o outro sub-objetivo chegando a um grafo da seguinte forma:

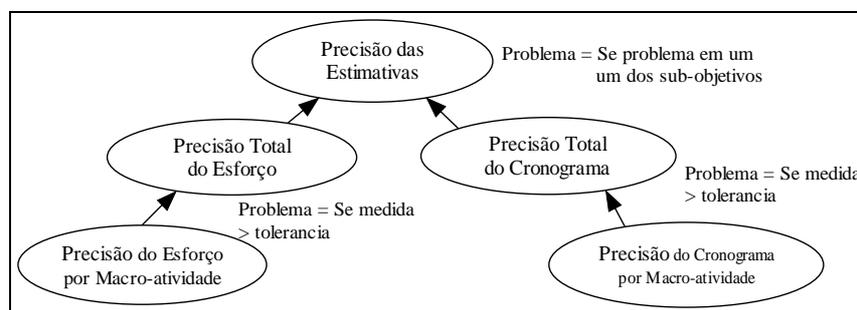


Figura 5 – Estrutura de decisão para o objetivo 1 parcialmente desenvolvida.

Para a “Precisão do Cronograma por Macro-atividade” duas podem ter sido as causas de um atraso no cronograma: aumento de esforço ou emprego incorreto do esforço. O aumento de esforço já está sendo representado pela “Precisão do Esforço por Macro-atividade”

devendo somente ser ligado como uma possível causa para o problema da precisão no cronograma. Por outro lado, o “Emprego do Esforço na Macro-atividade” deverá ser incluído à estrutura como a segunda causa possível. Para determinar se houve um problema na “Precisão do Cronograma por Macro-atividade”, também será utilizada a tolerância já definida para a “Precisão Total do Cronograma”.

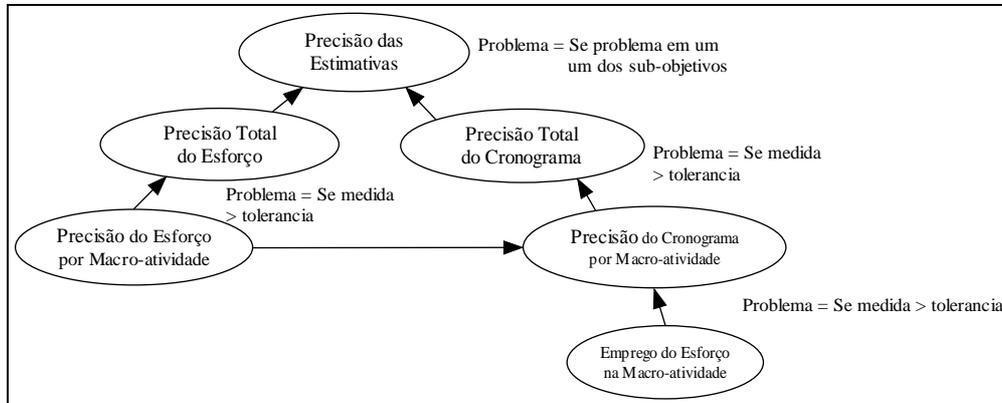


Figura 6 – Estrutura de decisão para o objetivo 1 parcialmente desenvolvida.

Ao repetir esta análise para o “Emprego do Esforço na Macro-atividade” observamos que não é possível, para a grande maioria das empresas de software, detalhar melhor as possíveis causas para este problema. Desta forma, este não será detalhado em sub-objetivos sendo um nó terminal da estrutura. Pode-se considerar que ocorreu um problema no “Emprego do Esforço na Macro-atividade” quando não houver um problema de “Precisão do Esforço na Macro-atividade”, ou seja, se ocorreu um problema no tempo, mas o esforço foi dentro do previsto, então, conclui-se que o problema foi o incorreto emprego do esforço. Somente para facilitar a visualização, todos os nós terminais da estrutura serão representados por retângulos ao invés de elipses como pode ser visto na figura 7.

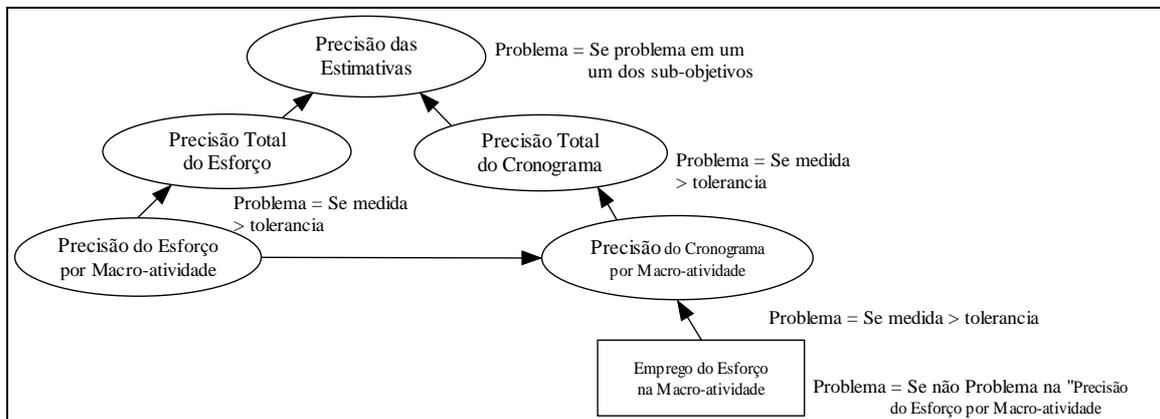


Figura 7 – Estrutura de decisão para o objetivo 1 com a representação do primeiro nó terminal.

Este procedimento foi repetido para a “Precisão do Esforço por Macro-Atividade” e seus sub-objetivos até que não existissem mais nós passíveis de serem expandidos, ou que não fosse interessante sua expansão no nível em que se encontram os trabalhos. A estrutura final pode ser vista na figura 8.

3.3 Medição do Processo de Software

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, além disso, deve ser devidamente executado pela equipe de desenvolvimento para que os valores coletados sejam válidos para a sua avaliação.

Na nossa abordagem a tarefa de definição do processo de software segue a proposta de [10] e 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). A definição deste processo (chamado de processo padrão) e dos processos especializados permite, também, sua instanciação para projetos específicos, considerando-se as características particulares de cada projeto de desenvolvimento.

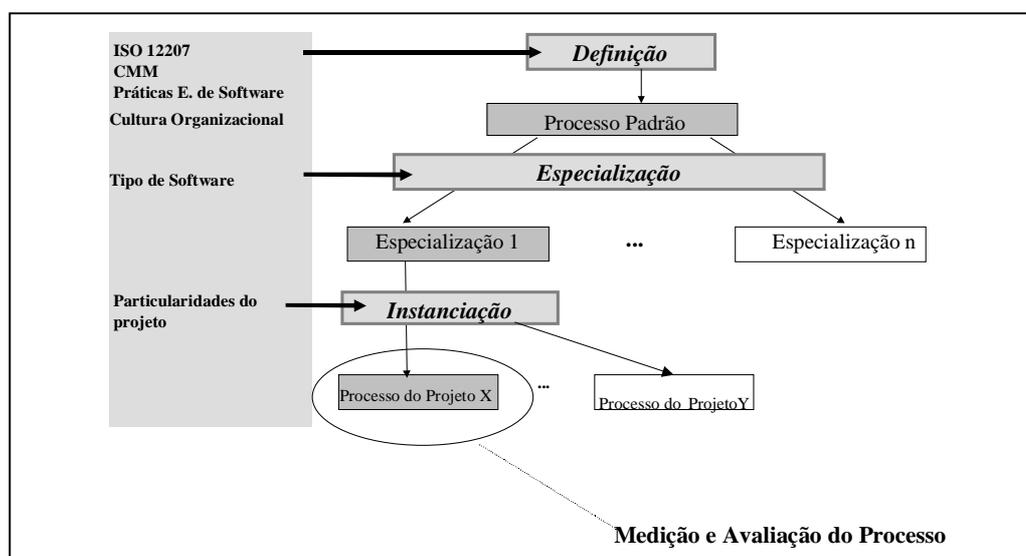


Figura 9 – Modelo para Definição de Processos de Software

Alguns dos documentos já previstos de serem elaborados durante o desenvolvimento do projeto foram adaptados e outros novos foram inseridos de forma que todos os dados necessários para a análise do processo fossem facilmente obtidos. 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. Os seguintes documentos foram, portanto, inseridos ou adaptados como descrito abaixo:

- **Planilha de Atividades:** todos os membros da equipe devem informar em uma planilha diária quais foram as atividades do processo executadas, o tempo gasto em cada uma delas e se trata-se ou não de re-trabalho.
- **Aprovação de uma Atividade:** para declarar que uma macro-atividade terminou é necessário um documento contendo a data da sua aprovação e a assinatura do responsável por sua aprovação.

Com estas informações pode ser criada uma forma automática de levantar o tempo e o esforço empregado em cada macro-atividade do processo e o esforço empregado em re-trabalho por macro-atividade.

- **Relatório de Erros:** todas as atividades de avaliação produzem um relatório dos erros encontrados durante a avaliação.

- **Aprovação de um Produto de Software:** para declarar que um produto de software foi totalmente concluído é necessário um documento contendo a data da sua aprovação e a assinatura do cliente.
- **Relatório de Modificações:** sempre que for necessário modificar algum documento já aprovado, um relatório contendo uma breve descrição das modificações deve ser gerado.
- **Questionário sobre a Experiência da Equipe:** todos os membros da equipe precisam responder a um questionário simples contendo perguntas sobre sua experiência em projetos de porte similar e no domínio do projeto, no processo, nas ferramentas, na linguagem de programação e no método utilizados. Para diminuir a subjetividade desta métrica, foi estabelecida a seguinte ordenação: 0 – nenhuma experiência; 1 – treinamento acadêmico; 2 – prática em até três projetos; 3 – experiente; 4 – capaz de orientar outros.
- **Relatório de Rotatividade:** documentação de todas as mudanças na equipe ocorridas durante o projeto.
- **Questionário sobre o Contexto do Projeto:** informações sobre o contexto do projeto respondidas pelo gerente do projeto antes de iniciar a avaliação do processo.

Com o correto preenchimento de todos estes documentos durante os trabalhos, os dados necessários para a avaliação do processo segundo o primeiro objetivo definido poderão ser, facilmente, obtidos por ferramentas automáticas e passados para o sistema que fará a avaliação propriamente dita, conforme descrito na seção seguinte.

4. Apoio Automatizado para a Avaliação de Processo

Para que um programa de medição seja efetivo e eficiente é necessário prover os desenvolvedores com o adequado ferramental tecnológico e metodológico. Com o intuito de tentar garantir que todos os passos definidos na abordagem fossem devidamente executados, foi construído um sistema capaz de apoiar a equipe de desenvolvimento nos procedimentos de coleta e análise dos dados obtidos nas medições. Este sistema foi desenvolvido a partir da infra-estrutura já definida na estação Taba, projeto no qual este trabalho está inserido.

A Estação Taba é um meta-ambiente capaz de gerar ambientes de desenvolvimento de software específicos para um determinado projeto, definidos a partir de uma descrição do processo de desenvolvimento a ser utilizado. Nestes ambientes instanciados foram incorporadas pequenas funcionalidades para orientar a execução dos procedimentos de coleta de dados para cada uma das métricas definidas na abordagem. Todas as modificações nos documentos do processo, necessárias para permitir que a equipe de desenvolvimento pudesse armazenar os dados das métricas durante a execução do projeto, foram incorporados aos ambientes instanciados. Apenas para exemplificar este procedimento, apresentamos na figura 10 a tela da Planilha de Atividades. Esta planilha deve ser preenchida por cada membro da equipe de desenvolvimento para que os dados referentes às atividades executadas possam ser corretamente armazenados. Com estas informações, são calculadas as métricas de esforço, tempo e re-trabalho em todo o projeto e em cada uma de suas macro-atividades.

Foi construído um sistema baseado em conhecimento para a implementação da estrutura de decisão apresentada anteriormente. Após o usuário do sistema determinar qual dos três objetivos propostos nesta abordagem será utilizado para a avaliação do processo (atualmente somente o primeiro objetivo foi implementado), 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 analisa os dados e informa as causas mais prováveis para os problemas ocorridos no alcance deste objetivo. É importante observar que,

somente após o término de todas as atividades do projeto, será possível a análise do processo utilizado, pois somente neste momento todos os dados referentes às métricas estarão coletados e disponíveis para o sistema baseado em conhecimento.

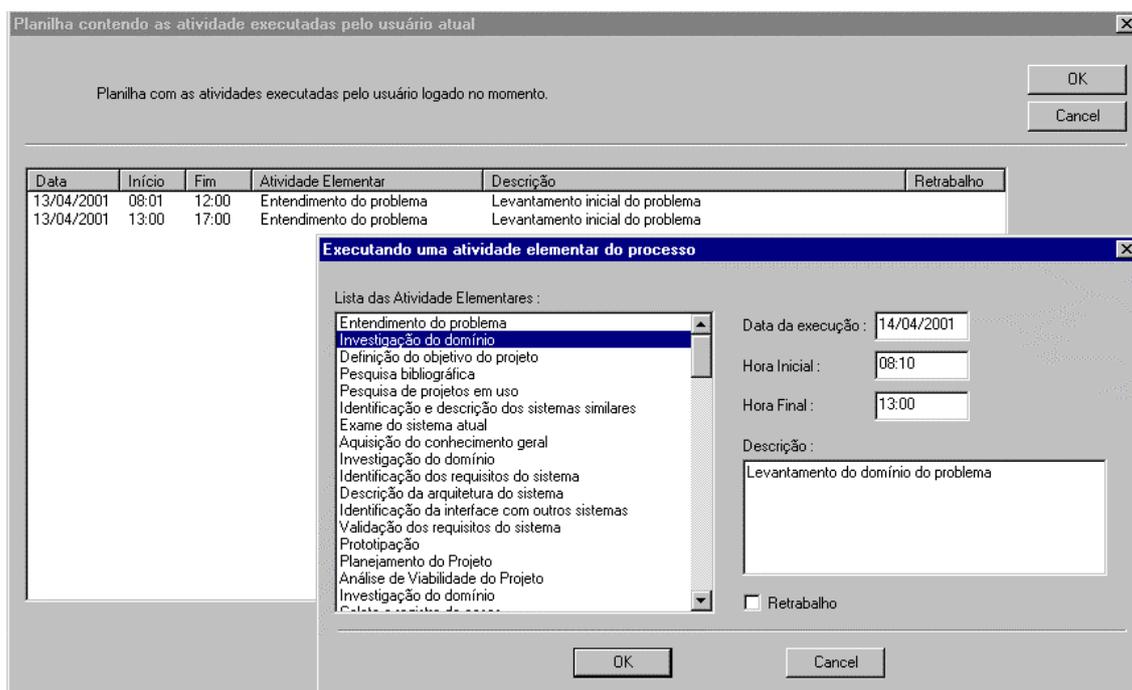


Figura 10 – Planilha de Atividades parcialmente preenchida.

Como pode ser visto no exemplo abaixo, a resposta do sistema é dada em uma forma de árvore de diretórios onde os nós mais internos procuram definir causas para os problemas encontrados nos nós mais externos. No exemplo abaixo, gerado para a análise do primeiro objetivo, o problema encontrado na “Precisão das Estimativas de Projeto” pode ser explicado pelo problema encontrado 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.

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. Desta forma, a organização tem uma ferramenta capaz de orientar onde deve concentrar seus esforços para corrigir problemas e melhorar o processo de software.

5. Experiência de Uso

Com o objetivo de verificar a viabilidade desta abordagem em um sistema real, acompanhamos e coletamos os dados referentes às métricas durante o desenvolvimento de um projeto em uma empresa para a qual já havíamos definido o processo de software.

Os problemas sugeridos pelo sistema como os mais importantes de serem analisados, para melhoria do processo, coincidiram com a nossa observação sobre o processo e o desenvolvimento do projeto. As observações do responsável pelo projeto na empresa destacaram, principalmente, o valor de se estar usando e medindo o processo durante os

trabalhos por tornar possível determinar com maior objetividade onde estão os problemas do processo utilizado e seus impactos no projeto.

Outro aspecto destacado foi que o próprio processo de disciplinar as pessoas em coletar os dados referentes ao uso do processo e ao seu próprio desempenho, aumentou o conhecimento da equipe de desenvolvimento a respeito de suas capacidades e limitações fazendo com que o processo de coleta de métricas se torne por si só a primeira forma de melhoria do processo de software da empresa.

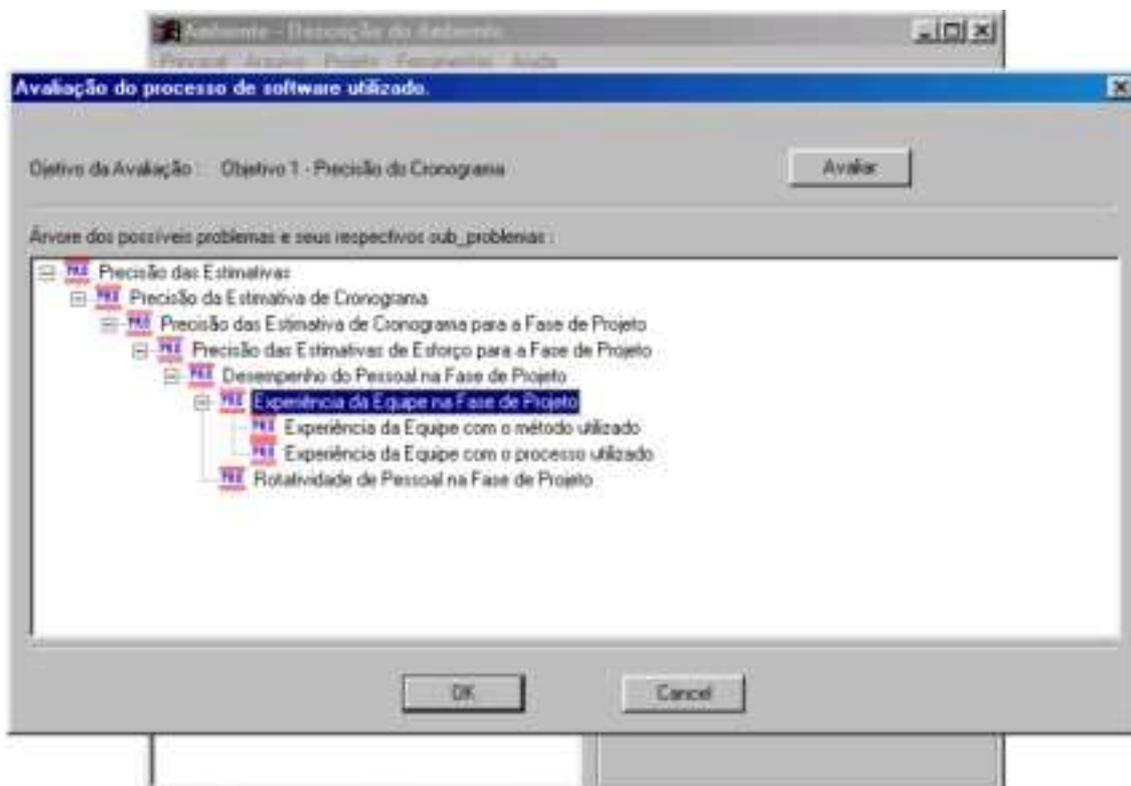


Figura 11: Resposta final do sistema baseado em conhecimento gerada para a avaliação do processo segundo o primeiro objetivo definido nesta abordagem.

6. Conclusão

É possível começar a entender o mundo real utilizando métricas pouco sofisticadas, mas à medida que a maturidade e o entendimento aumentam, formas mais detalhadas de medir as características das entidades do mundo real se tornam possíveis e, até mesmo, necessárias. O uso de métricas, atualmente, provê pouca informação útil para o alcance destes objetivos. De forma isolada, cada métrica descreve uma característica particular que pode ser analisada e comparada com uma meta ou norma. Examinadas individualmente no decorrer do tempo, podem ser úteis para encontrar tendências e prever comportamentos futuros. Porém, vistas em conjunto, as métricas podem prover uma representação mais precisa do mundo real onde o contexto geral do projeto pode ser visualizado. Esta pode ser uma poderosa ferramenta de análise ainda pouco explorada atualmente. Um pequeno, mas significativo, conjunto de métricas analisado como um todo pode trazer benefícios que, da forma tradicional, dificilmente poderiam ser alcançados.

Neste artigo foi apresentada uma abordagem para o uso sistemático de métricas com o objetivo de auxiliar e orientar a avaliação de processos de software. Nesta, foram definidos procedimentos para a seleção e definição das métricas utilizadas na avaliação, foram determinados os procedimentos que devem ser seguidos para o correto levantamento dos dados durante a execução do projeto e foi proposta uma estrutura capaz de analisar os resultados obtidos nas medições de forma conjunta a fim de encontrar as causas mais prováveis quando o objetivo definido para o programa de medição não é alcançado.

Podem ser encontrados na literatura vários trabalhos que se baseiam no uso de métricas para promover a melhoria do processo de desenvolvimento de software. Porém, é importante observar que dentre as abordagens encontradas, nenhuma propõem o uso do conhecimento de especialistas embutidos em uma ferramenta visando automatizar o procedimento decisório de melhoria do processo de software como é o caso da abordagem proposta neste trabalho.

Referências Bibliográficas

- [1] Oman P., Pfleger S.L., *Applying Software Metrics*, Los Alamitos, CA, IEEE Computer Society.
- [2] Basili V.R., Caldiera G., Rombach H.D., *Goal Question Metric Paradigm*, Encyclopedia of Software Engineering, 2 Volume Set, John Wiley & Sons, Inc, 1994.
- [3] Fenton N.E., Pfleeger S.L, *Software Metrics – A Rigorous & Practical Approach*, Second Edition, Boston, MA, PWS Publishing Company, 1997
- [4] 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.
- [5] Fenton N.E., Neil, M., *Software Metrics: Roadmap*, Computer Science Department, Queen Mary and Westfield College, London, UK, 2000.
- [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 1997].
- [7] Carleton A.D., Park R.E, Goethert W.B., Florac W.A., Bailey E.K., Pfleeger 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.
- [8] 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.
- [9]Tajima D., Matsubara T., *The Computer Software Industry in Japan*, IEEE Computer, 14(5), 1981 in [2].
- [10] Machado,L.F.C. et al DEF-PRO: Uma Ferramenta para Apoiar a Definição do Processo Padrão, XI CITS, Curitiba, Junho 2000

- [11] Conte, S.D., Dunsmore, H.D., Shen, V.Y., *Software Engineering Metrics and Models*, Benjamin-Cummings, Menlo Park, CA, 1986 in [FENTON 1997]
- [12] Eman K.E., Madhavji N.H., *Elements of Software Process Assessment and Improvement*, IEEE Computer Society, 1999.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.