

# Controle de métricas no processo de desenvolvimento de software através de uma ferramenta de *workflow*

Gustavo Zanini Kantorski, Marcelo Lopes Kroth

Centro de Processamento de Dados  
Universidade Federal de Santa Maria  
Santa Maria, RS, Brasil

{gustavoz, tuco}@cpd.ufsm.br

**Abstract.** *The quality in the software development is an essential characteristic so that the software product attends to its users' expectations. This paper introduces a tool based on knowledge, to control the software development process since its conception until the distribution for the final user, allowing the inference of some metric through the users' interaction with the tool.*

**Key words:** *software process, software engineering, metrics, workflow*

**Resumo.** *A qualidade no desenvolvimento de software é uma característica essencial para que o produto de software atenda às expectativas de seus usuários. Este artigo apresenta uma ferramenta baseada em conhecimento, para controlar o processo de desenvolvimento de software desde a sua concepção até a distribuição para o usuário final, permitindo a inferência de algumas métricas através da interação dos usuários com a ferramenta.*

**Palavras-chave:** *processos de software, engenharia de software, métricas, workflow,*

## 1 Introdução

No contexto de desenvolvimento de software, qualidade pode ser entendida como um conjunto de características a serem satisfeitas em determinado grau, de maneira que o produto de software atenda às necessidades explícitas e implícitas de seus usuários [Rocha 1994]. Todavia, não se obtém qualidade do produto de forma espontânea. A qualidade tem que ser construída. Desta forma, a qualidade do produto depende fortemente da qualidade de seu processo de desenvolvimento [Nita 2002].

Para avaliar a qualidade é necessário haver meios de medi-la. Isto é, é preciso obter uma medida que quantifique o grau de alcance de uma característica de qualidade. Assim, para computar uma característica de qualidade é necessário estabelecer uma métrica capaz de quantificá-la e fazer uma medição para determinar a medida, resultado da aplicação da métrica [Duarte 2000].

Este artigo tem como objetivo apresentar uma ferramenta de *workflow* que possibilite o controle da qualidade do processo de desenvolvimento de software e, ao mesmo tempo, identifique características de qualidade através de um conjunto de métricas utilizadas desde o levantamento de requisitos até a distribuição do produto de software, indicando possíveis problemas enfrentados pela equipe de desenvolvimento durante o decorrer dos trabalhos e que podem ter afetado o desempenho do processo. 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. Melhorias essas, que podem ser configuradas na ferramenta de *workflow* proposta, agilizando o processo de desenvolvimento.

Além da introdução este artigo apresenta mais 4 seções. Na seção 2 são apresentadas as métricas a serem utilizadas na medição dos processos de software. A seção 3 descreve a

ferramenta proposta para configuração e utilização dos *workflows*. Na seção 4 é apresentada a derivação das métricas a partir dos *workflows* definidos na ferramenta. Finalmente, na seção 5 são apresentadas as conclusões e trabalhos futuros.

## 2 Medindo e Melhorando o Processo de Software

Organizações que possuem um programa de métricas implantado com sucesso apresentam os seguintes benefícios [Baumert 1992]:

- Gerentes mais engajados no projeto.
- Maior capacidade para quantificar decisões a serem tomadas.
- Melhor planejamento, controle e monitoramento de projetos.
- Melhor entendimento do processo e ambiente de desenvolvimento
- Identificação de áreas de potencial melhoria no processo, bem como uma métrica objetiva do esforço necessário para melhoria.
- Comunicação entre membros da organização otimizada.

Para a realização do presente trabalho, foi estabelecido um conjunto de métricas baseado em [Gomes 2000]. O conjunto selecionado está longe de ser o melhor conjunto de métricas, mas é um conjunto válido para implementar essa idéia e, a partir de resultados obtidos, poder melhorar.

O conjunto de métricas foi organizado da seguinte forma [Gomes 2000]:

- Tempo: tempo total de projeto; tempo nas fases de análise, projeto, codificação, teste de unidades feito por analistas, testes do sistema e testes de homologação; tempo em reuniões de revisão; tempo em retrabalho.
- Precisão da estimativa de cronograma: precisão da estimativa de cronograma de todo o projeto, precisão das estimativas de cronograma para as fases de análise, projeto, codificação, testes de unidade feitos por analistas, testes do sistema e testes para homologação.
- Esforço: esforço total de projeto; esforço nas fases de análise, projeto, codificação, testes de unidade feitos por analistas, testes do sistema e testes de homologação; esforço em reuniões de revisão; esforço em re-revisões (novas reuniões de revisão realizadas devido a não aprovação do produto em reuniões anteriores); esforço em retrabalho.
- Precisão da estimativa de esforço: precisão da estimativa de esforço para todo o projeto; precisão da estimativa de esforço para as fases de análise, projeto, codificação, teste de unidades feitos por analistas, teste do sistema e teste para homologação.
- Tamanho do sistema: número de linhas de código.
- Número de erros: número de erros na especificação de requisitos e no projeto do sistema encontrados em reuniões de revisão; erros no código encontrados no teste de unidade feitos por analistas;
- Número de modificações: número de modificações na especificação de requisitos, projeto ou código após a sua aprovação.
- Densidade de defeitos: número de erros somado ao número de modificações em relação ao tamanho do sistema.
- Rotatividade do pessoal: percentual de pessoas que saíram, entraram ou mudaram de função durante o desenvolvimento do projeto.
- Produtividade: número de linhas de código produzidas por unidade de esforço.

- Deterioração do software: relação entre o esforço gasto para corrigir problemas encontrados após a liberação do sistema para o usuário comparado ao esforço gasto antes da liberação do software para o usuário.
- Experiência da equipe: experiência na linguagem de programação, no domínio da aplicação, nas ferramentas, no método e no processo de desenvolvimento, tipo de treinamento em engenharia de software, tempo total de experiência profissional.

### **3 O ambiente de *workflow* para controle de métricas**

Para alcançar níveis mais altos de qualidade torna-se necessário melhorar cada etapa do ciclo de vida do software [Oman] e, para tornar isto possível, dados quantitativos que descrevem 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 [Gomes 2001].

Para que o processo de medição seja efetivo e eficiente é necessário prover os desenvolvedores com uma ferramenta adequada. Com o objetivo de tentar garantir que todos os passos definidos para um projeto fossem devidamente executados, foi construída uma ferramenta capaz de controlar o processo de desenvolvimento, desde os requisitos até a distribuição, garantindo a medição das métricas, de forma configurável e dinâmica.

A ferramenta proposta permite a criação de *workflows* para gerenciar o processo de desenvolvimento e a indicação de variáveis que possibilitam a inferência sobre as métricas apresentadas na seção 2.

A ferramenta é dividida em duas fases: a fase de configuração e criação dos *workflows* e a fase de utilização pelos usuários desenvolvedores.

#### **3.1 Fase de Configuração**

Na fase de configuração são montados os fluxos de trabalho a serem utilizados no projeto, definindo-se responsáveis, os artefatos de entrada e os artefatos de saída, tempo de duração de atividades e a semântica do próprio fluxo.

A criação dos *workflows* é dinâmica e configurada por projeto. Isto é, cada projeto pode ter o seu fluxo de requisitos, análise e projeto, implementação, teste e evolução. Existe, também, a comunicação entre *workflows* de etapas diferentes, permitindo a medição de métricas importantes no processo. Deve-se salientar que os valores medidos para uma métrica estão no próprio *workflow* e não serão informados pelos membros da equipe de desenvolvimento durante a utilização da ferramenta. Na configuração de cada *workflow*, o usuário informa as variáveis pertinentes para a medição da métrica, liberando do usuário desenvolvedor o preenchimento de valores, como por exemplo, tempo gasto em determinada atividade, se a atividade realizada é de re-trabalho ou não.

Para exemplificar a ferramenta será mostrada a configuração do *workflow* de implementação. É na configuração do *workflow* que é definido o processo de trabalho existente entre as equipes envolvidas no projeto. As métricas serão derivadas a partir desta configuração.

Para criar um *workflow* de implementação é necessário dispor dos documentos padrões a serem definidos e encaminhados em cada passo do *workflow*. Não está no escopo deste artigo discutir a padronização de documentos.

A figura 1 mostra a interface necessária para a criação de um dos fluxos do *workflow* de implementação. Para criar os fluxos deve-se informar a situação atual e situação final da atividade, qual a aplicação necessária para realização dessa atividade, o tempo estimado para realização da atividade, se alguma ação será executada pela atividade de forma a modificar o comportamento do fluxo, os responsáveis por essa etapa do fluxo e quem receberá o resultado desta fase.

Descrição do tipo de documento	Situação atual	Situação futura	Nº dias
Implementação	30	50	1

Restrição de abertura na Caixa	Aplicação de abertura
Grupos de Usuários	02.07 Solicitação de Serviço

Restrição de Destino	Aplicação de ação
Programadores (Não contém)	

Retorno:  Sim  Não

Consulta:  Sim  Não

Destino: \_\_\_\_\_

**Figura 1. Configuração de um fluxo do *workflow*.**

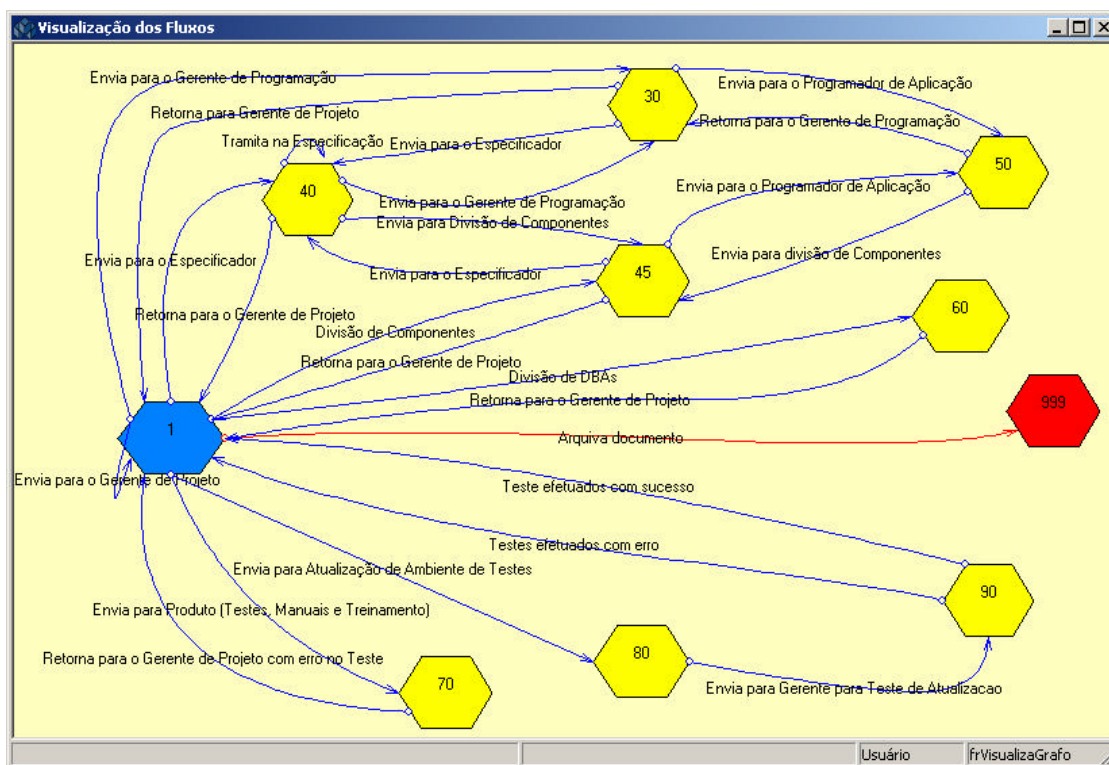
Na guia “Documento Padrão para o Fluxo”, da figura 1, está vinculada a formatação dos documentos padrões que devem ser preenchidos em cada etapa do processo de desenvolvimento, durante a utilização da ferramenta. A compilação dos documentos de todas as etapas do *workflow* gerará a documentação de cada etapa do processo de desenvolvimento.

Uma característica importante da ferramenta é a possibilidade de ajustar a cada etapa do *workflow* qual a aplicação que será disponibilizada para o usuário. Esta informação é relevante visto que as informações necessárias para registrar um requisito são diferentes das informações que são repassadas a um programador. Ou seja, os artefatos em cada etapa do *workflow*, sejam eles, de entrada ou saída, são diferentes. Além disso, o programador tem acesso para consulta, aos artefatos gerados nas etapas anteriores.

Após a configuração de todos os fluxos do *workflow*, a ferramenta permite a visualização global do mesmo, permitindo a correção de eventuais falhas na criação do processo. A figura 2 mostra todos os fluxos existentes para o *workflow* de implementação de um projeto

específico. Além da implementação estão configuradas as demais etapas, como requisitos, análise e projeto, teste e evolução.

Na figura 2 os hexágonos representam as situações (eventos) de um fluxo. As linhas contínuas representam as atividades que serão realizadas quando um documento passar pelo fluxo. Linhas contínuas em azul significam que somente o trâmite será realizado pelo documento. Linhas contínuas em vermelho representam que alguma ação será executada ao enviar o documento no fluxo, modificando alguma semântica no processo. O *workflow* inicia no hexágono azul, nomeado com o valor 1, sendo considerado o evento inicial e encerra seu processo no hexágono vermelho, nomeado com o valor 999, passando a agir em outro *workflow* do processo de desenvolvimento.



**Figura 2. Workflow de Implementação.**

A comunicação entre *workflows* é importante, para determinar, dentro do processo de desenvolvimento, como está o andamento de determinado projeto. Através da ferramenta é possível gerenciar a interação entre *workflows* diferentes através de uma funcionalidade chamada relacionamento de documentos, onde um documento ao encerrar um *workflow* é, automaticamente, criado e associado ao próximo *workflow* do processo de desenvolvimento. Desta forma, na configuração dos *workflows* é necessário informar as possíveis associações existentes entre os *workflows*. No exemplo ilustrado, ao final do *workflow* de implementação, o próximo passo pode ser, enviar para testes ou retornar para análise e projeto para esclarecimentos.

### 3.2 Fase de Utilização

Na fase de utilização, os usuários desenvolvedores interagem com a ferramenta conforme a customização realizada na fase de configuração. Essa interação é feita através de um repositório de documentos, chamado de caixa postal, conforme apresentado na figura 3, onde

todos os documentos que estão no *workflow* podem ser visualizados, desde que o usuário esteja devidamente autorizado.

Na caixa postal podem ser visualizados os documentos enviados e recebidos e, após a realização da tarefa, pode ser enviado para o próximo passo do *workflow*. O usuário somente poderá enviar para os passos configurados na fase anterior, não podendo alterar a configuração do *workflow* criado. Ao enviar, ele informa o responsável pela próxima etapa e encaminha os artefatos gerados durante a permanência em sua caixa postal para o próximo passo do fluxo. Durante a permanência do documento na caixa postal do usuário desenvolvedor, os vários artefatos gerados em etapas anteriores estão disponíveis para consulta.

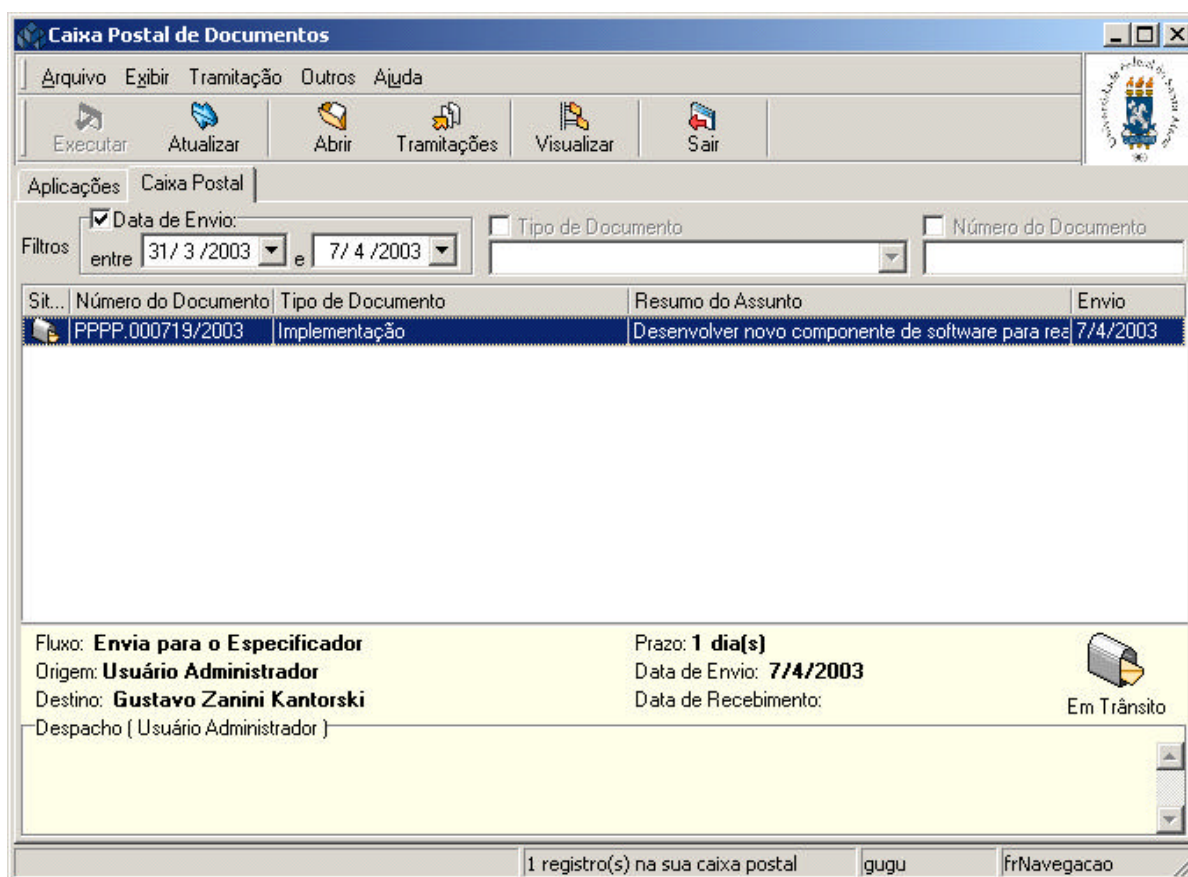


Figura 3. A Caixa Postal de Documentos..

As medidas para obtenção das métricas são baseadas na comparação das variáveis definidas na fase de configuração com a fase de utilização e na sequência dos documentos presentes em todos os *workflows* do processo de desenvolvimento.

#### 4 Derivando as métricas a partir da ferramenta de *workflow*

Definidos os *workflows* do processo de desenvolvimento, é possível aplicar as regras de derivação para permitir as medições no processo.

Aplicando as métricas definidas em [Gomes 2000] com os *workflows* propostos, tem-se o mapeamento apresentado na tabela 1.

**Tabela 1.Mapeamento de Métricas.**

Métrica	Mapeamento
Tempo total do projeto	Somatório do tempo real utilizado em todos os <i>workflows</i> desde requisitos até testes.
Tempo na fase de análise e projeto	Tempo total utilizado no <i>workflow</i> de análise e projeto.
Tempo na fase de codificação	Tempo total utilizado no <i>workflow</i> de implementação.
Tempo na fase de teste	Tempo total utilizado no <i>workflow</i> de teste.
Tempo em reuniões de revisão	Tempo dos documentos que foram enviados para o <i>workflow</i> de implementação e retornaram ao <i>workflow</i> de análise e projeto, para reuniões de revisão.
Tempo em re-trabalho	Tempo dos documentos que foram enviados para um <i>workflow</i> de teste e retornaram ao <i>workflow</i> de implementação.
Precisão da estimativa de cronograma de todo o projeto, precisão das estimativas de cronograma para as fases de análise, projeto, codificação, testes de unidade feitos por analistas, testes do sistema e testes para homologação.	Somatório dos tempos estimados para os <i>workflows</i> de Requisitos até Teste. Isto é, o maior somatório de tempos dos <i>workflows</i> considerando que etapas podem ser paralelizadas. Tempo do caminho crítico dos workflows.
Esforço Total do Projeto	Somatório das pessoas/unidade de tempo envolvidas na utilização em todos os <i>workflows</i> desde requisitos até testes. (*)
Esforço na fase de análise e projeto	Total de pessoas/unidade de tempo utilizadas no <i>workflow</i> de análise e projeto. (*)
Esforço na fase de codificação	Total de pessoas/unidade de tempo utilizadas no <i>workflow</i> de implementação. (*)
Esforço na fase de teste de unidade feito por analistas.	Total de pessoas/unidade de tempo utilizadas nos fluxos de testes de unidade. (*)
Esforço na fase de teste de sistema e homologação	Total de pessoas/unidade de tempo envolvidas no <i>workflow</i> de teste, envolvendo sistema e homologação. (*)
Número de erros na especificação de requisitos e no projeto do sistema encontrados em reuniões de revisão.	Número de documentos que estão em um <i>workflow</i> posterior à análise e projeto e que retornaram a esse <i>workflow</i> .
Número de erros no código encontrados no teste de unidade feitos por analistas	Número de documentos que retornam para o gerente, no <i>workflow</i> de implementação, após fluxo de teste de unidade.
Número de modificações na especificação de requisitos, projeto ou código após a sua aprovação.	Número de documentos que estão em um <i>workflow</i> posterior a requisitos e que retornaram a esse <i>workflow</i> .

(\*) Após a coleta dessas métricas, é necessário um ajuste nas informações, que deve ser feito pelas pessoas envolvidas nas atividades do projeto.

Na tabela 1 o conceito de documento é relativo em cada etapa do *workflow*. No *workflow* de requisitos entende-se como documento, um requisito. No *workflow* de análise e projeto como a especificação para solução do requisito. No *workflow* de implementação como uma tarefa da especificação que pode ser atendida por um desenvolvedor. E, finalmente, no *workflow* de teste, documento significa a validação ou não uma funcionalidade, que pode envolver mais de um requisito.

O tempo total utilizado em um *workflow* significa o tempo real despendido para realização de uma atividade do fluxo. Ele é tomado com base no envio e recebimento de documentos dos usuários durante a fase de utilização da ferramenta.

O tempo total estimado um *workflow* é o tempo estimado para realização da atividade, sendo que o somatório do tempo de cada atividade no *workflow* dará o tempo total estimado para realização. Estes tempos são tomados na configuração dos fluxos de cada *workflow*, definidos durante a fase de configuração da ferramenta.

As métricas relativas a esforço são referenciadas como o número de pessoas envolvidas diretamente no projeto em determinado espaço de tempo. Como os *workflow* possuem responsáveis em cada etapa, o número de pessoas envolvidas no projeto são aquelas que enviam e recebem documentos no transcorrer da utilização do *workflow*. Depois de coletadas essas métricas sofrem um ajuste pelas pessoas que trabalharam na atividade, pois elas podem estar executando mais de uma atividade no projeto ou envolvidas em mais de um projeto ao mesmo tempo. Assim, um ajuste nesse valor deve ser realizado.

Algumas métricas não estão sendo mapeadas ainda, pois não existe uma forma de recuperar os valores para realizar as medições. Por exemplo, a métrica tamanho do sistema, que é medida como o número total de linhas de código, não é possível de ser conhecida pois o *workflow* projetado para o projeto não conhece essa informação. Uma alternativa seria a entrada da informação pelo usuário desenvolvedor, do número de linhas que desenvolveu em determinada tarefa, como sendo um atributo da aplicação de abertura configurada no fluxo do *workflow*.

## 5 Conclusões

Gerenciar projetos de software não é uma tarefa simples. Controlar grandes equipes de desenvolvimento, de modo que o aproveitamento das mesmas seja satisfatório, exige um acompanhamento próximo, com a utilização de técnicas e modelos de métricas que quantificam e qualificam o andamento do projeto.

Este artigo apresentou uma ferramenta de *workflow* que controla o processo de desenvolvimento de software de maneira configurável e dinâmica, permitindo a identificação de métricas durante o processo de desenvolvimento.

Uma questão importante é a característica dinâmica da ferramenta, onde cada projeto pode ser configurado conforme a necessidade, possuindo *workflows* próprios, e o mapeamento realizado para a determinação das métricas continua válido, permitindo uma certa independência do processo de desenvolvimento com os valores das métricas e, ao mesmo tempo, mantendo uma integração entre eles.



Esta característica dinâmica apresentada pela ferramenta na configuração dos *workflows* de cada projeto, permite que o processo de desenvolvimento de software seja definido de forma clara e precisa, evitando problemas de interpretação pela equipe de desenvolvimento e permitindo que os valores coletados sejam válidos para a medição do processo de software.

Outra característica é a definição dos valores a serem medidos em fases de configuração e não durante a utilização da ferramenta, liberando o desenvolvedor do preenchimento de informações específicas para medição. Informações estas, que são derivadas do processo de desenvolvimento.

Atualmente, esta ferramenta está sendo utilizada no Centro de Processamento de Dados, da Universidade Federal de Santa Maria (UFSM) para controlar o processo de desenvolvimento de *software* na instituição, nos diversos projetos existentes. Após a aplicação desta ferramenta, adquiriu-se uma qualidade maior no produto final desenvolvido e um controle mais preciso nos prazos de execução das atividades de cada projeto.

Entre os trabalhos futuros destaca-se a criação de uma ferramenta de avaliação das métricas resultantes dos *workflows*, identificando alertas em etapas que não estejam de acordo com o plano de execução previsto, servindo como referência para o gerente de projeto identificar problemas e projetar soluções, mostrar de forma clara para o cliente a atual situação do projeto e fornecer uma justificativa concreta da necessidade de gastar mais recursos para a realização do projeto.

## Referências

- Rocha, A.R.C., et al. (1994), “Uma experiência na Definição do Processo de Desenvolvimento e Avaliação de Software segundo as normas ISSO”, Relatório Técnico ES-302;94, COPPE-UFRJ
- Nita, E.F. (2002), “Melhoria da Qualidade de Produto e de Processo de Software a partir da Análise de Indicadores de Teste”. Simpósio Brasileiro de Engenharia de Software. Gramado, RS.
- Duarte, K.C., Falbo, R.A. (2000) *Uma Ontologia de Qualidade de Software*. Anais do Workshop de Qualidade de Software. Simpósio Brasileiro de Engenharia de Software, João Pessoa.
- Baumert, J.H., Mc Whinney M.S., Software Measures and the Capability Maturity Model, Documento no. CMU-SEI-TR-25, Carnegie Mellon University Software Engineering Institute, 1992.
- Gomes, A., et al. (2000), Medição e Melhoria de Processos de Software. Anais do Workshop de Qualidade de Software, Simpósio Brasileiro de Engenharia de Software, Joao Pessoa.
- Oman P., Pfleger S.L., *Applying SoftwareMetrics*, Los Alamitos, CA, IEEE Computer Society.
- Gomes, A., Oliveira K., Rocha, A.R. (2001) *Avaliação de Processos de Software Baseada em Medições*. Simpósio Brasileiro de Engenharia de Software. Rio de Janeiro, RJ.