

Inspector: Um Processo de Avaliação de Progresso para Projetos de Software

Javé Barbosa de Meneses
Hermano Perrelli de Moura

Centro de Informática
Universidade Federal de Pernambuco
Caixa Postal: 7851, 50732-970, Recife-PE, Brasil
{jbm, hermano}@cin.ufpe.br

Resumo

Este artigo propõe um processo para avaliação de progresso de projetos de software denominado Inspector. O conjunto de métricas definidas no Inspector visam permitir ao gerente de projeto recuperar informações sobre o desempenho das equipes de desenvolvimento e o progresso técnico do projeto. Para que o cálculo, a manipulação e o armazenamento das métricas sejam realizados eficientemente, foram definidos responsáveis, artefatos e atividades que indicam passo a passo como se realizar o acompanhamento preciso do projeto, identificando e solucionando problemas existentes ou que ainda poderiam vir a acontecer.

Abstract

This paper proposes a process for progress evaluation of software projects called Inspector. The set of metrics defined by Inspector provides the project manager with information about the development teams performance and the project technical progress. To guarantee that the calculate, use and storage of these metrics can be realized efficiently was defined workers, artifacts and activities witch guide step by step how realize the precise monitoring of projects, identifying and solving existents problems or that still could be occur.

Palavras-Chave: Gerenciamento de Projeto, Métricas de Software, Avaliação de Progresso de Projetos de Software.

1. Introdução

No cenário econômico e tecnológico atual as transformações no mercado são constantes. As organizações que buscam o sucesso empresarial devem estar preparadas para aceitar grandes desafios técnicos e o aproveitar as oportunidades de negócios através da implantação de um processo contínuo de transformação e melhoria. A cada dia torna-se mais importante a utilização de técnicas e conceitos de engenharia de software que surgem como aliados em um ambiente onde o amadorismo e a produção não controlada de software não se enquadram, cedendo lugar para o desenvolvimento sistemático e gerenciado, aumentando a qualidade dos produtos desenvolvidos e a produtividade dos membros da organização [Pre99, Hum90]. Nesse contexto, a utilização de métricas de software consiste em um fator essencial de auxílio ao gerente de projeto na captura das informações relevantes para o gerenciamento da qualidade do produto e do processo de desenvolvimento, proporcionando um melhor entendimento das relações existentes entre as atividades e as entidades que as mesmas afetam: produto ou processo [Que99].

Este trabalho apresenta a definição do Inspector, que consiste em uma proposta para solucionar as dificuldades encontradas pelo gerente de projeto durante o acompanhamento de projetos de software, mais especificamente projetos de software orientado a objetos. O Inspector define um conjunto de métricas de avaliação de progresso que possibilitam a análise e tratamento estatístico do *status* do projeto em um determinado momento. Além desse conjunto de métricas, foi definido também um conjunto de responsáveis, artefatos e atividades que representam um processo de avaliação de progresso que visa fornecer maturidade suficiente à organização de modo a torná-la capaz de configurar e manipular as métricas de avaliação de progresso definidas. Desse modo, o trabalho realizado tem como objetivos principais:

- Definir métricas que buscam recuperar valores realísticos sobre o *status* atual do projeto e o desempenho das equipes envolvidas;
- Definir um processo que permita à organização fazer bom uso das métricas definidas; e
- Realizar a aplicação prática do processo e das métricas em um estudo de caso.

Além desta seção introdutória, o artigo se divide em outras seis seções. A Seção 2 focaliza as métricas de avaliação de progresso, salientando a participação das mesmas no gerenciamento de projeto. A Seção 3 detalha as métricas de progresso definidas, a importância do resultado que elas representam e o modo como são calculadas. A Seção 4 contém a descrição dos diversos elementos que compõem o Inspector, apresentando uma visão geral do processo e das pessoas envolvidas na sua aplicação. A Seção 5 apresenta a metodologia utilizada na aplicação do Inspector sobre um projeto e organização reais, detalhando as melhorias e os resultados obtidos durante a realização deste estudo de caso. A Seção 6 contempla alguns trabalhos relacionados fazendo um rápido paralelo destes com o Inspector. Por fim, a Seção 7 apresenta as considerações finais, perspectivas e trabalhos futuros sobre o processo de avaliação proposto.

2. Métricas de Avaliação de Progresso

Antes de entrar em detalhes com relação as métricas de avaliação de progresso é importante responder uma pergunta essencial para o entendimento de tais métricas: qual o significado de *progresso*? Fala-se muito em progresso no desenvolvimento de software, mas, muitas vezes, passa-se despercebido que a noção de progresso não é bem definida. O progresso de um determinado projeto de software corresponde a um indicador¹ que fornece informação sobre quão bem o projeto está sendo realizado com respeito ao cronograma de atividades e aos serviços que devem ser fornecidos para o cliente [BM92]. Um indicador é quantificado através de métricas que resultam em valores que permitem obter uma avaliação precisa do *status* do indicador.

Métricas de avaliação de progresso são usadas para monitorar as atividades durante o ciclo de vida de desenvolvimento do software [Hen96]. Cada etapa do ciclo de vida tem atividades que devem ser realizadas e que podem ser quantitativamente medidas com métricas de processo. Além disso, cada atividade produz saídas tangíveis tais como código fonte ou documentação do projeto de algum subsistema que podem ser efetivamente monitorados com métricas de produto. Tais métricas fornecem ao gerente informações não somente sobre o produto, mas também sobre o processo utilizado para desenvolver o produto. As métricas de avaliação de progresso podem ser classificadas em três categorias distintas, de acordo com o

¹ Um indicador é uma propriedade utilizada para monitorar algum aspecto do projeto.

foco da avaliação. São elas: desempenho, progresso das unidades de trabalho e capacidade incremental [BM92].

As métricas de desempenho fornecem informações básicas sobre o progresso no cronograma de atividades e eventos de um determinado projeto. Tais métricas também ajudam a identificar e avaliar dependências entre atividades e eventos relativos ao desenvolvimento de software [Joi98]. Alguns exemplos de métricas utilizadas para avaliação do desempenho do projeto são: número de iterações (somente para o caso de processo de desenvolvimento iterativo²) que permite realizar uma validação prévia do esforço gasto no desenvolvimento [BM92], número de contratos concluídos que verifica quanto dos serviços que o sistema deve fornecer para satisfação do cliente foram desenvolvidos [LK94], e datas dos marcos de referências³ que mede o desempenho da equipe de desenvolvimento para a realização marcos inicialmente previstos [BJR99a].

Métricas de progresso das unidades de trabalho se preocupam em verificar o progresso baseado no quão completo estão as unidades de trabalho que se combinam incrementalmente para formar um produto ou atividade de software completa [Joi98]. Unidades de trabalho correspondem aos produtos que são gerados e documentados durante o desenvolvimento do projeto, por exemplo, componentes de software, requisitos, casos de teste, etc. Se forem definidos critérios objetivos que indicam se uma unidade está ou não corretamente concluída, tais métricas se tornam extremamente efetivas para avaliar progresso em qualquer ponto do projeto. Algumas das principais métricas utilizadas para avaliação das unidades de trabalho do projeto são: *status* dos componentes que conta o número de componentes de software que completaram uma atividade específica do desenvolvimento, *status* dos requisitos que conta o número de requisitos que foram alocados à componentes de software e casos de teste, *status* dos casos de teste que conta o número de casos de teste especificados e realizados com sucesso [Joi98].

Por fim, métricas que indicam a capacidade incremental verificam o incremento funcional no conteúdo do produto associado a cada distribuição (iteração). Uma distribuição incremental pode ser um produto a ser entregue para o cliente, ou um *build*⁴ interno distribuído para a próxima fase do desenvolvimento [Joi98]. Estas métricas são usadas para determinar se as funcionalidades previstas estão sendo desenvolvidas como planejado. Algumas das principais métricas utilizadas para avaliação da capacidade incremental do projeto são: número de componentes do *build* que identifica os componentes incluídos em *builds* incrementais e número de funcionalidades do *build* que identifica a funcionalidade incorporada em *builds* incrementais [Joi98].

3. As Métricas de Progresso Definidas

Foram definidas quatro métricas para avaliação de progresso que serão utilizadas para controlar o desenvolvimento do projeto, sendo três voltadas para avaliação do desempenho das equipes de desenvolvimento e uma métrica genérica que observa o progresso nas unidades de trabalho e na capacidade incremental do projeto. As duas subseções seguintes descrevem

² Um processo de desenvolvimento iterativo representa o desenvolvimento do sistema através de um conjunto de iterações, onde cada iteração representa um mini-projeto [BJR99a].

³ Um marco de referência representa a realização de alguma atividade que indique, claramente, que o projeto obteve progresso no seu desenvolvimento.

⁴ Um *build* é um conjunto de componentes que compõem o sistema em uma fase qualquer do desenvolvimento deste.

com maiores detalhes as métricas de desempenho e de progresso funcional definida no Inspector.

3.1 Métricas de Desempenho

O Inspector considera interessante o gerente de projeto utilizar cronogramas de atividades, representados através de gráficos Gantt [Hen96] e PERT [BM92], para monitorar as atividades planejadas para as equipes de desenvolvimento. A representação gráfica permite observar a dependência entre as atividades planejadas e facilita a identificação de atividades em atraso. Apesar de importante, percebe-se que gráficos de atividades não são suficientes no gerenciamento do desempenho das equipes em um determinado projeto [BM92, Hen96, LK94]. É necessária a utilização de métricas que indiquem quantitativamente o desempenho de cada equipe e que permitam manter uma base histórica para consultas e comparações com projetos futuros. Desse modo, o Inspector define três métricas que auxiliam na avaliação do desempenho das equipes de desenvolvimento. São elas: $\mu_{concluído}$, $\mu_{atraso\ médio}$ e $\mu_{novas\ atividades}$.

A métrica $\mu_{concluído}$ representa o quanto uma determinada equipe concluiu das atividades que estavam inicialmente planejadas para terminarem dentro do escopo de tempo da avaliação. Desse modo, $\mu_{concluído}$ é dada pela seguinte equação (3.1).

$$\mu_{concluído}(equipe) = \frac{\sum_{x \in Atividades(equipe)} tempoEstimado(x) \times concluído(x)}{\sum_{x \in Atividades(equipe)} tempoEstimado(x)} \quad (3.1)$$

Onde:

- $equipe$ indica a equipe que está sendo avaliada;
- $Atividades(equipe)$ representa o conjunto de atividades relacionadas com $equipe$ que estavam inicialmente planejadas para serem concluídas no escopo de tempo da avaliação;
- $concluído(x)$ consiste na função que retorna o quanto da atividade x se realizou, com $0 \leq concluído(x) \leq 1$; e
- $tempoEstimado(x)$ é a função que retorna o tempo estimado para a atividade x , ou seja, é a função que indica o quanto se espera que a atividade x demore para ser realizada.

A métrica $\mu_{atraso\ médio}$ indica quantitativamente o percentual de atraso médio por atividade de uma determinada equipe de desenvolvimento e é dada pela equação (3.2). Em toda avaliação de progresso, o escopo da avaliação define o intervalo de tempo que esta avaliação irá considerar, consequentemente, serão analisadas somente as atividades que estavam planejadas para serem

$$\mu_{atraso\ médio}(equipe) = \frac{\sum_{x \in Atividades(equipe)} \frac{tempoEstimado_f(x) - tempoEstimado_i(x)}{tempoEstimado_i(x)}}{\# Atividades(equipe)} \quad (3.2)$$

Onde:

- $equipe$ indica a equipe que está sendo avaliada;
- $Atividades(equipe)$ representa o conjunto de atividades relacionadas com $equipe$ que estavam inicialmente planejadas para serem concluídas no escopo de tempo da avaliação;
- $tempoEstimado_i(x)$ é a função que retorna o tempo inicialmente estimado para a atividade x ;
- $tempoEstimado_f(x)$ é a função que retorna o tempo gasto para realização da atividade x caso ela tenha sido completada ou, caso contrário, retorna a estimativa do novo tempo necessário para realização desta atividade; e
- $\# Atividades(equipe)$ é a cardinalidade de $Atividades(equipe)$.

executadas dentro desse intervalo de tempo. Porém, muitas vezes surgem novas atividades não planejadas que alocam o pessoal da equipe e prejudica o planejamento inicial e o desempenho. O cálculo da métrica $\mu_{novas\ atividades}$ tem como objetivo indicar a porcentagem média de tempo gasto com novas atividades que surgem dentro do escopo da avaliação. Assim sendo, valores altos para $\mu_{novas\ atividades}$ representam problemas de planejamento que aumentam os riscos no desenvolvimento do projeto e podem indicar problemas graves no desempenho das equipes. A métrica $\mu_{novas\ atividades}$ é calculada através da equação (3.3).

$$\mu_{novas\ atividades} = \frac{\sum_{x \in Atividades\ Iniciadas} tempoGasto(x) - \sum_{x \in (AtividadesIniciadas \cap AtividadesPlanejadas)} tempoGasto(x)}{\sum_{x \in AtividadesIniciadas} tempoGasto(x)} \quad (3.3)$$

Onde:

- *AtividadesIniciadas* é o conjunto que contém as atividades realmente iniciadas dentro do escopo da avaliação;
- *AtividadesPlanejadas* é o conjunto que contém as atividades planejadas para serem iniciadas dentro desse escopo; e
- *tempoGasto(x)* é a função que retorna a quantidade de tempo já trabalhado para realização da atividade *x*.

Assim, $\mu_{novas\ atividades}$ identifica o tempo gasto com atividades que foram iniciadas, ou seja, que pertencem ao conjunto *AtividadesIniciadas*, mas não foram planejadas, ou seja, não pertencem ao conjunto *AtividadesPlanejadas* e, em seguida, calcula o percentual que estas atividades representam em relação ao total de atividades iniciadas.

3.2 Métrica de Progresso Funcional

Durante a definição do processo de avaliação de progresso verificou-se a necessidade de utilizar uma métrica (ou um conjunto de métricas) que permitisse visualizar o *status* do projeto de modo mais técnico, apresentando claramente o que foi desenvolvido e o que falta desenvolver. Desse modo, foi definida a métrica $\mu_{sistema}$ que indica quantitativamente o aspecto funcional já inserido no sistema. Esta métrica permite ao gerente de projeto fazer diversas considerações, tais como: visualizar o crescimento percentual das funcionalidades desenvolvidas em relação às avaliações anteriores, comparar com resultados obtidos em outros projetos, além de permitir a análise de tendências que permitam identificar possíveis problemas no desenvolvimento.

A métrica $\mu_{sistema}$ apresenta algumas propriedades que a tornam mais adequada para projetos de software orientado a objetos que utilizem modelagem UML [BJR99b, Qua99] para análise e projeto do problema a ser resolvido. São elas:

- Dirigida a casos de uso;
- Baseada na inspeção de artefatos; e
- Facilidade de observação.

A métrica proposta verifica a incorporação de novas funcionalidades dentro do sistema a partir da inspeção dos artefatos que devem ser produzidos para realização de um determinado caso de uso. Dessa forma, o cálculo do progresso é baseado nos casos de uso do sistema que servirão como um guia, ou seja, o progresso do sistema como um todo será obtido a partir do cálculo do progresso funcional de cada caso de uso. A seguir serão apresentadas as principais equações relacionadas ao cálculo de $\mu_{sistema}$ para um projeto específico, sendo que uma descrição detalhada da métrica pode ser encontrada em [MM00, MM01a, MM01b].

A métrica $\mu_{sistema}$ é derivada a partir da equação definida por Champeaux [Cha97] que indica o valor de uma métrica composta como sendo o somatório da contribuição de cada artefato que compõe a métrica, ou seja:

$$\mu_x = \sum_i D_i \quad (4.1)$$

Onde:

- i representa o conjunto de artefatos relevantes à x ; e
- D_i representa a contribuição que o artefato i representa para a métrica μ_x .

Como o progresso funcional do projeto deriva diretamente do progresso funcional dos diversos casos de uso relacionados ao sistema, temos que a seguinte equação pode ser retirada a partir da Equação 4.1.

$$\mu_{sistema}(i) = \frac{\sum_{casodeuso \in DCU} p_{casodeuso}(i) \times \mu_{casodeuso}(i)}{\sum_{casodeuso \in DCU} p_{casodeuso}(i)} \quad com \quad 1 \leq i \leq n \quad (4.2)$$

Onde:

- $p_{casodeuso}$ indica o peso do caso de uso para o progresso do sistema;
- $\mu_{casodeuso}$ indica o progresso do caso de uso;
- DCU representa o conjunto do(s) Diagrama(s) de Caso de Uso do sistema;
- i indica a iteração na qual o projeto se encontra no momento da avaliação; e
- n representa o número de iterações planejadas para o projeto.

Uma característica interessante observada durante a definição de $\mu_{sistema}$ foi a necessidade de se considerar a possibilidade de avaliação de progresso em um processo de desenvolvimento iterativo. A métrica considera a iteração na qual a avaliação está sendo realizada e, por conseguinte, identifica o progresso em relação ao número de iterações planejadas. É importante notar que o valor relacionado ao peso que o caso de uso tem para o sistema pode mudar de iteração para outra, de acordo com um conjunto de indicadores que demonstram as prioridades do cliente e os riscos envolvidos com o desenvolvimento do caso de uso. Outra consideração relevante é que para processos de desenvolvimento que não utilizam o modelo de ciclo de vida iterativo, basta se fazer i igual a 1 para toda avaliação de progresso realizada durante o projeto.

De acordo com a Equação 4.2 é necessário calcular o progresso de cada caso de uso no cálculo de $\mu_{sistema}$. A métrica relacionada ao progresso de um caso de uso considera as principais etapas necessárias para o seu desenvolvimento, partindo desde a identificação e documentação inicial do caso de uso até seu desenvolvimento e integração, como mostrado na Equação 4.3.

O progresso de um caso de uso em uma determinada etapa ($\mu_{casodeuso}(i, j)$) é calculado a partir da contribuição dos diversos artefatos necessários para que esta etapa seja realizada. Desse modo, temos a função $\mu_{casodeuso}(i, j) \equiv \mu_{casodeuso}(i, j)(A_{ij})$ que representa a dependência da métrica em relação aos artefatos, com A_{ij} representando o conjunto de artefatos relevantes para a realização do caso de uso na iteração i e etapa de desenvolvimento j .

$$\mu_{casodeuso}(i) = \frac{\sum_{j \in ETAPAS} p_{casodeuso}(i, j) \times \mu_{casodeuso}(i, j)}{\sum_{j \in ETAPAS} p_{casodeuso}(i, j)} \quad (4.3)$$

Onde:

- i indica a iteração atual;
- j representa a etapa de desenvolvimento do caso de uso avaliada;
- $p_{casodeuso}(i, j)$ indica o peso relativo ao caso de uso na iteração i e etapa de desenvolvimento j ;
- $\mu_{casodeuso}(i, j)$ indica o progresso do caso de uso na iteração i e etapa de desenvolvimento j ; e
- $ETAPAS$ corresponde ao conjunto de etapas para o desenvolvimento do caso de uso. Um valor padrão esse conjunto é: $ETAPAS = \{\text{especificação inicial, análise e projeto, implementação, teste}\}$, mas nada impede que esse conjunto seja configurado para cada caso de uso identificando as etapas que realmente devem ser realizadas para sua definição.

Novamente seguindo a Equação 4.1 (equação de composição proposta por Champeaux [Cha97]), definiu-se:

$$\mu_{casodeuso}(i, j) = \frac{\sum_{x \in A_{ij}} p_x(casodeuso) \times \mu_x(casodeuso)}{\sum_{x \in A_{ij}} p_x(casodeuso)} \quad (4.4)$$

Onde:

- $p_x(casodeuso)$ indica o peso do artefato x para construção de $casodeuso$;
- $\mu_x(casodeuso)$ representa a métrica que indica se o artefato x já foi produzido ou alterado, de modo a realizar a etapa j no desenvolvimento do caso de uso; e
- A_{ij} representa o conjunto de artefatos que devem ser produzidos ou atualizados na iteração i e etapa j .

A configuração do conjunto de artefatos que devem ser produzidos no desenvolvimento de um caso de uso e que, portanto, deverão ser inspecionados durante a avaliação de progresso do caso de uso, é realizada durante o planejamento da avaliação de progresso. Cada artefato identificado deve conter critérios de avaliação que permitem identificar se tais artefatos incorporam as informações necessárias ao desenvolvimento do caso de uso. O trabalho realizado definiu superficialmente alguns critérios de inspeção que não capturam problemas semânticos nem avaliam a qualidade dos artefatos produzidos, podendo critérios mais completos serem retirados a partir de trabalhos existentes na área de inspeção de projetos de software [TSF⁺99]. Uma descrição detalhada do conjunto inicial de artefatos considerados no Inspector com seus respectivos critérios de inspeção, incluindo os vários diagramas UML [BJR99b, Qua99], podem ser encontrados em [MM00, MM01a, MM01b].

A métrica de avaliação do progresso funcional pode ser representada de duas maneiras distintas: tabelas e gráficos. A tabela resume todos os valores relativos ao *status* atual do projeto e dos seus diversos casos de uso de acordo com as etapas de desenvolvimento (vide Tabela 1).

Já na forma de representação gráfica, cada gráfico considera o progresso de apenas um elemento do projeto, ou do projeto como um todo. Desse modo, cada caso de uso tem um gráfico associado indicando seu progresso em relação ao tempo. Além disso, existe um gráfico que expressa o incremento no progresso global do projeto em relação ao tempo (vide Figura 1). Os dois tipos de representação permitem avaliar o progresso funcional do projeto

Caso de Uso	Especificação Inicial	Análise e Projeto	Implementação	Teste	Progresso Total
1	0,66	0,66	0	0	0,33
2	1	1	0,66	0,50	0,79
3	0,75	0,66	0	0	0,35
4	0,75	0,66	0	0	0,35
5	1	1	1	1	1
Projeto	0,94	0,92	0,64	0,58	0,77

Tabela 1. Exemplo de tabela resumo do progresso funcional

em termos de cada funcionalidade individualmente ou do projeto como um todo, permitindo comparar com resultados obtidos em projetos anteriores e realizar análise de tendências sobre os gráficos de linha em busca de problemas em funcionalidades (casos de uso) específicas.

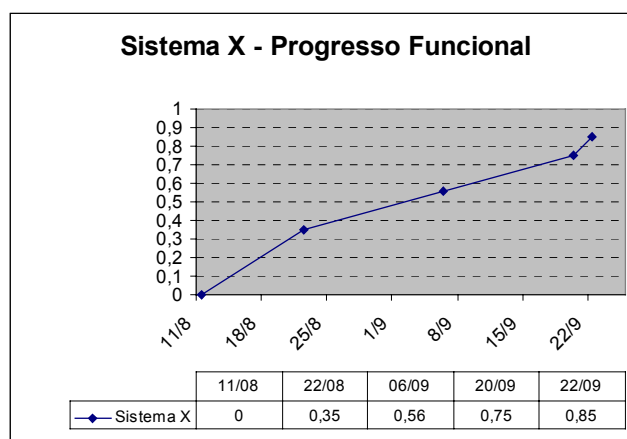


Figura 1. Exemplo de gráfico de linha para representação de $\mu_{sistema}$.

4. O Inspector

As métricas definidas no Inspector e apresentadas na Seção 3, quando calculadas e analisadas corretamente fornecem ao gerente de projeto duas visões do progresso do sistema, permitindo um acompanhamento preciso do desempenho das equipes de desenvolvimento e do progresso técnico do projeto. São elas: visão de desempenho e visão de funcionalidade. A visão de desempenho verifica a realização das atividades planejadas para cada equipe de desenvolvimento no período de tempo previsto através de um acompanhamento sistemático do cronograma de atividades e do cálculo de $\mu_{concluído}$, $\mu_{atraso\ médio}$ e $\mu_{novas\ atividades}$, permitindo ao gerente de projeto verificar a necessidade de inserir, realocar ou retirar pessoal do projeto. Já a visão de funcionalidade define o *status* das funcionalidades previstas para o sistema através do cálculo de $\mu_{sistema}$, observando o progresso no desenvolvimento do sistema, dos vários subsistemas que o compõem e dos casos de uso pertencentes aos subsistemas. Assim, esta visão é eficiente para se identificar problemas técnicos relacionados com a definição e construção dos artefatos do projeto, capturando deficiências no desenvolvimento que dificilmente seriam recuperadas pela visão de desempenho.

A obtenção destas duas visões não é uma tarefa simples e para tornar sua utilização sistemática e eficiente o Inspector define um conjunto de responsáveis, artefatos e atividades que apresentam passo a passo o trabalho envolvido com a adaptação da organização, planejamento, coleta, cálculo e avaliação das métricas de progresso para um determinado

projeto. A Figura 2 mostra o fluxo de atividades definido no Inspector contendo as atividades, a dependência de execução entre tais atividades e o responsável por cada atividade.

As subseções seguintes descrevem brevemente cada uma das atividades definidas no Inspector. Devido ao limite de espaço para apresentação não será possível detalhar os vários passos e técnicas definidos para realizar cada atividade, nem os artefatos que devem ser recuperados e produzidos durante a aplicação do processo. A versão estendida do Inspector detalha todos os artefatos, responsáveis e atividades envolvidos [MM00, MM01a, MM01b].

Avaliar Status das Métricas na Organização

Para que o Inspector seja implantado com sucesso na organização e instanciado para um projeto específico, é necessário entender o contexto de desenvolvimento do projeto, ou seja, o estado corrente da organização, em relação ao desenvolvimento de software e à utilização de métricas. O Engenheiro de Processo é responsável por capturar este entendimento a partir dos membros da organização, dos processos de desenvolvimento e gerenciamento aplicados, das ferramentas de suporte utilizadas e da cultura da organização. O artefato Visão Geral das Métricas na Organização é produzido como resultado da observação e análise da organização, descrevendo sua maturidade quanto a utilização das métricas de progresso e relatando as principais dificuldades que deverão ser superadas na aplicação do Inspector.

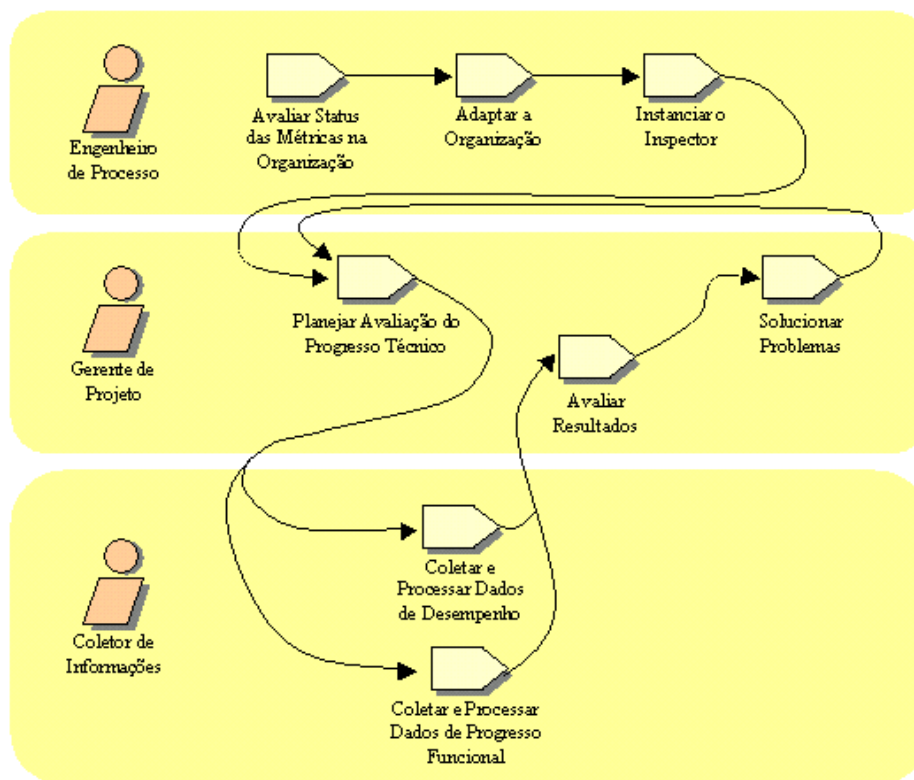


Figura 2. Fluxo de Atividades do Inspector

Adaptar a Organização

Esta atividade tem como responsável o Engenheiro de Processo e tem como objetivo inserir uma cultura de utilização das métricas necessárias para implantação do Inspector dentro da organização. Para tanto é preciso analisar os principais problemas identificados na atividade Avaliar Status das Métricas na Organização, pois a partir dessa análise é possível definir o que deverá ser adaptado na organização e qual a melhor estratégia para se fazer isso. É importante desprender o tempo que se julgar necessário para realizar essa atividade, pois ela

é essencial para que a aplicação do Inspector seja realizada com sucesso. Nela o gerente de projeto deve capacitar os membros da organização para utilização do Inspector e tornar claro para os mesmos que o processo de avaliação não prejudicará ninguém individualmente [Wie99].

Instanciar o Inspector

Esta atividade tem como objetivo adequar o Inspector a um determinado projeto de acordo com as peculiaridades do mesmo. Muitos projetos não produzem determinados artefatos, ou produzem artefatos distintos, de acordo com o seu porte, tipo da aplicação a ser desenvolvida, entre outros fatores. Desse modo é importante entender o projeto e identificar quais artefatos serão realmente produzidos no seu desenvolvimento. Definidos os artefatos, o Inspector deve ser configurado e, conseqüentemente, suas métricas associadas também, para que ele esteja adequado ao projeto específico. Essa atividade resulta no artefato Critérios de Inspeção dos Artefatos onde são apresentados os possíveis artefatos que poderão ser inspecionados durante a avaliação de progresso de um caso de uso, contendo seus respectivos critérios de inspeção.

Planejar Avaliação do Progresso Técnico

Essa atividade é realizada de acordo com a demanda, ou seja, quando surgir a necessidade de se avaliar o progresso do projeto, tanto para fornecer uma satisfação ao cliente sobre o projeto que ele requisitou quanto como forma do Gerente de Projeto acompanhar e monitorar o desenvolvimento. Tem como responsável o próprio Gerente de Projeto que deve produzir o artefato Plano de Avaliação do Progresso Técnico, definindo o escopo da avaliação, quando ela será realizada, quem é o responsável pela coleta das métricas e quais os resultados esperados.

Coletar e Processar Dados de Desempenho

Essa atividade visa recuperar as informações que serão utilizadas para análise do desempenho do projeto, ou seja, os gráficos Gantt e PERT, e as métricas de desempenho $\mu_{concluído}$, $\mu_{atraso\ médio}$ e $\mu_{novas\ atividades}$. Cada Coletor de Informações definido na atividade Planejar Avaliação do Progresso Técnico deve realizar essa atividade, coletando as informações referentes ao escopo que lhe foi atribuído no Plano de Avaliação do Progresso Técnico.

Coletar e Processar Dados do Progresso Funcional

Nessa atividade, cada Coletor de Informações deve capturar as informações relacionadas ao progresso dos casos de uso que fazem parte do seu escopo. O conjunto de casos de uso que o Coletor ficará responsável pela coleta de informações é definido na atividade Planejar Avaliação do Progresso Técnico. A saída dessa atividade é, para cada caso de uso, o Modelo de Coleta de Informação sobre o Caso de Uso preenchido com o progresso funcional do caso de uso calculado.

Avaliar Resultados

Consiste na atividade mais complexa, e que exige maior esforço durante a aplicação do Inspector. Nesse momento o Gerente de Projeto deve obter uma visão geral do progresso do projeto que corresponde a análise das visões de funcionalidade e desempenho.

O Gerente de Projeto deve analisar os dados fornecidos pelos coletores fazendo uma comparação do que foi realizado com o que fora inicialmente planejado. Além disso, ele deve observar mudanças substanciais nos gráficos Gantt e PERT das equipes, monitorar o

desempenho geral das equipes de desenvolvimento ($\mu_{concluído}$, $\mu_{atraso\ médio}$ e $\mu_{novas\ atividades}$), calcular o progresso funcional do projeto ($\mu_{sistema}$) e, a partir desse cálculo, fazer ponderações sobre os resultados obtidos através da representação dos dados em gráficos e tabelas, observando incrementos em relação à última avaliação e analisando tendências para o futuro. Essa atividade produz como saída o artefato Avaliação do Progresso Técnico onde são detalhadas informações relacionadas às duas visões de progresso, identificando problemas encontrados no desempenho das equipes de desenvolvimento e no progresso funcional dos casos de uso e do projeto como um todo.

Solucionar Problemas

Nessa atividade o Gerente de Projeto busca soluções para os problemas encontrados durante a avaliação do progresso técnico do projeto. Essa atividade consiste em um pequeno *framework*, onde cada problema é trabalhado individualmente visando sua eliminação. O Gerente de Projeto, juntamente com as pessoas envolvidas deverão encontrar as possíveis soluções para cada problema e identificar a solução mais adequada à situação atual do projeto. Como saída essa atividade produz o artefato Documento de Solução dos Problemas Identificados que relata a situação de cada solução aplicada para os problemas identificados.

5. Aplicação do Inspector a um Projeto Real

A aplicação do Inspector em um projeto real, denominado Nota Fiscal Virtual, permitiu um melhor acompanhamento do projeto encontrando problemas técnicos no desenvolvimento das funcionalidades e problemas no desempenho das equipes. Além disso, este estudo de caso permitiu realizar melhorias profundas no Inspector para que ele se torne mais aplicável em organizações reais.

5.1 Metodologia Utilizada

O projeto Nota Fiscal Virtual, realizado pela Empresa Municipal de Informática do Recife (Emprel), visa a construção de um sistema que busca tornar a relação da Prefeitura da Cidade do Recife com os seus contribuintes mais amigável e eficiente. Inicialmente foram realizadas reuniões de esclarecimento com a equipe de desenvolvimento, tornando clara a importância de cada membro da equipe na recuperação das informações e no cálculo e avaliação das métricas a serem utilizadas. Nesse momento foi extremamente importante tornar claro que o processo não é usado para prejudicar nenhum membro individualmente, ao contrário, o processo visa melhorar a comunicação entre a equipe e o gerente de projeto, bem como aumentar a produtividade da organização. Realizou-se também uma análise da maturidade da organização através da observação do cotidiano da empresa e de leitura da documentação do processo de desenvolvimento utilizado [Emp00]. O estudo de caso iniciou quando o projeto já estava em andamento e teve a duração de dois meses e quinze dias. Neste período foram realizadas as atividades definidas no Inspector para monitorar o desenvolvimento do projeto e produzidos os artefatos estabelecidos como resultados dessas atividades.

5.2 Resultados das Avaliações Realizadas

Foram realizadas cinco avaliações de progresso que focalizaram os aspectos funcionais incorporados ao projeto Nota Fiscal Virtual, ou seja, o acompanhamento de $\mu_{sistema}$ através da

comparação dos valores alcançados com os esperados e da análise de tendências. A Figura 3 representa a variação do progresso funcional do projeto Nota Fiscal Virtual durante o período em que seu desenvolvimento foi monitorado.

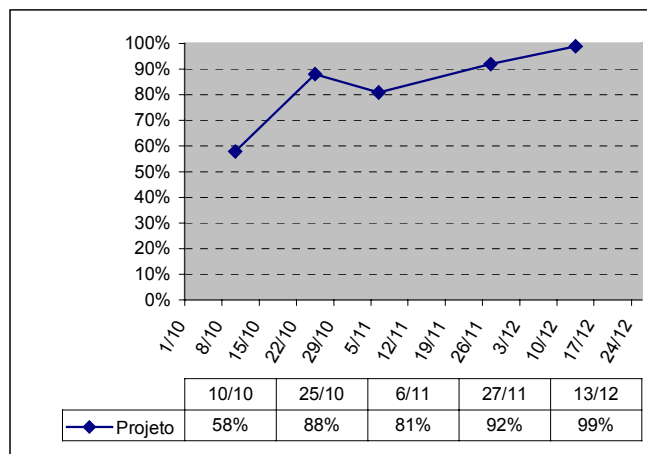


Figura 3. Progresso do projeto durante as avaliações realizadas.

As avaliações de progresso realizadas sobre o projeto permitiram identificar atrasos em casos de uso específicos, distribuindo o esforço de desenvolvimento entre aqueles casos de uso que demonstraram maior dificuldade. Os casos de uso do sistema também foram monitorados através de gráficos que permitem observar o progresso dos mesmos em relação ao tempo. Foram monitorados os oito casos de uso desenvolvidos durante a construção do Nota Fiscal Virtual. A Figura 4 apresenta os resultados obtidos em três dos casos de uso monitorados.

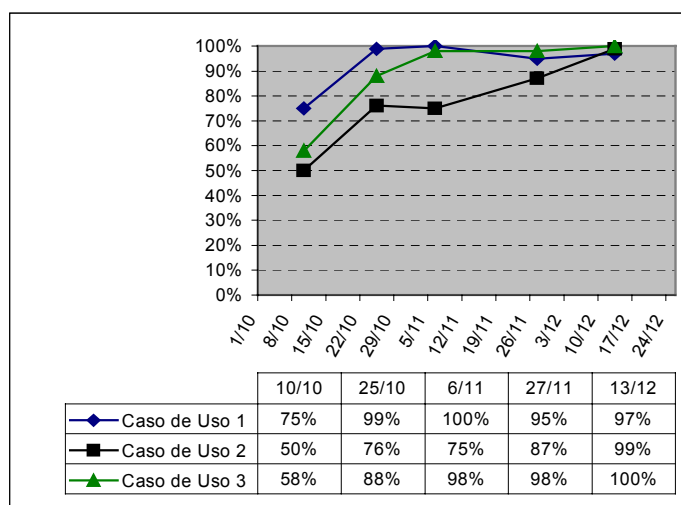


Figura 4. Progresso de três casos de uso durante as avaliações realizadas

É interessante citar que algumas vezes o progresso de um determinado caso de uso ou até mesmo do projeto obteve queda. Poderia se pensar que não é possível haver um decréscimo no progresso funcional de um projeto pois como um trabalho realizado poderia retroceder? A métrica de progresso funcional captura a quantidade de funcionalidade desenvolvida em relação às funcionalidades previstas pelo projeto, assim, o surgimento não esperado de novas funcionalidades pode implicar em um decréscimo no progresso do projeto, como pode ser observado na avaliação realizada em 6/11 (Figura 3). Nesse caso houve um decréscimo de 7%

no progresso em relação à avaliação anterior cujo motivo identificado foi o surgimento de novas funcionalidades a serem desenvolvidas oriundas de uma má elicitação de requisitos. Outras vezes, o progresso do projeto pode retroceder devido à identificação de problemas no desenvolvimento de determinado caso de uso, como por exemplo o caso de uso 1 no dia 6/11 se apresentava como completamente realizado e, no entanto, a próxima avaliação (27/11) revelou um decréscimo de 5% no seu progresso. Esse decréscimo motivou o gerente de projeto a realizar um estudo mais preciso da situação do caso de uso, onde constatou-se junto aos desenvolvedores que o caso de uso estava com problemas na implementação identificados em fase de implantação.

Além da representação gráfica foram geradas tabelas que resumem toda a informação sobre o *status* do projeto em uma avaliação. Tais tabelas são fáceis de serem armazenadas e manipuladas em banco de dados permitindo comparações rápidas e detalhadas. A Tabela 2 representa um exemplo de tabela resumo gerada na primeira avaliação realizada (10/10). É possível observar através da tabela resumo gerada no dia 10/10 que a primeira avaliação realizada localizou cinco casos de uso onde observa-se progresso maior nas etapas de análise e projeto dos casos de uso, sendo que deve ser dada atenção especial ao caso de uso 4 que não havia concluído até aquele momento a documentação de sua especificação inicial, ou seja, $\mu_{\text{caso de uso 4}}(\text{especificação inicial}) = 0,75$.

Caso de Uso	Especificação	Análise e Projeto	Implementação	Teste	Progresso Total
1	1	1	1	0	0,75
2	1	0,66	0,33	0	0,50
3	1	1	0,33	0	0,58
4	0,75	0,66	0,33	0,20	0,49
5	1	1	0,33	0	0,58
Projeto	0,95	0,87	0,46	0,04	0,58

Tabela 2. Progresso funcional obtido na primeira avaliação

6. Trabalhos Relacionados

Existem diversos trabalhos relacionados com as áreas de gerenciamento de projeto e métricas de acompanhamento de projetos de software que por serem necessidades reais das grandes organizações, representam áreas de grande interesse científico e comercial. As métricas propostas pelo Inspector buscam incorporar as características positivas encontradas nos trabalhos relacionados, quantificando o progresso obtido no desenvolvimento das funcionalidades previstas para o sistema. Além disso, tais métricas possuem novas propriedades que contemplam aspectos não capturados pelos trabalhos citados abaixo, tais como: visualizar o progresso do caso de uso através de um mapeamento de artefatos, visualizar o progresso das etapas para o desenvolvimento dos casos de uso e identificar a qualidade do planejamento. A captura dessas propriedades tornam o Inspector uma ferramenta mais completa e eficaz para o acompanhamento de projetos de software.

Arnold e Pedross apresentam em [AP98] um método de medição implantado no *Swiss Banking Institute* que tem como objetivo avaliar a produtividade da organização. Este método, denominado *use case point*, baseia-se na utilização de uma métrica que, assim como μ_{sistema} , é guiada por casos de uso. A idéia do *use case point* é inspirada na análise por pontos de função [CAF⁺98], observando a funcionalidade especificada com casos de uso e cenários, e os fatores técnicos significativos relacionados a tais funcionalidades. Apesar de apresentarem algumas

semelhanças, existem grandes diferenças entre $\mu_{sistema}$ e o *use case point*, tais como: o *use case point* não considera as etapas necessárias para o desenvolvimento de um caso de uso e nem pode ser configurado para se adequar a um determinado processo de desenvolvimento.

Jacquet e Abran [JA97] definem um modelo de processo para utilização de métodos de medição de software. O modelo proposto, de forma semelhante ao Inspector, detalha passos para coletar e avaliar os resultados de métricas de software. A grande diferença é que o primeiro representa um modelo genérico de processo de utilização de métricas, sendo independente da métrica a ser utilizada e do aspecto focalizado (progresso, qualidade, entre outros). Este modelo apresenta também um interessante *framework* para analisar a validade de uma métrica.

Além dos trabalhos citados anteriormente, outros trabalhos encontrados em [CK94, FR90, Joi98, KSW99, LY98, Que99, Sch99, Par92] buscam auxiliar o gerente de projeto no controle da qualidade do processo de desenvolvimento e do produto em construção.

7. Conclusões

O Inspector utiliza a manipulação de métricas como técnica para encontrar problemas no desenvolvimento e auxiliar o gerente na tomada de decisões. Porém, se observou que não basta definir um conjunto de métricas de progresso prontas para serem usadas, é necessário que a organização esteja disposta e preparada para coletar, calcular e avaliar tais métricas, seguindo um série de passos bem definidos que permitem identificar desvios nos valores esperados para o desempenho das equipes e progresso das funcionalidades, indicando problemas no desenvolvimento e permitindo que o gerente identifique soluções para estes problemas. O Inspector representa, desse modo, uma tentativa de formalização das tarefas que devem ser realizadas para utilização sistemática de um conjunto de métricas de avaliação de progresso. Os responsáveis, artefatos e atividades são elementos básicos para o gerenciamento de projetos dentro do Inspector e foram definidos buscando melhorar a comunicação entre os membros da organização, garantindo que os resultados encontrados sejam condizentes com a realidade do projeto. As subseções seguintes apresentam brevemente trabalhos futuros que podem ser realizados para aperfeiçoar o Inspector e algumas considerações finais sobre o trabalho realizado.

7.1 Trabalhos Futuros

Apesar de representar um importante auxílio para gerentes de projeto tornarem o processo de desenvolvimento mais controlável e mensurável, pode-se perceber que a aplicação do Inspector não é uma tarefa trivial. Ela exige grande dedicação, esforço e disciplina na coleta, cálculo e avaliação das métricas tanto de desempenho quanto de progresso funcional. Assim sendo, é interessante a construção de uma ferramenta que automatize ou pelo menos semi-automatize o Inspector, de forma que a recuperação dos artefatos, os cálculos que devem ser realizados e a geração dos gráficos e tabelas resultantes da avaliação de progresso sejam realizados automaticamente, bastando ao gerente de projeto introduzir alguns parâmetros de configuração, tais como o conjunto de artefatos e etapas necessários para desenvolvimento do caso de uso, os pesos relacionados a cada caso de uso, entre outros parâmetros. É importante salientar que para o gerente de projeto configurar corretamente o Inspector é necessário que ele tenha grande conhecimento do processo de desenvolvimento utilizado pela organização. Uma descrição mais detalhada de tal ferramenta pode ser encontrada em [MM01a].

Além de automatizar o Inspector, é importante também superar algumas limitações nas métricas definidas pelo Inspector, especialmente na métrica de progresso funcional $\mu_{sistema}$ para melhorar a efetividade dos resultados obtidos durante a aplicação do processo. Um trabalho interessante seria adaptar $\mu_{sistema}$ de modo que ela se torne capaz de capturar o progresso relativo ao desenvolvimento dos requisitos não funcionais [Chu91].

Outro trabalho interessante busca ampliar o escopo do Inspector de forma a capturar todos os aspectos do gerenciamento de projetos, identificando não somente o andamento do projeto, mas observando também a qualidade do produto que está sendo desenvolvido, as relações entre os membros da equipe, as relações da organização com o cliente e preocupando-se em mapear riscos e características não técnicas que influenciam o progresso de um projeto, tais como o conhecimento imperfeito dos estimadores no início do projeto, o conhecimento e experiência da equipe de desenvolvimento, complexidade do sistema, entre outros.

7.2 Considerações Finais

Este trabalho abre espaço para realização de diversos novos trabalhos em uma área ainda bastante nova e que necessita de pesquisa e incentivo com o intuito de fornecer maior suporte para o processo de desenvolvimento de software. Um processo que formalize a utilização de um conjunto de métricas representa grande interesse para grandes empresas, que necessitam de um maior controle sobre seus projetos e estão dispostas a investir para obter tais recursos.

O Inspector demonstrou ser bem aplicável no estudo de caso realizado (projeto Nota Fiscal Virtual), despertando o interesse de empresas atuantes na área de tecnologia da informação em Pernambuco. Emprtel, Procenge, Chesf e CSI acharam as contribuições do trabalho relevantes como auxílio ao gerenciamento e demonstraram interesse em aplicar o Inspector nos seus projetos. O estudo de caso realizado na Emprtel serviu como um *feedback* para a organização avaliar a aplicação do Inspector nos seus projetos e, a partir dos resultados positivos obtidos na sua aplicação, ficou decidido que o mesmo será implantado dentro da metodologia de desenvolvimento [Emp00] que está em fase de implantação na organização.

Referências

- [AP98] Arnold M., Pedross P., Software Size Measurement and Productivity Rating in a Large-Scale Software Development Department, IEEE, 0-8186-8368-6/98, 1998.
- [BJR99a] Booch G., Jacobson I., Rumbaugh J., The Unified Software Development Process, Addison-Wesley, 1999.
- [BJR99b] Booch G., Jacobson I., Rumbaugh J., Unified Modeling Language – User’s Guide, Addison-Wesley, 1999.
- [BM92] Baumert J. H., McWhinney M. S., Software Measures and the Capability Maturity Model, Documento No. CMU/SEI-TR-25, Carnegie Mellon University, Software Engineering Institute, 1992.
- [CAF⁺98] Caldiera G., Antoniol G., Fiuten R., Lokan C., Definition and Experimental Evaluation of Function Points for Object-Oriented Systems, Proceedings of the 5th. International Symposium on Software Metrics, 1998.
- [Cha97] Champeaux D., Object-Oriented Development Process and Metrics, Prentice Hall, 1997.
- [Chu91] Chung L., Representation and Utilization of Non-Functional Requirements for Information System Design, Department of Computer Science, University of Toronto, CAISE’91, 1991.

- [CK94] Chidamber S.R, Kemerer C.F., A Metrics Suite for Object Oriented Design, IEEE Transactions on Software Engineering, vol.20, No. 6, Junho, 1994.
- [Emp00] Empresa Municipal de Informática – Recife, Metodologia Emprel de Desenvolvimento de Software, Documento Interno, 2000.
- [FR90] Fowler P., Rifkin S., Software Engineering Process Group Guide, Software Engineering Institute, CMU/SEI-90-TR-24, ADA235784, 1990.
- [Hen96] Henderson-Sellers B., Object Oriented Metrics: Measures of Complexity, Prentice Hall, 1996.
- [Hum90] Humphrey W. S., Managing the Software Process, Addison-Wesley, 1990.
- [JA97] Jacquet J., Abran A., From Software Metrics to Software Measurement Methods: A Process Model, IEEE, 1082-3670/97, 1997.
- [Joi98] Joint Group on System Engineering, Pratical Software Measurement – A Foundation for Objective Project Management, Version 3.1a, 1998.
- [KSW99] Kirsopp C., Shepperd M., Webster S., An Empirical study into the use of measurement to support OO design evaluation, Empirical Software Engineering Research Group, ESERG TR99-01, 1999.
- [LK94] Lorenz. M, Kidd J., Object-Oriented Software Metrics, Prentice Hall, 1994.
- [LY98] Lin J., Yeh C., An Object-Oriented Formal Model for Software Project Management, Proceedings of the Asia Pacific Software Engineering Conference, 1998.
- [MM00] Meneses J. B., Moura H. P., Um Framework de Avaliação de Progresso de Projetos de Software Orientado a Objetos, Workshop de Qualidade de Software (WQS2000), João Pessoa-PB, outubro 2000.
- [MM01a] Meneses J. B., Moura H. P., Inspector: Um Processo de Avaliação de Progresso para Projetos de Software, Centro de Informática, Universidade Federal de Pernambuco, Dissertação de Mestrado, 2001.
- [MM01b] Meneses J. B., Moura H. P., Inspector – Processo de Avaliação de Progresso de Projetos de Software Orientado a Objetos, disponível em <http://www.cin.ufpe.br/~inspector>, 2000. Último acesso em 04/2001.
- [Par92] Park E. R., Software Size Measurement: A Framework for Counting Source Statements, Documento No. CMU/SEI-TR-20, Carnegie Mellon University, Software Engineering Institute, 1992.
- [Pre99] Pressman S. R., Software Engineering: A Practitioner's Approach, McGraw-Hill, 5ª ed., 1999.
- [Qua99] Quatrani T., Visual Modeling with Rational Rose 2000 and UML, Addison-Wesley, 2 ed., outubro, 1999.
- [Que99] Queiroz E., Ambiente de Mensuração de Software Orientado a Objetos, Departamento de Ciências da Computação, Universidade de Brasília, Dissertação de Mestrado, 1999.
- [Sch99] Schroeder M., A Practical Guide to Object-Oriented Metrics, IT Professional - IEEE, Vol.1, No.6, novembro/dezembro, 1999.
- [TSF⁺99] Travassos G. H., Shull F., Fredericks M., Basili V. R., Detecting Defects in Object Oriented Designs: Using Reading Techniques to increase Software Quality, ACM Sigplan Notices, USA, v.34, n.10, p.47-56, ISSN 0362-1340, 1999.
- [Wie99] Wiegers K., A Software Metrics Primer, Software Development Magazine, julho, 1999.