

A Fábrica de Experiência do Tribunal de Contas do Distrito Federal: Primeiros Resultados

Clério B. Brasil - clerio@tc.df.gov.br
Flávio José F. de Souza – flavio@tc.df.gov.br
Walcelio L. Melo – walcelio.melo@oracle.com

RESUMO

Neste artigo, nós apresentaremos a Fábrica de Experiência do Tribunal de Contas do Distrito Federal (TCDF). Uma Fábrica de Experiências é uma estrutura física e/ou lógica de apoio o desenvolvimento de sistemas por meio da análise e sintetização de experiências, agindo como um repositório para essas experiências e fornecendo-as a outros projetos na medida da demanda.

Nós mostraremos o processo padronizado de desenvolvimento de sistemas idealizado no âmbito desta fábrica. Como iremos ver, tal processo é sujeito a mensuração. O programa de mensuração foi realizado segundo as diretrizes da metodologia *Goal, Question, Metric - GQM*. A adoção desse processo permite acompanhar a qualidade dos produtos obtidos e facilitar a troca de experiências entre os membros da equipe, nos moldes preconizados no *Quality Improvement Paradigm - QIP/SEL/NASA*, com um *overhead* tolerável de serviço.

Tribunal de Contas do Distrito Federal's Experience Factory: First Results

ABSTRACT

In this paper, we present Tribunal de Contas do Distrito Federal (TCDF)'s Experience Factory. An Experience Factory is a logical and/or physical organization aimed at supporting software development organizations. The Experience Factory is responsible for packaging and storing experiences originated in such organizations as well as providing under demand experience packages to other project organizations.

We also show the measurable-based software development process built up under the context of TCDF's Experience Factory. Such process was conceived according to the guidelines provided by the Goal-Question-Metric Method. The use of such process allows us to follow up the software product quality and facilitates the exchange of experience among software developers, as described by the Quality Improvement Paradigm. This was done with a very low overhead.

A FÁBRICA DE EXPERIÊNCIA DO TRIBUNAL DE CONTAS DO DISTRITO FEDERAL: PRIMEIROS RESULTADOS

1 INTRODUÇÃO

Neste artigo, nós apresentamos a Fábrica de Experiência do Tribunal de Contas do Distrito Federal (TCDF). Uma Fábrica de Experiências é uma estrutura física e/ou lógica de apoio ao desenvolvimento de sistemas por meio da análise e sintetização de todos os tipos de experiências, agindo como um repositório para essas experiências e fornecendo-as a outros projetos na medida da demanda.

Neste trabalho, nós apresentaremos os pontos motivadores para a criação, implantação e uso de uma Fábrica de Experiência no TCDF. Nós mostraremos o processo padronizado de desenvolvimento de sistemas idealizado no âmbito desta fábrica, aplicando a metodologia *Goal, Question, Metric - GQM* para identificação de objetivos e medidas. A adoção desse processo permite acompanhar a qualidade dos produtos obtidos e facilitar a troca de experiências entre os membros da equipe, nos moldes preconizados no *Quality Improvement Paradigm - QIP/SEL/NASA*, com um *overhead* tolerável de serviço.

Este trabalho inicia-se com a apresentação dos conceitos utilizados na elaboração do processo de desenvolvimento de software. Dentre as diversas metodologias existentes, elegeram-se alguns aspectos daquelas que mais se adequavam à realidade do TCDF - para, em conjunto, formarem um processo de trabalho e avaliação de resultados apropriado e consistente.

Em seguida, apresenta-se brevemente o contexto em que foi desenvolvido o processo de desenvolvimento de software e o conjunto de fatores que motivou esse desenvolvimento.

Para que o processo de desenvolvimento de sistemas fosse coerente com as necessidades do TCDF, identificou-se um conjunto de objetivos e questões a que o processo deve satisfazer. A mensuração da qualidade do processo é feita pelo acompanhamento de um conjunto de medidas adequado aos objetivos e questões identificados.

São detalhados os quatro produtos básicos de um processo padronizado, a saber: documentos que disciplinam o processo; os padrões que devem ser utilizados; formulários para registro das atividades; e um sistema de informação que permita a captura de informações sobre essas atividades.

Ao final, são apresentados os resultados obtidos com a utilização desse processo pela equipe de desenvolvimento de sistemas do TCDF por meio da comparação das métricas face as questões e objetivos definidos.

2 REFERENCIAL TEÓRICO

Este capítulo objetiva apresentar os conceitos utilizados ou considerados na definição da Fábrica de Experiência e do processo produtivo de software para o TCDF. Basicamente, aplicaram-se ao processo de desenvolvimento de sistemas no TCDF a abordagem GQM, aliada às idéias de melhoria do QIP, com o objetivo de se atingir, pelo menos, o nível 3 do CMM para software do SEI.

2.1 Capability Maturity Model (CMM)

2.1.1 Definição

O *Capability Maturity Model (CMM)* para Software [14], compilado pelo *Software Engineering Institute – SEI* da *Carnegie Mellon University (EUA)*, pode ser entendido como um modelo para avaliação do grau de maturidade das organizações quanto à capacidade produtiva de software.

O CMM representa uma proposição de recomendações para empresas, cujo negócio seja a produção de software, que pretendam incrementar a capacidade (ou qualidade) do processo de desenvolvimento de sistemas.

O modelo apresentado pelo CMM não se preocupa em “como” a organização implementa, mas, antes, busca descrever “o quê” se espera do processo produtivo de software.

O CMM classifica as organizações produtoras de software em cinco grupos (ou níveis) de maturidade (a saber, inicial; repetitivo; definido; gerenciado; e otimizado), identificando, para cada grupo, as áreas chave do processo de desenvolvimento de sistemas que devem ser observadas na busca da melhoria da capacidade produtiva. Cada área chave possui objetivos que precisam ser alcançados para que a organização seja considerada como tendo atingido determinado nível.

2.1.2 Uso do CMM no TCDF

Tendo em vista que o CMM apresenta-se como um modelo externo à corporação, pode parecer, a princípio, uma ambigüidade pretender utilizar-se desse modelo e, ao mesmo tempo, seguir as recomendações preconizadas no QIP (adiante discutido), onde se busca principalmente o crescimento contínuo da empresa a partir de seus próprios recursos, modelos, objetivos e estratégias.

Porém, a aplicabilidade do modelo ao TCDF vem no sentido, tão somente, de obter-se um referencial externo consagrado que possa servir de calibrador nas intenções de perseguição de melhoria, ainda que apenas inicialmente ou parcialmente aludido.

Convém lembrar ainda que a certificação CMM depende de contratação de consultoria e avaliação externas, o que, definitivamente, por questões institucionais e até orçamentárias, foge ao objetivo do TCDF. Além disso, o CMM aplica-se, por concepção, a empresas cujo negócio seja a produção de software. A sua utilização fora desse escopo pretende acompanhar a concepção de Donna L. Johnson e Judith G. Brodman [10], quando apresentam uma adaptação do modelo para pequenas empresas, pequenas organizações e pequenos projetos.

Finalmente, um último ponto a salientar seria a diferença conceitual entre a forma escalar de definição do CMM e a maneira cíclica de aperfeiçoamento do QIP.

Por essas razões, a adoção do CMM pelo setor de informática do TCDF não seguiu rigorosamente, nem poderia fazê-lo, as diretrizes conceituais do modelo.

2.2 Quality Improvement Paradigm (QIP)

2.2.1 Definição

O *Quality Improvement Paradigm* – QIP (ou Paradigma do Aperfeiçoamento da Qualidade), desenvolvido por Victor Basili e outros [1], é o resultado da aplicação do método científico para resolução do problema de melhoria da qualidade de software.

O QIP desenvolve-se por meio dos seguintes passos:

1. Caracterizar: conhecer o ambiente com base nos dados e modelos disponíveis e na intuição. Estabelecer os fundamentos com os processos existentes na organização e caracterizá-los criticamente.
2. Definir os objetivos: com base na caracterização inicial e nas capacidades estratégicas da organização, definir com objetivos quantificáveis o que seriam projetos bem sucedidos e avaliar o desempenho e melhoria da organização. As exceções aceitáveis são definidas com base nos fundamentos estabelecidos no passo de caracterização.
3. Escolher os processos: com base na caracterização do ambiente e dos objetivos definidos, escolher os processos apropriados para melhoria bem como as ferramentas e métodos de apoio, certificando-se que esses sejam consistentes com os objetivos.

4. Executar: executar os processos elaborando os produtos e provendo retorno, a partir do projeto, dos dados que estão sendo coletados para avaliação do alcance dos objetivos.
5. Analisar: ao final de cada projeto específico, analisar os dados e informações acumuladas para avaliar as práticas correntes, identificar problemas, registrar achados, e fazer recomendações para melhoria de futuros projetos.
6. Empacotar: Consolidar a experiência adquirida em forma de refinamento do modelo praticado ou mesmo pela adoção de novos modelos e, ainda, por meio de outras formas de estrutura de conhecimento alcançadas no último ou em processos anteriores e armazená-las em uma base de experiências, disponível para uso futuro.

Uma caracterização apropriada e sem ambigüidades do ambiente é um pré-requisito para a aplicação correta do paradigma. Essa caracterização requer a classificação minuciosa do projeto, permitindo a identificação de classes de projetos com características e objetivos similares. Assim, pode-se estabelecer o ambiente adequado ao projeto corrente. Com a correta caracterização obtém-se um contexto para a definição de objetivos, para a reutilização de experiências e produtos, para a seleção de processos, para a avaliação e comparação de resultados, e para uma maior exatidão nas previsões.

Há um grande número de características de projetos e de ambiente que afetam o processo de desenvolvimento e o produto de software, tais como:

- fatores de pessoal - números de empregados, nível de especialização, organização de grupo, experiências;
- fatores relacionados ao problema - domínio da aplicação, suscetibilidade a mudança, limites do problema;
- fatores do processo - modelos de processo, métodos, técnicas, ferramentas, linguagem de programação, outros documentos;
- fatores do produto - implantação/entrega, tamanho do sistema, qualidades requeridas;
- fatores dos recursos - máquinas de produção e desenvolvimento, tempo calendário, orçamento, software existente.

A definição realista dos objetivos é importante para a caracterização do ambiente. É preciso estabelecer modelos e objetivos para os processos e para os produtos. Esses objetivos precisam ser mensuráveis, dirigidos pelos modelos, e definidos para uma variedade de perspectivas, tais como, a do usuário, do cliente, do projeto, da organização, etc.

A Abordagem *Goal/Question/Metric* (demonstrada adiante) é o mecanismo usado pelo *Quality Improvement Paradigm* para definir e avaliar um conjunto de objetivos operacionais usando métricas. Essa abordagem representa uma sistemática para ajuste e integração de objetivos com modelos de processos, produtos e perspectivas de qualidade de software, baseadas em necessidades específicas do projeto e da organização.

2.2.2 A Fábrica de Experiência

O QIP baseia-se na idéia de que o aperfeiçoamento do processo e do produto de software requer uma acumulação contínua de experiências (aprendizado) em formato que pode ser efetivamente entendido e modificado (modelos de experiências) em um repositório integrado (base de experiências) que pode ser acessado e modificado em conformidade com as necessidades do projeto corrente (reutilização). Para apoiar o paradigma de aperfeiçoamento foi desenvolvida um estrutura distinta do desenvolvimento de software denominada Fábrica de Experiências (*Experience Factory* – EF).

A Fábrica de Experiências é uma estrutura física e/ou lógica que apóia o desenvolvimento de sistemas por meio da análise e sintetização de todos os tipos de experiências, agindo como um repositório para essas experiências e fornecendo-as a outros projetos na medida da demanda.

A Fábrica de Experiências realmente representa um paradigma de mudança do pensamento corrente de desenvolvimento de software. Ela separa os tipos de atividades que precisam ser executadas em diferentes estruturas, reconhecendo que elas verdadeiramente representam processos e enfoques diferentes.

Assim, identificam-se duas unidades organizacionais distintas: a Organização Projeto, para o desenvolvimento de sistemas; e a Fábrica de Experiências, para o empacotamento de conhecimentos adquiridos. Na Organização Projeto, são resolvidos os problemas, na Fábrica de Experiência, são entendidas as soluções e empacotadas as experiências para futura reutilização.

A Fábrica de Experiências representa o grupo que trabalha na base do conhecimento da engenharia de desenvolvimento de software como um bem da corporação, enquanto a Organização Projeto representa o grupo cujo trabalho é usar aquele conhecimento para produzir os mais avançados produtos da corporação.

A Organização Projeto *caracteriza* o projeto e seu ambiente, *define os objetivos* para o projeto e *escolhe o processo* apropriado dadas as características e objetivos. O *suporte ao projeto* é provido pela Fábrica de Experiências pela articulação das características, formulação de objetivos e seleção do conjunto apropriado de experiências passadas para uso no projeto. Um plano específico é estabelecido para o projeto e ajustado se necessário. Durante a execução do projeto, este fornece dados, modelos e lições aprendidas para a Fábrica de Experiências para ajudá-la a prover retorno em tempo real ao projeto. Esse retorno é *analisado* pela Fábrica de Experiências para determinar que partes das experiências podem ser úteis em futuros projetos. Aquelas partes úteis são então *empacotadas* para reutilização por meio da generalização, ajustamento, formalização e armazenamento para favorecer o uso.

Pela abordagem da Fábrica de Experiências, a reutilização não representa apenas o reaproveitamento de código mas de todos os tipos de experiências e do contexto que envolve essas experiências. A Fábrica de Experiências reconhece que as experiências nem sempre são reutilizáveis como foram concebidas, antes elas precisam ser ajustadas e empacotadas para tornar mais fácil o seu acesso e reutilização.

Na prática, a abordagem QIP/EF é implementada colocando-se primeiramente a organização em seu lugar, a partir de sua caracterização. Isso significa a coleta de dados para estabelecimento dos fundamentos (*baselines*) e avaliação dos pontos fortes e fracos da organização a fim de se obter o enfoque nos negócios e objetivos da empresa para o processo de melhoria. A coleta inicial de dados é crítica também para a definição da qualidade do produto que deveria ser aprimorada pelo programa. Com o uso dessas informações, a organização seleciona e experimenta os métodos e técnicas para a melhoria dos processos e produtos. Processos melhores e mensuráveis podem ser definidos e ajustados com base na experiência e nos conhecimentos adquiridos nos ambientes dos projetos. Os resultados serão sempre validados com relação à conformidade ao processo e ao entendimento do domínio.

Quando o relacionamento entre características dos projetos e as qualidades do produto dentro de um ambiente específico é compreendido, a organização está pronta para manipular seus processos e para alcançar as características pretendidas para o produto. Mudanças nos processos estabelecem novos fundamentos bem como novos objetivos de aperfeiçoamento.

Dessa forma, a organização define em si mesma um contínuo aperfeiçoamento, mesmo se o nível de maturidade não for muito alto, porque ela aprende a partir do seu próprio negócio, não de um modelo de processo ideal, externamente concebido. Melhorias no processo serão baseadas na compreensão do relacionamento entre processo e produto na própria organização. A introdução de

novas tecnologias será motivada por problemas locais e o corpo técnico estará mais disposto a utilizá-las.

2.2.3 Implementação QIP/EF no TCDF

A aplicação de conceitos elaborados para grandes corporações voltadas exclusivamente à produção de software é extremamente difícil para entidades de pequeno porte e, em especial, àquelas que são apenas unidades auxiliares em instituições cujo objetivo não é o desenvolvimento de sistemas. Por isso, aqui também, faz-se necessária a colocação de algumas ponderações com respeito à forma de implementação do um modelo QIP/EF para o Tribunal de Contas do DF.

Não é possível a um pequeno *bureau* que atua como unidade de apoio, manter uma estrutura, fisicamente separada da linha de produção, para implementar um Fábrica de Experiências. Dessa forma, foram aproveitados os conceitos básicos possíveis dentro da realidade do Tribunal (inclusive, esta é uma premissa do QIP) e eleito um sistema para ser usado como teste na aplicação de uma nova tecnologia.

Embora essa não seja uma aplicação pura da idéia de Fábrica de Experiência, pode-se entender que o escopo de um projeto seja a estrutura lógica dessa implementação. Nessa experiência, é caracterizado o ambiente, são definidos os objetivos, executados os procedimentos necessários e avaliados os resultados.

Com a finalidade de promover a integração entre objetivos e modelos, a implementação do processo utilizou a Abordagem *Goal/Question/Metric*, discutida no próximo tópico - onde serão também apresentadas considerações sobre a definição do modelo, dos objetivos e das métricas a serem capturadas.

Finalmente, resta deixar claro que a mais importante idéia a extrair-se da intenção de se implantar os conceitos do QIP no TCDF seria o compromisso com o constante aperfeiçoamento da qualidade dos produtos, a partir da formalização e melhoria contínua do processo produtivo.

2.3 Goal Question Metric (GQM)

Esta apresentação da Abordagem *Goal/Question/Metric* (GQM), utilizada como metodologia na definição do processo de desenvolvimento de software no TCDF, toma por base o texto de Victor R. Basili, Gianluigi Caldiera e H. Dieter Rombach [2].

2.3.1 A necessidade de um processo de medida

Como em qualquer disciplina de engenharia, o desenvolvimento de software requer um mecanismo de mensuração para avaliação e retorno. O procedimento de medir é uma forma de criar uma memória corporativa e um auxílio na resposta de várias questões associadas ao estabelecimento de um processo de software.

Um processo de medidas estabelecido auxilia no planejamento de novos projetos, na determinação de pontos fortes e fracos de produtos e processos, na racionalidade para adoção ou refinamento de técnicas em uso, e ainda, na avaliação da qualidade de processos ou produtos específicos. Medir também ajuda, durante o curso de um projeto, a avaliar o seu progresso, tomar as medidas corretivas baseadas nesse julgamento, e avaliar o impacto de tal ação.

A abordagem GQM parte da premissa de que para uma organização adotar um processo de medida definitivo é necessário primeiramente estabelecer os objetivos da própria organização e de seus projetos, definir esses objetivos operacionalmente e, finalmente, criar um ambiente de apoio capaz de interpretar os dados comparando-os com os objetivos estabelecidos. Assim, é importante deixar claro, pelo menos em termos gerais, qual a necessidade de informações da organização, se essas informações podem ser quantificadas e em que momento podem ser colhidas, e se podem ser analisadas em função dos objetivos.

A abordagem foi definida originalmente para avaliação de defeitos em um conjunto de projetos no ambiente da *NASA Goddard Space Flight Center*. A aplicação envolveu um conjunto de estudos de casos e foi expandida para incluir várias abordagens experimentais. Embora a abordagem tenha sido usada originalmente para definir e avaliar os objetivos de um projeto específico, seu uso foi expandido para contextos maiores.

A abordagem foi usada como um passo dentro de outra técnica, o *Quality Improvement Paradigm* (discutido anteriormente neste trabalho), processo de melhoria da qualidade que se utiliza de um ambiente operacional restrito, a Fábrica de Experiência (*Experience Factory*), que pode, por sua vez, ser entendida como um ambiente experimental com o objetivo de produção e/ou avaliação de software e processos para uso nos demais projetos.

O resultado da aplicação da abordagem GQM é a especificação de um sistema de medidas objetivando um conjunto particular de casos e de regras na interpretação dos dados medidos.

Um modelo GQM é uma estrutura hierárquica que inicia com a definição de um objetivo (*goal*), especificando o propósito da medição, os objetos e aspectos desses objetos que serão avaliados, e o ponto de vista em que as medidas serão analisadas. O objetivo é, então, refinado em diversas questões (*question*). Cada questão é, por sua vez, delineada nas métricas (*metric*).

Há que se observar que uma mesma métrica pode ser usada para responder diferentes questões de um mesmo objetivo; diversos modelos GQM podem, também, ter questões e métricas em comum, desde que seja assegurado que quando da coleta das métricas sejam observados os diversos pontos de vista a que se destinam, pois a mesma medida pode assumir valores diversos dependendo do ponto de vista em que será analisada.

2.3.2 O processo GQM

Um modelo GQM é desenvolvido a partir de um conjunto de objetivos acerca de qualidade e/ou produtividade definidos para a organização, para a divisão ou para o projeto, tais como satisfação do usuário, entrega de serviço no prazo, melhoria de desempenho.

A partir da identificação dos objetivos e com base em modelos do objeto em avaliação, derivam-se questões que possam definir esses objetivos de forma mais completa. Por exemplo, se o objetivo é caracterizar um software quanto a determinadas qualidades (e.g., portabilidade), então faz-se necessária a escolha de um modelo para o produto que qualifique esses interesses (e.g., relação de características funcionais que precisam ser implementadas para diferentes arquiteturas).

O próximo passo consiste em especificar as medidas que devem ser coletadas a fim de responder às questões e acompanhar a conformidade dos produtos e processos aos objetivos.

Por fim, há que se desenvolver os meios de coleta dos dados, incluindo-se mecanismos de avaliação e análise.

O processo de identificação de objetivos é crítico para o sucesso da aplicação da Abordagem GQM, e será apoiado por passos metodológicos específicos. Para a consecução de um objetivo concorrerão três fontes básicas de informação.

A primeira fonte diz respeito à política e à estratégia da organização que aplica a Abordagem GQM. A partir dessa fonte, com a análise da política da corporação, dos planos estratégicos e, ainda, levando-se em conta os interesses relevantes na organização, derivam-se o “aspecto” e o “propósito” para o objetivo a ser perseguido.

A segunda fonte de informações é a descrição dos processos e produtos da organização, ou, pelo menos, daqueles que estão dentro do escopo do programa de medidas que se pretende efetuar. A partir desse fonte, com a especificação dos modelos de processos e produtos, dentro da maior formalidade possível, deriva-se a coordenada do “objeto” para o objetivo em questão.

A terceira fonte de informações é o modelo do negócio da organização, que fornece a coordenada “ponto de vista”. É evidente que nem todos os assuntos e processos são relevantes para todos os pontos de vista na organização. Assim, há que se fazer uma análise da relevância dos objetivos para a organização, antes de se considerar concluída a lista de objetivos.

Dessa forma, conclui-se a definição dos “objetivos” para o modelo GQM, tomando-se em conta a estrutura e os objetivos da organização. A partir da especificação de cada objetivo podem-se derivar questões significantes que quantificam o objetivo. Geralmente, são feitos pelo menos três grupos de questões:

Grupo 1. Como se pode caracterizar o objeto (produto, processo ou recurso) com o respeito ao objetivo global de determinado modelo GQM?

Exemplo:

Questão: Qual o prazo corrente no atendimento às solicitações de mudanças num sistema em manutenção?

Questão: Existe um processo (documentado) no atendimento às solicitações de mudanças?

Grupo 2. Como se podem caracterizar os atributos relevantes do objeto em observação num modelo GQM específico?

Exemplo:

Questão: Qual o desvio do prazo no atendimento de solicitações em função do estimado?

Questão: O desempenho da equipe no atendimento de solicitações vem melhorando?

Grupo 3. Como podem ser avaliadas as características do objeto que são relevantes com respeito ao aspecto tratado no modelo GQM?

Exemplo:

Questão: O desempenho tem sido satisfatório sob o ponto de vista do gerente de projeto?

Questão: Há melhoria visível no desempenho?

Uma vez que as questões tenham sido formuladas, deve-se proceder a associação destas com as métricas apropriadas. Diversos fatores devem ser considerados, dentre eles:

- Volume e quantidade dos dados existentes - deve-se buscar maximizar o uso das fontes de dados existentes, desde que estejam disponíveis e sejam confiáveis.
- Maturidade dos objetos em medição - devem-se aplicar medidas objetivas para objetos com maior nível de maturidade, e avaliações subjetivas quando se trabalha com objetos instáveis ou informais.
- Aprendizado do processo - o uso de modelos QGM requer sempre refinamento e adaptação. Assim, as medidas definidas devem auxiliar não somente na análise do objeto medido, mas também na avaliação do próprio modelo.

Uma vez que o modelo GQM esteja definido, faz-se necessária a seleção das técnicas, ferramentas e procedimentos apropriados à coleta de dados. Os dados coletados devem ser mapeados para o modelo e interpretados de acordo com esquemas previamente definidos pela organização.

Conclui-se, então, que a Abordagem *Goal/Question/Metric* é uma metodologia para definição de um mecanismo de mensuração que possibilite o acompanhamento e avaliação do processo operacional de produção de software.

2.3.3 Uso do GQM no TCDF

Nós apresentaremos na próxima seção um extrato modelo GQM para o desenvolvimento de sistemas no Tribunal de Contas do DF. O modelo completo pode ser obtido em [5].

Na elaboração desse modelo foram considerados os objetivos do setor de informática, bem como o planejamento estratégico do Tribunal e sua missão institucional, delineados no item 3.2. *O papel da informática no TCDF* deste trabalho, dentro do contexto de aplicação do processo.

Os objetivos operacionais podem ser inferidos a partir das razões apresentadas no item 3.3. *Motivação para o estabelecimento do processo de desenvolvimento*, onde demonstram-se os fatores básicos que nortearam a criação de um processo de desenvolvimento bem assim as intenções pretendidas no seu estabelecimento.

Em nível estratégico, buscou-se definir os objetos a serem mensurados dentre aqueles, cujas medidas poderiam ser mais facilmente captadas e que melhor responderiam às necessidades de verificação. Assim, foram definidos objetos, questões e medidas para processos (tempo de análise, codificação, teste, etc.), e para produtos (LOC, número de arquivos, etc.) de maneira objetiva (i.e., não serão coletadas medidas subjetivas).

O ambiente de apoio à coleta de dados é definido no item 3.5 do trabalho, onde são apresentados os sistemas que auxiliam no registro de solicitações do usuário final e no acompanhamento do processo de desenvolvimento.

Finalmente, na seção 4, estão reportadas as primeiras observações verificadas com a implantação do processo de medidas baseado na Abordagem GQM.

2.4 AINSI

Para a elaboração de um programa de controle de qualidade que atue na "linha de produção" de sistemas de *software*, consideramos bem apropriada a abordagem AINSI [6]. Através deste método, os autores sugerem uma alternativa à utilização dos modelos de avaliação baseados em modelos de referência, como: CMM, TRILLIUM, BOOTSTRAP e SPICE; todos estes, mais direcionados às pessoas que não sabem ao certo como, nem por onde, iniciar uma programa de melhoria de *software*. Além disto, estes modelos podem não atacar diretamente o problema pelo qual está exposto o processo de desenvolvimento de *software* de uma determinada organização. Alternativamente a estas soluções de atacado, eles propõem uma abordagem de varejo, onde, consideram relevantes as particularidades do processo de *software* específico que se pretende melhorar, buscando as verdadeiras causas do problema, para em seguida sugerir uma solução customizada.

A AINSI foi inspirada no *Quality Improvement Paradigm - QIP* [1], objeto de estudo de vários pesquisadores. É importante ressaltar que o processo para inserir a AINSI, do seu início até às mudanças organizacionais necessárias, é laborioso e consumidor de tempo e recursos, principalmente o humano. Sendo assim, mostraremos aqui, o estágio atual do nosso trabalho.

2.4.1 ANSI no TCDF

ANSI propõe um conjunto de diretrizes visando melhor gerenciar o processo de implantação de um programa de qualidade baseado nos conceitos do QIP/EF/GQM. Mostraremos aqui tais diretrizes e como elas foram aplicadas para construir a Fábrica de Experiência do TCDF.

2.4.2 - Montar uma Equipe de Melhoria do Processo:

No ambiente para implementação do Modelo GQM, o primeiro passo constitui-se na montagem da equipe para levantamento e melhoria dos processos. No caso do TCDF, dado o reduzido número de empregados do setor de informática, o levantamento e definição do processo de desenvolvimento de sistema foi realizado por dois analistas, num total de **80 (oitenta) horas**.

2.4.3 - Modelar os Processos Existentes:

No TCDF, o processo de desenvolvimento de sistemas foi modelado e descrito de forma a explicar claramente suas atividades intrínsecas, as dependências entre as atividades, o tempo de duração de cada uma delas, os atores que as executam e as interdependências entre estes atores. Devido a problemas de espaço, nós não mostraremos aqui tal processo. Contudo, a documentação completa sobre tal processo pode ser encontrada em [5].

No apoio ao processo de desenvolvimento de sistemas concorrem dois sistemas informatizados: o Sistema de Registro e Acompanhamento de Ocorrências, que permitem ao usuário final registrar erros, falhas, dúvidas ou sugestões de melhorias nos demais sistemas de computador em operação na Casa; e o Sistema de Acompanhamento do Desenvolvimento de Sistemas, responsável pelo registro e acompanhamento das métricas do processo de desenvolvimento.

2.4.4 - Conduzir uma Análise Qualitativa:

No TCDF, como os processos já haviam sido modelados e bem definidos, nesta fase foram elaborados também os procedimentos de coleta de dados e desenhados os formulários. Foi selecionado o Sistema de Protocolo e Acompanhamento Processual para uma análise preliminar da metodologia a ser empregada. Foram levantados os problemas apresentados na versão do sistema e classificadas as falhas segundo critérios de gravidade estabelecidos pelo gerente de informática.

2.4.5 - Definir e Documentar um Plano de Ação:

Como Plano de Ação para o TCDF foi estabelecida uma metodologia que contempla: procedimentos de desenvolvimento e manutenção de sistemas, estabelecimento de um fluxo de solicitações de sistemas, o Sistema de Acompanhamento do Desenvolvimento de Sistemas – SADS, que permite o registro do processo e a extração das métricas estabelecidas (todos estes itens podem ser vistos em [5]).

2.4.6 - Montar um Programa de Medição:

Neste passo da metodologia foram definidas as metas corporativas e de medidas. Foram aplicados os procedimentos de coleta de dados e implantadas as rotinas de utilização dos formulários desenhados anteriormente, sendo orientados os funcionários responsáveis pela captação e alimentação dos dados. Este passo não apresentou problemas, pois, a coleta dos dados foi voltada para as metas estabelecidas, que ficaram claramente entendidas pela equipe de desenvolvimento. Não houve uma sobrecarga no trabalho dos desenvolvedores, uma vez que a coleta dos dados não requer mais de 20 minutos diários, numa jornada de 8 horas. Nós acreditamos que o sucesso da implementação deveu-se a integração e engajamento da equipe de desenvolvedores na coleta dos dados.

2.4.7 - Mudar os Processos e a Organização:

Neste capítulo os autores apresentaram uma abordagem ascendente para a melhoria prática do processo de produção de software e seus produtos. Estes passos devem ser vistos como uma implantação do Paradigma de Aperfeiçoamento da Qualidade – *QIP*. A fim de facilitar sua aplicação em diferentes organizações de desenvolvimento e manutenção de software, os autores montaram um conjunto de passos e diretrizes deste paradigma, baseados em suas experiências adquiridas, conduzindo estudos quantitativos e qualitativos, em diversas organizações públicas e privadas.

3 CONTEXTO DE APLICAÇÃO DO PROCESSO DESENVOLVIDO

Aqui serão apresentadas a missão institucional do TCDF e, inserida nessa missão, o papel desenvolvido pelo setor de informática.

3.1 A missão do TCDF

As competências constitucionais do Tribunal de Contas do Distrito Federal são:

- apreciar, mediante emissão de parecer prévio, as contas anuais do Governador e julgar aquelas relativas aos administradores e demais responsáveis por dinheiro, bens e valores públicos;
- apreciar, para fins de registro, a legalidade dos atos de admissão de pessoal e de concessão de aposentadorias, reformas e pensões;
- avaliar a execução das metas estabelecidas no plano plurianual, nas diretrizes orçamentárias e no orçamento anual;
- realizar inspeções e auditorias de natureza contábil, financeira, orçamentária, operacional e patrimonial nas unidades administrativas dos Poderes Executivo e Legislativo;
- fiscalizar as aplicações do Poder Público em empresas de cujo capital social o Distrito Federal participe de forma direta ou indireta;
- fiscalizar a aplicação de recursos repassados ou recebidos pelo Distrito Federal, a qualquer título;
- atender às solicitações da Câmara Legislativa relativas às atividades de Controle Externo;
- aplicar, em caso de ilegalidade ou irregularidade de contas, as sanções previstas em lei e sustar, se o Tribunal não for atendido, a execução de ato impugnado.

No planejamento estratégico para o período de 1999 a 2003 [8] é dado especial relevo ao estabelecimento da missão institucional do TCDF com base nas competências constitucionais acima e nos arts. 77 e 78 da Lei Orgânica do Distrito Federal (LODF) [3]. Esses artigos elucidam que a fiscalização contábil, financeira, orçamentária, operacional e patrimonial dos órgãos e entidades da administração do Distrito Federal, quanto à legalidade, legitimidade, economicidade, aplicação das subvenções e renúncia de receitas é exercida pela Câmara Legislativa - mediante controle externo e pelo sistema de controle interno de cada Poder - com o auxílio do Tribunal de Contas do Distrito Federal.

Assim, inferida da essência dos citados dispositivos legais, a missão do Tribunal pode ser enunciada como:

“Exercer o controle externo da administração dos recursos públicos do Distrito Federal, em auxílio à Câmara Legislativa, zelando pela legalidade, legitimidade, efetividade, eficácia, eficiência e economicidade na gestão desses recursos.”

Releva notar que o atributo principal do papel da Corte de Contas Distrital é garantir à sociedade segurança quanto à transparência e boa gestão dos recursos arrecadados e gastos pelo Governo do Distrito Federal.

3.2 O papel da informática no TCDF

A política de informatização do Tribunal tem se pautado pela adoção de soluções que associem qualidade, funcionalidade e economicidade.

Nesse contexto, ao setor de informática do Tribunal, denominado Núcleo de Informática e Processamento de Dados – NIPD, compete apoiar a atividade de fiscalização por meio da criação de sistemas de informação para acompanhamento dos gastos públicos, avaliação de riscos e identificação de pontos críticos para auditoria, produção, organização e divulgação de súmulas, decisões, votos, pareceres e instruções. Para auxiliar essas atividades foram desenvolvidos, em especial a partir de 1996, vários sistemas integrados em repositório de informações único.

Ao NIPD compete também administrar a rede de microcomputadores, executar e acompanhar a manutenção dos equipamentos de informática, bem como propor a compra de bens e serviços relativos a tecnologia.

Adicionalmente, o setor mantém atualizado, com o apoio de diversas Unidades do TCDF, o *site* www.tc.df.gov.br que permite a qualquer cidadão consultar os documentos públicos produzidos no exercício do controle externo.

Por ser um ente público com especial apreço pela boa gestão dos recursos, antes de efetuar investimentos são feitas análises de alternativas com vistas a encontrar a melhor relação custo x benefício. Essa prática, aplicada ao setor de informática será, como apresentado a seguir, motivadora do desenvolvimento de um processo de produção de software.

3.3 Motivação para o estabelecimento do processo de desenvolvimento

Devido ao esforço necessário ao estabelecimento de um processo de desenvolvimento, em especial quando se conta com poucos recursos humanos e materiais, foi fundamental para o empenho na produção do mesmo a existência das motivações a seguir relacionadas.

3.3.1 Melhoria da qualidade de produtos

Sendo o TCDF um órgão de informatização relativamente recente, a partir de 1995, a equipe responsável pelo desenvolvimento de sistemas teve a oportunidade de construir todo um conjunto de aplicações corporativas, com base nas necessidades identificadas junto aos usuários, integrado e homogêneo. No entanto, o processo de construção dessas soluções foi fortemente influenciado pela experiência profissional dos técnicos e, apesar de ter sido um processo repetitivo, não era explicitado formalmente.

Hoje, contando com mais de dez aplicações em produção (150.000 linhas de código fonte em Visual Basic, VB Script, Java, Java Script e HTML), que, portanto, requerem manutenção, e sem modificação no quadro de desenvolvedores (2 analistas e 4 programadores), faz-se necessário aprimorar o processo de trabalho visando a elaboração de produtos com menor número de erros possível, que observem a integração e a homogeneidade dos sistemas já desenvolvidos.

3.3.2 Planejamento de novas demandas

Da forma atual, as estimativas de prazo para desenvolvimento ou manutenção de sistemas são baseadas somente na experiência pessoal dos técnicos, sem um acompanhamento efetivo dos prazos.

Essa situação, apesar de bastante comum nas organizações, é fator de risco para a credibilidade do setor no caso de falha. Felizmente, desde 1995, os técnicos tem acertado em suas previsões à administração da Casa. No entanto, o acompanhamento sistemático do processo de desenvolvimento de software será fundamental para aperfeiçoar as estimativas de prazos e recursos de novas demandas de sistemas.

3.3.3 Mudança de plataforma computacional

Com a relativa estabilização da demanda por desenvolvimento de novos serviços e programas ao setor de informática do TCDF, a partir do final de 1998, a equipe técnica iniciou trabalho de busca de alternativas com vistas a reduzir a necessidade de investimentos em informática e identificou grupos de programas que, se substituídos, levariam a essa redução.

Esse trabalho subsidiou a Presidência do Tribunal a decidir substituir, até o final do ano de 2002, o sistema operacional Windows 95 pelo Linux nas estações de trabalho do TCDF, proporcionando elevada redução de investimentos com hardware e software [9].

Assim, com vistas a possibilitar essa substituição, será necessário converter todos os sistemas desenvolvidos e configurar aplicativos de escritório que possibilitem ao usuário final usufruir das mesmas funcionalidades hoje disponíveis.

Esse fato gerará intenso esforço de codificação que, sem acompanhamento formalizado por meio de um processo de desenvolvimento definido e utilizado por todos os técnicos, poderá inviabilizar a substituição da plataforma computacional proposta pelo fracasso na conversão dos sistemas.

Para que seja possível substituir gradualmente as aplicações, optou-se por utilizar a linguagem Java como padrão para a execução dessa tarefa. A linguagem de programação Java permite disponibilizar os sistemas tanto para estações Windows 95 como Linux e irá facilitar a reutilização de código.

3.4 Modelo GQM aplicado ao desenvolvimento de sistemas no TCDF

Considerando o contexto e as razões acima descritos estabeleceu-se um modelo *Goal, Question, Metric*, para a área de desenvolvimento de sistemas do TCDF. Devido a limitações de espaço, mostraremos aqui somente os objetivos e questões deixando de lado as métricas.. O modelo GQM completo e os formulários de coleta de dados que implementam tal modelo podem ser obtidos em [5].

3.4.1 Objetivo G1 - Melhoria da Qualidade dos Sistemas Desenvolvidos

Procura-se aqui ter condições de dimensionar a melhoria dos produtos entregues aos usuários finais, utilizando-se como indicador principal a quantidade de erros percebida pelos usuários. Para esse objetivo foram definidas duas questões:

Q1 - Qual a quantidade de erros encontrados pelos usuário? - procura dimensionar os vazamentos de erros não encontrados na produção do sistema, permitindo quantificar os transtornos causados aos usuários. Em resposta a essa questão foram extraídas as seguintes medidas:

Q2 - A qualidade dos produtos está melhorando? - visa avaliar a evolução da qualidade dos produtos entregues por meio da comparação das métricas da *Q1* acima descritas ao longo do tempo. Em resposta a essa questão foram extraídas as métricas abaixo:

3.4.2 Objetivo G2 - Avaliar o tempo de conversão de sistemas em Visual Basic para Java

Tendo em vista a mencionada estratégia de substituição do sistema operacional utilizado no TCDF, faz-se necessário reescrever todos os sistemas em produção na linguagem Java para que os mesmos possam continuar sendo utilizados. Esse objetivo visa aperfeiçoar as estimativas de conversão dos sistemas por meio da análise das seguintes questões:

Q3 - Qual o tamanho dos sistemas originais em Visual Basic? - essa questão é relevante para dimensionar a tarefa a ser executada.

Q4 - Qual o tamanho dos sistemas convertidos em Java? - assim, pode-se avaliar o impacto de substituição da linguagem em termos de tamanho (LOC) dos sistemas convertidos e acompanhar o progresso do processo de conversão.

Q5 - Qual o tempo para conversão dos sistemas? - questão principal para o objetivo pretendido pois informa o esforço realizado até o momento e permite estimar a tarefa restante.

3.4.3 Objetivo G3 - Avaliar o planejamento do processo de desenvolvimento de sistemas

O processo de desenvolvimento de sistemas implantado contempla o planejamento de prazos e recursos. No entanto, para a efetividade do planejamento é necessário avaliar o seu grau de precisão e efetuar as correções necessárias.

Q6 - Qual a diferença entre o esforço (tempo) estimado e o realizado? - questão que possibilita verificar a precisão do planejamento de recursos; no caso do TCDF, o principal recurso para desenvolvimento de sistemas é a hora de trabalho dos técnicos.

Q7 = Qual a diferença entre os prazos previstos e reais? - possibilita aferir a precisão das estimativas de datas de início e fim de cada fase do processo de desenvolvimento.

3.4.4 Objetivo G4 - Avaliar a qualidade do processo de desenvolvimento de sistemas

Com a implantação do processo de desenvolvimento de sistemas pretende-se reduzir o número de erros encontrados pelo usuário, ou seja, erros após a entrada do sistema em produção, estabelecendo-se pontos de teste no processo de desenvolvimento de sistemas com a execução das atividades de leitura de código e teste de integração e implantação. Assim, para avaliar a efetividade do processo em termos de identificação de erros e redução de vazamentos (erros encontrados pelos usuários) foram definidas as seguintes questões:

Q8 - Qual o número de erros identificados pela equipe de desenvolvimento durante a elaboração do sistema? - possibilita aferir a eficácia das atividades de teste implantadas com o processo de desenvolvimento de sistemas.

Q9 - Qual a relação entre o número de erros identificados pelo usuário e os erros identificados durante a elaboração do sistema? - possibilita a comparação entre as quantidades de erros encontrados pelos usuários dos sistemas (vazamentos) e de erros encontrados pela equipe de desenvolvimento de sistemas nas fases de leitura de código e testes de integração e implantação.

Q10 - Qual a origem (fonte) dos erros encontrados? - visa caracterizar a origem dos erros para auxiliar na redução dos mesmos. As fontes de erro possíveis no processo de desenvolvimento implantado são: codificação, especificação, modificação anterior e requerimentos do usuário.

3.5 O processo de desenvolvimento de sistemas no TCDF

Devido ao crescimento do parque computacional do TCDF e, conseqüentemente, das solicitações de serviços ao setor de informática - que envolvem desde a criação de um novo sistema de informação até a instalação de um ponto elétrico - foi necessário disciplinar e categorizar as demandas com vistas a facilitar o acompanhamento das mesmas tanto pela gerência como pelos próprios usuários. Para isso foi implantado, em maio de 1999, um aplicativo na intranet do TCDF que possibilita aos usuários efetuarem suas solicitações e acompanharem o andamento dos serviços.

Esse sistema se mostrou adequado para as demandas relativas a manutenção de equipamentos, dúvidas de usuários, instalação de programas, etc., mas não permitia maior detalhamento das fases relativas ao processo de desenvolvimento de sistemas.

Assim, considerando a importância de disciplinar o processo de desenvolvimento, as razões que motivaram sua implantação descritas no item contexto da aplicação do processo, o modelo *GQM* definido e a maturidade da equipe de desenvolvimento, foram criados os documentos contendo o processo de desenvolvimento de sistemas, os formulários que suportam esse processo e desenvolvido o Sistema de Acompanhamento do Desenvolvimento de Sistema no TCDF - SADS - que permite o registro do processo e a extração das métricas estabelecidas [5]

Vale notar que esse processo vem sendo ajustado, desde setembro de 1999, para se adequar às necessidades do setor de informática. Inicialmente, foram utilizados formulários mais detalhados para as fases de codificação, baseados nos utilizados pelo SEL/NASA [5], que se mostraram de difícil preenchimento e aceitação por parte da equipe técnica. Assim, complementarmente ao registro de tempo feito por meio do RSA, o registro do tamanho dos sistemas e programas é feito por rotina de apuração semanal.

Merece destaque o fato de que o processo definido foi adequado à forma de trabalho dos técnicos e à cultura e disciplina já em uso. O trabalho de definição do processo permitiu conhecer melhor a forma de atuação da equipe e promover ajustes de procedimentos.

4 ANÁLISE DA APLICAÇÃO DO PROCESSO

O acompanhamento do processo de desenvolvimento de sistemas realizado a partir de novembro de 1999 possibilitou obter os resultados apresentados neste tópico, segundo o modelo GQM apresentado anteriormente. Para melhor compreensão dos mesmos as análises serão realizadas para cada objetivo definido, tomando-se por base as métricas obtidas até 16 de janeiro de 2001.

Os resultados obtidos da análise dessas métricas não necessariamente representam uma tendência neste momento, uma vez não há ainda uma base consolidada para comparação. Restando dizer que essas medidas passam a representar a formação de conjunto de indicadores para análises futuras mais consistentes.

4.1 Objetivo G1 - Melhoria da Qualidade dos Sistemas Desenvolvidos

O acompanhamento do número mensal de ocorrências de manutenções corretivas (métrica M1) indica evolução na qualidade dos sistemas vez que a média, nos últimos 14 meses, foi de 6 ocorrências por mês, sendo que nos últimos três meses apenas 1 ocorrência de manutenção corretiva foi identificada.

Há que se considerar que a grande maioria dos erros encontrados não se originam de produtos do atual processo de desenvolvimento, resultam de sistemas em produção antes da implantação deste processo.

4.2 Objetivo G2 - Avaliar o tempo de conversão de sistemas em Visual Basic para Java

O resultado mais expressivo para avaliação desse objetivo é a métrica que representa o tempo médio de conversão de uma linha de código VB para Java. Por essa medida cada linha de código VB é convertida para Java em 2,14 minutos, em média.

Assim, pode-se comparar o tempo médio de conversão com a média de conversão por arquivo e identificar quais funções consumiram mais recursos ou são mais complexas.

A partir da associação entre as medidas M16 (tempo médio de conversão) e M12 (LOC dos sistemas a serem convertidos) [5] pode-se obter uma estimativa do tempo necessário para a completa conversão dos sistemas. Por exemplo, o Sistema de Acesso às Bases de Dados - SABD, atualmente em conversão, por essa medida, requer 1019 horas para estar totalmente convertido em Java. Uma vez que já foram realizadas 730 horas de conversão para esse sistema (M15), pode-se deduzir que resta um esforço de 289 horas para conclusão de sua conversão.

4.3 Objetivo G3 - Avaliar o planejamento do processo de desenvolvimento de sistemas

Quanto ao planejamento de recursos, pode-se constatar, pela variação média de erro de estimativa de horas a serem despendidas na realização de uma tarefa (M20) por objetivo e fase do processo de desenvolvimento, que as estimativas para manutenção (corretiva ou evolutiva) tendem a ser maiores que os tempos realizados. Por outro lado, as estimativas para desenvolvimento de novas funcionalidades tem sido menores que os tempos reais.

Relativamente às estimativa de datas para início e conclusão de cada fase por objetivo (M22 e M24), verifica-se uma tendência de acerto para a previsão de início e de superestimação das datas de término.

Os erros de estimativa, de um modo geral, decorrem do fato de a base de informações do processo estabelecido estar ainda em construção. Além disso, observa-se uma tendência dos técnicos em não acompanharem os resultados de suas previsões.

4.4 Objetivo G4 - Avaliar a qualidade do processo de desenvolvimento de sistemas

No período observado foram identificados, pela equipe de desenvolvedores, 10 erros na atividade de leitura de código (M41) e 22 erros na atividade de teste de integração e implantação (M42).

O resultado dessa avaliação é ainda pouco expressivo. Com o curto período de observação e a falta de hábito dos técnicos, os números apontados nesse objetivo mostram-se muito modestos, o que inviabiliza uma melhor análise e indica necessidade de reforçar a orientação para uso correto do processo.

4.5 Outros resultados observados

As informações coletadas permitiu-nos inferir o perfil das atividades relativas a desenvolvimento de sistemas no âmbito do TCDF. Quando se observa a distribuição do tempo empregado com relação às fases do ciclo de desenvolvimento de sistemas, obtém-se relação mostrada na Tabela 1:

Fase	% Tempo
Análise	10
Codificação	85
Teste	5

Tabela 1: Distribuição de Esforço em Relação as Fases de Desenvolvimento

Quando se observa a distribuição do tempo com relação aos objetivos da atividade de desenvolvimento de sistemas, obtém-se a distribuição de esforço em termos percentuais mostrada na Tabela 2:

Objetivo	% Tempo
Desenvolvimento	78
Manutenção Evolutiva	8
Manutenção Corretiva	14

Tabela 2: Distribuição de Esforço em Relação ao Tipo de Atividade de Desenvolvimento

5 CONCLUSÃO

Pode-se concluir que, ressalvadas as observações quanto ao objetivo G4, a implantação do processo de desenvolvimento de sistema, produto deste trabalho, tem-se mostrado uma etapa válida na busca da qualidade e no aperfeiçoamento contínuo dos produtos de software criados pelo Tribunal de Contas do Distrito Federal (TCDF).

Pode-se destacar, principalmente, a qualidade da informação que se extrai do objetivo G2, onde torna-se possível uma estimativa precisa do esforço necessário para conversão dos sistemas VB para Java.

A construção de um processo de desenvolvimento de software induz a uma aprendizagem e conseqüente caracterização do funcionamento da instituição e de sua forma de trabalho.

Finalmente, é possível assegurar que a utilização de conceitos de modelos consagrados para grandes organizações (como CMM, QIP/EF) são aplicáveis a pequenos projetos ou pequenas organizações ou, ainda, departamentos setoriais de informática em empresas cujo negócio principal não seja a produção de sistemas.

Acreditamos que este artigo poderá ser útil para outros profissionais da tecnologia da informação de software que tenham interesse em construir um processo de desenvolvimento, em mensurar qualidade do software produzido e, dadas as características do estudo de caso ora apresentado, atuem em pequenas *software houses* ou em pequenos setores de desenvolvimento de sistemas de organizações cujo negócio principal não seja produção de software.

REFERÊNCIA BIBLIOGRÁFICA

- [1] BASILI, Victor R.; CALDIERA, Gianluigi; ROMBACH, H. Dieter. *The Experience Factory*. Institute for Advanced Computer Studies / Department of Computer Science / University of Maryland (EUA); & FB Informatik / Universität Kaiserslautern (Alemanha). In: <http://www.cs.umd.edu/projects/SoftEng/tame/tame.html>.
- [2] _____. *The Goal Question Metric Approach*. Institute for Advanced Computer Studies / Department of Computer Science / University of Maryland (EUA); & FB Informatik / Universität Kaiserslautern (Alemanha). In: <http://www.cs.umd.edu/projects/SoftEng/tame/tame.html>.
- [3] BRASIL. CÂMARA LEGISLATIVA - DF. *Lei Orgânica do Distrito Federal*, Brasília: Câmara Legislativa, 1993.
- [4] BRASIL. CONGRESSO NACIONAL. *Constituição Federal*, Brasília: Congresso Nacional, 1988.
- [5] BRASIL, C. B. & DE SOUZA, F. J. F., *Processo de Engenharia de Software para Pequenas Equipes*. Universidade Católica de Brasília, Pós-Graduação *Lato-Sensu* em Engenharia de Software, Monografia, Julho de 2000.
- [6] BRIAND, Lionel; EL EMAM, Khaled; MELO, Walcélio L. *An Inductive Method for Software Process Improvement: Concrete Steps and Guidelines*. In: Elements of Software Process Assessment and Improvement. Los Alamitos, California (EUA): IEEE Computer Society, 1999. 384 p. p. 113-130.
- [7] DION, Ray. *Starting the Climb Towards the CMM Level 2 Plateau*. In: Elements of Software Process Assessment and Improvement. Los Alamitos, California (EUA): IEEE Computer Society, 1999. 384 p. p. 259-269.
- [8] DISTRITO FEDERAL (Brasil). Tribunal de Contas do Distrito Federal -DF. *Plano Estratégico do Tribunal: PLANEST; Período 1999-2003*, Brasília: TCDF/DIPLAN, 1998.
- [9] _____. *Tecnologia da Informação – Triênio 2000-2002*, Brasília: TCDF/NIPD, 1999.
- [10] JOHNSON, Donna L.; BRODMAN, Judith G. *Tailoring the CMM for Small Businesses, Small Organizations, and Small Projects*. In: Elements of Software Process Assessment and Improvement. Los Alamitos, California (EUA): IEEE Computer Society, 1999. 384 p. p. 239-257.
- [11] MCGARRY, Frank; PAGE, Gerald; BASILI, Victor; et al. *Software Process Improvement in The NASA Software Engineering Laboratory*. Technical Report CMU/SEI-94-TR-22. ESC-TR-94-022. Dezembro, 1994. NASA/Goddard Space Flight Center; Computer Sciences Corporation (EUA); & University of Maryland (EUA). In: <http://www.cs.umd.edu/projects/SoftEng/tame/tame.html>.
- [12] NASA. *Sample SEL Data Collection Forms: Development Estimates Form and Report Form*. In: <http://sel.gsfc.nasa.gov/data/forms.htm>.
- [13] NASA. *Software Process Improvement Guidebook*. NASA – Janeiro, 1996. NASA-GB-001-95. In: <http://www.ivv.nasa.gov>.
- [14] PAULK, Mark C.; WEBER, Charles V.; CHRISSIS, Mary Beth. *The Capability Maturity Model for Software*. In: Elements of Software Process Assessment and Improvement. Los Alamitos, California (EUA): IEEE Computer Society, 1999. 384 p. p. 3-22.
- [15] PRESSMAN, Roger S. *Engenharia de Software*. Tradução de José Carlos Barbosa dos Santos. São Paulo: Makron Books, 1995. 1056 p. p. 721-875.