# NOVO PONTO DE VISTA NA CRIAÇÃO DE SISTEMAS COM APLICAÇÃO DE MODELOS DE QUALIDADE PARA SOFTWARE

**Danillo Leal Belmonte** – belmonte@pg.cefetpr.br
Centro Federal de Educação Tecnológica do Paraná – Unidade Ponta Grossa
Av. Monteiro Lobato, Km 04
84016-210 – Ponta Grossa – PR
**Prof. Dr. Luciano Scandelari** – luciano@cefetpr.br
Centro Federal de Educação Tecnológica do Paraná – Unidade Curitiba
Av. Sete de Setembro, 3165
80230-901 – Curitiba – PR
**Prof. Dr. João Luiz Kovaleski** – kovaleski@pg.cefetpr.br
Centro Federal de Educação Tecnológica do Paraná – Unidade Ponta Grossa
Av. Monteiro Lobato, Km 04
84016-210 – Ponta Grossa – PR

***Resumo:*** *Este artigo tem como objetivo o estudo de uma nova concepção às fábricas de software. Por meio de uma revisão bibliográfica dos principais estudos desenvolvidos sobre o tema, tem-se que a forma de desenvolvimento atual, somente com base na utilização de componentes e baseado em algumas ferramentas de Tecnologia de Informação, é, quase que em sua totalidade, adotada em projetos experimentais e em propostas open-source, e, por isso, não aplicados ao mercado comercial. Com isso, propõe-se uma nova visão na criação de sistemas através da fábrica de software, utilizando primeiramente o conceito de arquiteturas integradas e, posteriormente, a adoção de modelos de qualidade para software. Nota-se que a maior deficiência nos projetos é a não adoção de uma metodologia para o controle da qualidade. No final do projeto, apenas a fase de testes é tida como o controle de qualidade dado ao software.*

***Palavras-chave:*** *Fábricas de software, Projeto, Qualidade, Software.*

## 1. INTRODUÇÃO

É inegável que software é um dos componentes de maior importância em qualquer atividade do negócio, uma vez que o tratamento da informação, de forma precisa e correta, diferenciará uma empresa de suas concorrentes.

O software de computador é uma dentre poucas tecnologias-chave que causaram impacto sobre quase todos os aspectos da sociedade. Ele é um mecanismo para automatizar negócios, a indústria e o governo, um meio de transferir novas tecnologias, um modo de captar valiosa experiência para ser usada por outros, uma janela para o conhecimento corporativo coletivo. O software é fundamental em quase todos os aspectos dos negócios. Encontramos o software (freqüentemente sem nos darmos conta dele) quando nos dirigimos ao trabalho, fazemos qualquer compra no varejo, paramos no banco, fazemos uma chamada

telefônica, visitamos o médico ou realizamos qualquer uma das centenas de atividades do dia-a-dia que refletem a vida moderna (PRESSMAN, 1995).

O mercado global para software chegou a um e meio trilhão de dólares, afirma Meira (2004), quando considerados softwares embutidos (em equipamentos) e desenvolvimento realizado por empresas de software.

A pequena empresa desenvolvedora de software brasileiro, sobrevive apenas com contratos locais de pouca exigência de produtividade e qualidade. Por isso, são poucos no Brasil, os negócios de software que têm condições de exportar. Apenas os que atendem os requisitos de qualidade o fazem.

Para a melhora do desenvolvimento de software brasileiro, foi criado o conceito de fábrica de software. Oliveira e Neto (2003) definem que fábrica de software utiliza a metodologia de criação de software através da aproximação da abordagem da criação de produtos na gestão da manufatura. Sua principal característica é o reaproveitamento de código de programas já desenvolvidos. Porém, no quesito qualidade aliada à produtividade, seu conceito é baixo. As fábricas de software vivem de projetos experimentais, baseados em propostas *open-source* (código-livre), não aplicados comercialmente quase em sua totalidade.

## 2. FÁBRICA DE SOFTWARE

São várias as definições de fábrica de software. Porém, antes de saber seu significado, é necessário saber o que é software. Brooks (1987) descreve que:

> Software é um conjunto de construções conceituais, complexas e não-lineares, sujeitas a mudanças e modificações, e invisível, destacando, ambos, desta maneira a dificuldade e complexidade envolvida na atividade de construção de um software, muitas metodologias para a criação de software apontam para a aproximação cada vez maior com a abordagem da criação de produtos na gestão da manufatura.

Tal definição utiliza-se da gestão da manufatura. Com isso, podemos comparar a produção de software com a produção industrial, onde os equipamentos necessitam de atenção especial, pois incidem diretamente nas medidas de produtividade, requerendo estratégias que fazem-nas produzir mais com o mínimo possível de interrupções (OLIVEIRA e NETO, 2003).

Oliveira e Neto (2003) entendem que o estágio atual da Tecnologia da Informação (TI) determina que softwares são constituídos por módulos ou peças menores que são acopladas para a montagem do produto final. Esses módulos possibilitam que partes individuais possam ser desenvolvidas independentemente das demais, desde que planejadas.

O conceito de fábrica de software, explica Brito (2000), está baseado na idéia de prover uma linha de produção de soluções que atendam às necessidades específicas de cada cliente através da formalização de todas as atividades e seus produtos, com etapas e tarefas bem definidas para cada tipo de profissional, indo da produtividade da linha de produção à qualidade. Porém, Dias (2004) assegura que a produtividade e a qualidade ainda não estão sendo atingidas.

Com isso, pode-se associar fábrica de software à gestão da manufatura e dizer que é um processo inovador, que permite a construção de software, totalmente alinhado com a

necessidade para qual será construído, oferecendo um produto compacto, porém passível de incremento, adaptável e seguro.

## 3. ARQUITETURA INTEGRADA

As ferramentas de modelagem de processos são eficientes em levantar, modelar e redesenhar processos, a implementação efetiva de sistemas de informação, diretamente a partir dos processos, tem apresentado limitações. Por outro lado, as ferramentas CASE (*Computer Aided System Engineering*), partindo das visões de modelos, buscam construir sistemas que suportem os processos (CAMEIRA, 2003).

A engenharia de software utiliza ferramentas de apoio à modelagem de processos. A UML (Linguagem Unificada de Modelos) é uma ferramenta de modelagem que auxilia vários aspectos da construção de um sistema, dentre os quais Santos (2002) destaca:

- Formalização e padronização da modelagem de processos;
- Armazenamento de representações reutilizáveis de processos;
- Apoio à melhoria dos processos e do desenvolvimento de sistemas de suporte à operação;
- Maior facilidade para o gerenciamento dos processos;
- Aceleração da capacidade de desenvolvimento e de adequação dos sistemas que suportam os produtos e serviços;
- Aumento da flexibilidade frente às variações da demanda;
- Melhoria das interfaces processuais.

A definição completa da UML é, segundo Rumbaugh (1999), seu criador:

> UML é uma linguagem de modelagem de propósito geral que é usada para especificar, visualizar, construir e documentar os artefatos de um sistema de software. Ela captura decisões e entendimentos sobre sistemas que devem ser construídos. É usada para entender, desenhar, pesquisar, configurar, manter e controlar a informação sobre certo sistema. Objetiva ser utilizada por todos os métodos de desenvolvimento, estágios do ciclo de vida e domínios de aplicações e mídias. É uma linguagem de modelagem que objetiva unificar a experiência passada sobre técnicas de modelagem e incorporar as atuais melhores práticas em uma abordagem padrão. UML inclui conceitos de semântica, notação e linhas mestras. Possui partes estática, dinâmica, ambiental e organizacional. Objetiva ser suportada por ferramentas de modelagem visuais que possuam geradores de código e de relatórios. A UML não define um processo padrão mas objetiva ser útil em um processo interativo de desenvolvimento. Busca suportar a maioria dos processos de desenvolvimento orientados a objetos.

Para construir sistemas componentizados a partir de uma representação de processos, é necessário o entendimento da componentização de processos e de sistemas.

## 3.1. COMPONENTIZAÇÃO DE PROCESSOS E DE SISTEMAS

Através da componentização das representações de processos, Cameira (2003) afirma que se pode observar que algumas atividades, ou alguns "pedaços" de processos que compõem processos menores são semelhantes.

Estas atividades deverão gerar componentes capazes, pelas características das aplicações e padrões de integração, de facilmente se conectarem.

As atividades iguais gerarão apenas um componente que será reutilizado nos diversos processos.

De acordo com Pressman (1995), reutilização, ou reusabilidade, é uma característica importante de um componente de software de alta qualidade, ou seja, o componente deve ser planejado e implementado de forma que possa ser reusado em muitos programas diferentes. Um componente reusável possibilita que o engenheiro de software crie novas aplicações a partir de partes reusáveis.

Cada componente de processo terá relação com o componente de sistema que viabiliza o sistema daquele processo componentizado.

A representação da "passagem" processos-sistema é realizada, pelos modelos da UML, que fazem a ligação entre os processos que representam as "regras de negócio" e a análise de sistemas (KULAK, 2000).

## 3.2. AUTOMAÇÃO POR INTERMÉDIO DE FERRAMENTAS CASE

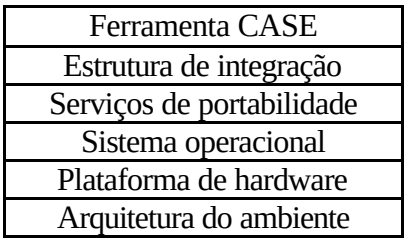
Na engenharia de software auxiliada por computador, Pressman (1995) propõe uma oficina de engenharia de software. Três características são primordiais para o sucesso da oficina:

1. Uma coleção de ferramentas úteis que ajudem em cada passo da construção de um produto;

2. Um layout organizado que possibilite que as ferramentas sejam encontradas rapidamente e usadas eficientemente;

3. Um artesão habilitado que entenda como usar as ferramentas efetivamente.

Figura 1 – Estrutura de uma ferramenta CASE (PRESSMAN, 1995)



### 3.2.1. AMBIENTE CASE INTEGRADO (I-CASE)

Para a industria de ferramentas de desenvolvimento de software, o verdadeiro poder do CASE só é obtido mediante integração.

Entre os benefícios do case integrado, Forte (1990) inclui:

- A transferência harmoniosa de informações (modelos, programas, documentos, dados) de uma ferramenta para outra e de uma etapa de engenharia de software para a seguinte;

- Uma redução do esforço exigido para realizar atividades como: gerenciamento de configuração de software, garantia de qualidade e produção de documentação;

- Aumento no controle do projeto, que é obtido por meio de um melhor planejamento, monitoração e comunicação;

- Coordenação melhorada entre os membros de uma equipe que esteja trabalhando em um projeto de software.

## 4. QUALIDADE DE SOFTWARE

Com a revolução tecnológica, a indústria passou a valorizar não somente o produto, mas todas as fases que a envolvem sua elaboração e posterior entrega. Passou a se comprometer com qualidade. Como a qualidade ainda está aquém da sua utilização, sugere-se a inserção de um modelo de qualidade a ser seguido no processo de desenvolvimento de sistema em uma fábrica de software.

Assim sendo, segue a definição de Pressman (1995) para qualidade de software: “conformidade a requisitos funcionais e de desempenho explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo software profissionalmente desenvolvido”.

Em linhas gerais, o software, como qualquer outro produto desenvolvido pelo homem, pode ser avaliado pela satisfação do cliente. Porém, Inthurn (2001) ressalva que a qualidade no desenvolvimento de sistemas passa pelas seguintes etapas:

- Alinhamento total entre as necessidades/expectativas do usuário e as especificações geradas;

- Compatibilidade entre especificações aprovadas e o produto construído;

- Produto final com menor quantidade de erros possível.

O desenvolvimento de software de qualidade depende de suas especificações, pois erros exercem forte pressão sobre custos e prazos de desenvolvimento. Para Inthurn (2001), algumas características devem ser levadas em conta:

a) Manutenibilidade: deve ser facilmente modificado quando se fizer necessário;

b) Reutilizabilidade: ao se realizar o desenvolvimento de outro software com área de aplicação semelhante, a especificação de um produto deve servir de apoio para o desenvolvimento de outro produto;

c) Avaliabilidade: uma especificação deve permitir o controle de sua qualidade;

d) Implementabilidade: avaliação da viabilidade de implementação do sistema.

### 4.1. ISO/IEC 9126 (NBR 13596)

Conforme Inthurn (2001), a ISO (Organização Internacional de Padrões) publicou uma norma que representa a atual padronização mundial para a qualidade de produtos de software. Esta norma chama-se ISO/IEC 9126 e foi publicada em 1991. Ela é uma das mais antigas da área de qualidade de software e possui tradução para o Brasil, publicada em agosto de 1996 como NBR 13596. Apesar da sua criação, poucos utilizam normas de qualidade de software no Brasil (DIAS, 2004).

A norma ISO/IEC 9126 (NBR 13596) lista o conjunto de características que devem ser verificadas em um software para que ele seja considerado um "software de qualidade". Esta norma abrange seis grandes grupos de características:

Figura 2 – Conjunto de características a serem verificadas num software (INTHURN, 2001)

| | |
|---|---|
| ISO/IEC 9126 (NBR 13596) | Funcionalidade |
| | Confiabilidade |
| | Utilizabilidade |
| | Eficiência |
| | Manutenibilidade |
| | Portabilidade |

a) Funcionalidade: existência de um conjunto de funções que satisfazem necessidades explícitas e implícitas, e suas propriedades específicas;

b) Confiabilidade: capacidade do software manter seu nível de desempenho, sob condições estabelecidas, por um período de tempo;

c) Utilizabilidade: esforço necessário para se utilizar o software, bem como para o julgamento individual desse uso, por um conjunto de usuários explícitos e implícitos;

d) Eficiência: relacionamento entre o nível de desempenho do software e a quantidade de recursos usados, sob condições estabelecidas;

e) Manutenibilidade: esforço necessário para fazer modificações específicas no software.

f) Portabilidade: habilidade do software ser transferido de um ambiente para outro.

### 4.2. O MODELO CMM

O CMM (*Capability Maturity Model*) para software, 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 do software (ORLANDI, 2000).

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

O modelo é adotado e consagrado mundialmente pelas organizações produtoras de software.

Oliveira e Neto (2003) observam que com a adoção do modelo, obtém-se a certificação necessária para a comercialização do software no exterior. Através de dados da Softex (2003), o Brasil é apontado como o sétimo mercado mundial na produção de software. Entretanto, Dias (2004) mostra que no Brasil, apenas 6 empresas utilizam o modelo, enquanto na Índia, cerca de 200 empresas o adotam.

## 5. UMA NOVA CONCEPÇÃO NA CRIAÇÃO DE SISTEMAS ATRAVÉS DA FÁBRICA DE SOFTWARE

Uma vez definida fábrica de software, pode-se então associar o desenvolvimento de sistemas a gestão da manufatura. A partir desse conceito, é importante seguir uma metodologia para o desenvolvimento de software (OLIVEIRA e NETO, 2003).

A criação de software em partes, ou módulos, deve ser planejada, para posterior acoplamento dessas peças menores à montagem do produto final.

Para o sucesso da confecção do projeto, propõe-se a utilização de uma arquitetura integrada, partindo das visões de modelos e buscando construir sistemas que suportem processos (CAMEIRA, 2003).

É de vital importância a utilização de ferramentas à modelagem de processos, pois caso contrário estar-se-á praticando a modelagem clássica. Numa segunda etapa, dispor as informações de forma gráfica, para facilitar o entendimento de outros profissionais da área e a futura documentação.

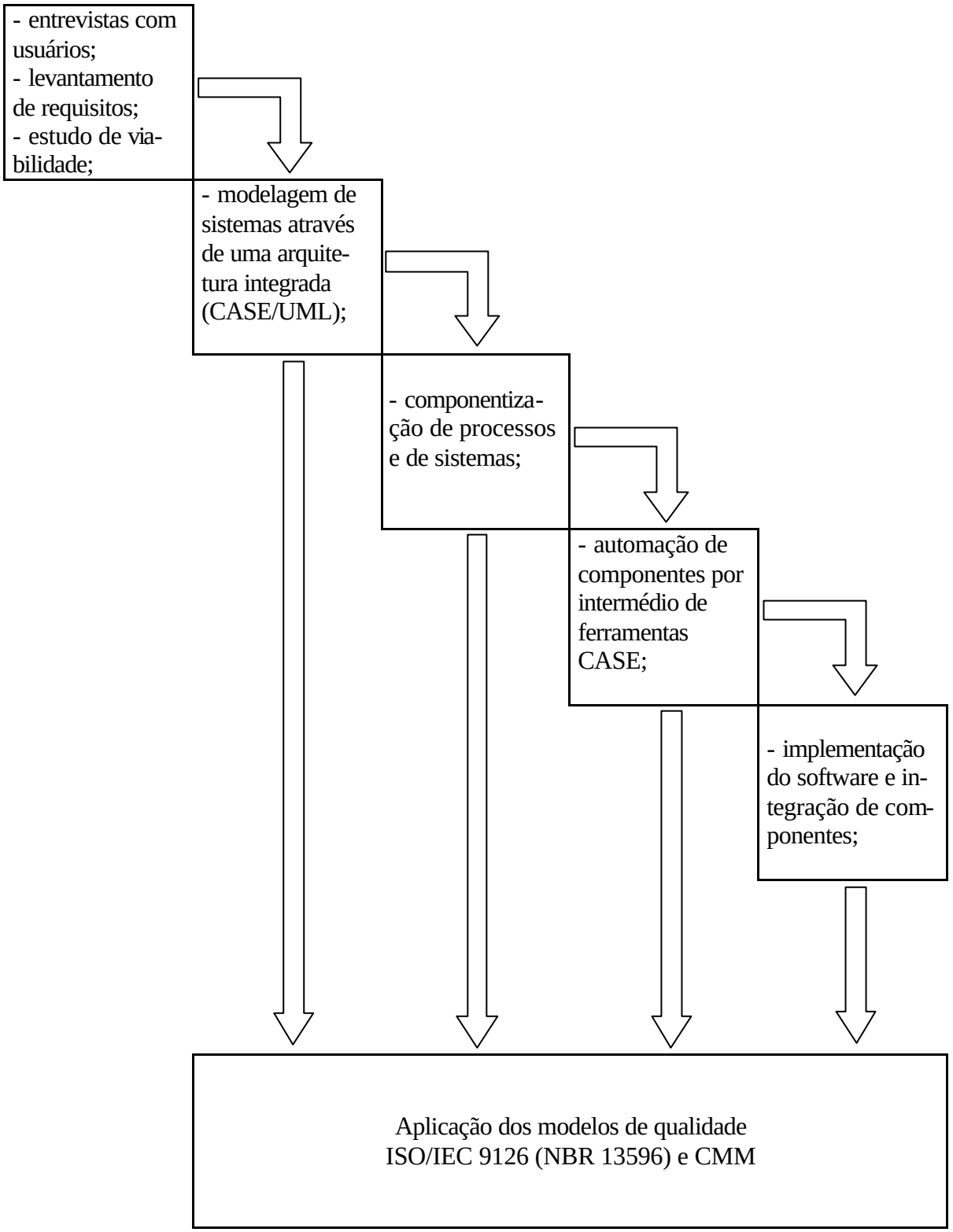
Essa componentização, deverá gerar componentes capazes, pelas características das aplicações e padrões de integração, de facilmente se conectarem. As atividades iguais gerarão apenas um componente que será reutilizado nos diversos processos. A UML faz a ligação entre os processos e o sistema e auxilia a construção do sistema (SANTOS, 2002).

As ferramentas de modelagem devem ser integradas para ocorrer o compartilhamento de informações entre as várias etapas na construção de um software e reduzir o esforço do gerenciamento do projeto. Ela auxiliará a coordenação dos envolvidos no projeto do software (FORTE, 1990).

Deve-se ter em mente que, apesar de a confecção de um software seguir os passos anteriores, de nada vale se não tiver a utilização da chamada “qualidade de software”. Na maioria das fábricas de softwares existentes no Brasil inexiste a qualidade, principalmente a qualidade através do modelo CMM. Dias (2004) concluí isso na comparação do Brasil com a Índia.

Segundo o cenário proposto acima, segue o novo modelo definido através do estudo realizado:

Figura 3 – Implantação de modelos de qualidade na criação de sistemas



## 6. CONSIDERAÇÕES FINAIS

Entre seus diversos aspectos, a fábrica concentra sólidas atividades de capacitação tecnológica e permite o surgimento de projetos inovadores.

As fábricas de software existentes no Brasil, quase em sua totalidade, são baseadas em alternativas de código-livre, no desenvolvimento de software não comercial e em projetos sem fins lucrativos.

Num período marcado por mudanças corporativas constantes, o conceito de fábrica de software tem um forte embasamento e permanece desde o seu surgimento. Contudo, o estudo realizado sobre as fábricas de software apontou as principais deficiências encontradas, e indica uma nova concepção com a mudança do processo produtivo e a inserção do controle de qualidade na criação do software em todo o seu projeto de desenvolvimento. Foi apontado também o desinteresse com projetos comerciais, onde há necessidade de inovação e têm-se recursos para investimentos em novas tecnologias de desenvolvimento.

Graças à forma de produção de software e a carência de uma metodologia para o controle da qualidade do software apontada por Dias (2004), foi proposta uma nova concepção no desenvolvimento de software através da fábrica de software, e a maior preocupação foi a inserção do modelo CMM, para a futura comercialização do software.

Pode-se dizer que, embora pareça algo complicado, se planejada a inserção da qualidade desde o início do desenvolvimento do projeto do software, ao final tem-se toda a documentação baseada nos padrões de qualidade, podendo ser comercializado no Brasil e no exterior.

## 7. REFERÊNCIAS BIBLIOGRÁFICAS


BRITO, J. **Metodologia para Gestão do Processo de Qualidade de Software para Incremento da Competitividade da Mobile**. Disponível em <http://www.mct.gov.br/temas> acesso em 13-abr-2004, 2003.

BROOKS, F. **No Silver Bullet: Essence and accidents of software engineering**. IEEE Computer, 1987.

CAMEIRA, R.; *et all*. **Componentização de Processos e de Sistemas: Impactos Metodológicos na Implantação de Sistemas Orientados por Processos**. In: ENEGEP, Ouro Preto. Anais Eletrônicos... ABEPRO. 1 CD. Minas Gerais, 2003.

DIAS, A. **Fora de rota**. Revista Você S/A, São Paulo, nº 70, p. 28-31, abr 2004, 2004.

FORTE, G. **Integrated CASE: a definition**. 3º Annual Team-Workers Intl. User Group Conference. Cadre Technologies. Providence: RI, 1990.

INTHURN, C. **Qualidade e Teste de Software**. Florianópolis: Visual Books, 2001.

KULAK; G. **Use Cases: Requirements in Context**. Boston: MA, 2000.

MEIRA, S. **Política de Software e Realidade Nacional**. Disponível em <http://www.meira.com.br> acesso em 10-mai-2004, 2004.

OLIVEIRA, D.; NETO, A. **Fábrica de Software: Promovendo a Criação de Empresas Competitivas em Tecnologia da Informação**. In: ENEGEP, Ouro Preto. Anais Eletrônicos... ABEPRO. 1 CD. Minas Gerais, 2003.

ORLANDI, T. R. **Processo de Implantação de Gestão de Qualidade de Software em Empresas Nacionais: O Estudo de Caso do Tribunal de Contas da União**. Brasília. 52 p. Dissertação (Mestrado em Informática) Universidade Católica de Brasília, 2000.

PRESSMAN, R. **Engenharia de Software**. São Paulo: Makron Books, 1995.

RUMBAUGH, J.; *et all*. **The Unified Modeling Language Reference Manual**. Massachusetts. Boston: MA, 1999.

SANTOS, R. **Engenharia de Processos: análise do referencial teórico conceitual, instrumentos, aplicações e casos com a finalidade de síntese sobre sua estrutura, conhecimentos, falhas e resultados**. Rio de Janeiro. 317 p. Dissertação (Mestrado em Engenharia de Produção). COPPE, UFRJ, 2002.

SOFTEX. **Slicing the Knowledge-Based Economy in India**. China and Brazil. MIT. SOFTEX: Brasil, 2003.

# A NEW POINT OF VIEW IN THE CREATION OF SYSTEMS WITH THE APPLICATION OF MODELS OF QUALITY FOR SOFTWARE

***Abstract:*** *The objective of this article's is the study of conception for software fabrics. Through a bibliographic review of the main studies wrote about the issue, it is evident that the current way of development, only based on the use of components and some information technology's tools, is, almost totally adopted in experimental projects and in open-source proposals, and, because of this, it is not applicable in the commercial market. Because of that, it is proposed a new view in the creation of systems through the software fabric, first using the concept of integrated architectures and, then, the adoption of models of quality for software. It is noticed that the main deficiency in the projects is the not adoption of a methodology for the control of quality. At the end of the project, only the tests phase is considered as the control of quality gave to the software.*

***Key-words:*** *Software fabrics, Project, Quality, Software.*