

# UMA ESTRATÉGIA PARA GERÊNCIA DO PROCESSO BASEADA NOS CUSTOS DA QUALIDADE DE SOFTWARE

Álvaro Augusto Neto  
Divisão de Ciência da Computação  
Instituto Tecnológico de Aeronáutica  
12228-900 S.José dos Campos-SP  
alvaro@comp.ita.br

Nilson Sant'Anna  
Lab. de Computação e Matemática Aplicada  
Instituto Nacional de Pesquisas Espaciais  
12227-010 S.José dos Campos-SP  
nilson@lac.inpe.br

## Resumo

Este trabalho apresenta uma estratégia para gerência do processo de produção de software, voltada à proposição de ações técnicas e gerenciais que conduzam a melhorias do seu processo de produção. Utiliza-se um enfoque baseado em aspectos eminentemente quantitativos, que permitem integrar a gerência das características técnicas dos produtos desenvolvidos, representada pelo seu controle de configuração, com as características administrativas ligadas à gerência de sua produção, enfocadas pelos controles de custos de desenvolvimento e dos custos da qualidade.

**Palavras-chave:** Qualidade de Software, Custos da Qualidade, Processos de Software.

## Abstract

This work presents a management-based strategy that suggests technical and administrative actions, leading to the improvement of software production process. This proposal is based eminently in quantitative aspects and allows to integrate control configuration of the product, with the administrative characteristics linked to the production management, evidenced by the controls of development costs and costs of quality.

**Keywords:** Software Quality, Cost of Quality, Software Process.

## 1. INTRODUÇÃO

A importância econômica do software cresceu na medida em que ele passou a ser incorporado em quase todos os produtos e atividades da sociedade moderna. Nos dias de hoje, produzir software com alta qualidade pode ser considerado como um dos fatores críticos para diferenciação de produtos e serviços, capaz de ocasionar impacto direto sobre o sucesso econômico dos empreendimentos.

A vantagem competitiva das empresas atuais está diretamente associada à melhoria de seus processos de produção de software. Conseqüentemente, essas melhorias refletem-se diretamente na composição de seus produtos e serviços, bem como, no próprio processo de gerenciamento empresarial.

Para obter um progresso consistente, as técnicas de gerenciamento da produção de software também necessitam ser aperfeiçoadas, tendo em vista que as tecnologias de produção encontram-se em fase de grande evolução tecnológica.

### 1.1. Objetivo do Trabalho

Neste trabalho propõe-se uma estratégia para gerência do processo de produção de software, baseada num enfoque eminentemente quantitativo, voltado a proposição de ações técnicas e gerenciais que conduzam ao aperfeiçoamento do seu processo de produção. Para sua implementação, utiliza-se como principal indicador gerencial, a contabilização dos custos da qualidade de software.

## 1.2. Premissas e Restrições

O enfoque gerencial dado às questões relativas à gerência da qualidade e produtividade no desenvolvimento de software antepõe-se como a principal diretiva empregada na elaboração deste trabalho.

Ao direcionar as pesquisas para esta abordagem, busca-se otimizar o processo de produção de software como um todo, não apenas com relação a alguns aspectos específicos, tendo em vista que os problemas observados distribuem-se por praticamente todas as etapas do seu desenvolvimento. A principal motivação para este enfoque deve-se principalmente a constatação de que:

- Compete à Administração conduzir os esforços organizacionais na direção das transformações necessárias ao aprimoramento dos processos produtivos utilizados;
- A gerência do desenvolvimento de software ainda não evoluiu suficientemente de forma a adequar-se às características e peculiaridades do produto e de seu processo de produção.

## 2. ASPECTOS GERENCIAIS DA PRODUÇÃO DE SOFTWARE

Em artigo de 1991 sobre as perspectivas gerenciais futuras no desenvolvimento de software, Basili e Musa já observavam que a tecnologia da informação havia permeado de tal forma as instituições modernas, que estas poderiam ser consideradas como totalmente dependentes dela [1]. Na sua essência, observa-se que as razões para esse aumento na demanda são as mesmas encontradas em outros setores da economia, e estão relacionadas com o esforço para aumentar a produtividade através da automação, conforme constata Boehm [3] e Humphrey [9].

Apesar deste crescimento significativo, observa-se que os produtos gerados pela chamada “*Indústria do Software*” não atendem satisfatoriamente aos seus usuários no tocante a prazos de desenvolvimento, custos e qualidade esperada.

Em função deste quadro e do intenso crescimento desta indústria, também tem crescido o número de estratégias e produtos propostos para resolução dos problemas envolvidos com a produção de software. As inúmeras alternativas que têm sido propostas nem sempre conduzem a resultados satisfatórios.

### 2.1. A Insuficiência das Sistemáticas Adotadas

Ao analisar-se a evolução experimentada pela tecnologia de produção de software, nota-se que ela não tem conseguido acompanhar o crescimento observado com relação ao hardware. Além disso, tem persistido a utilização de procedimentos pouco eficientes, quase não incorporando técnicas automatizadas, que em outros setores conduziram a melhorias significativas de qualidade e produtividade [12]. Freeman também destaca que as melhorias tecnológicas já obtidas são adotadas muito lentamente, permanecendo ignoradas ou rejeitadas por algumas organizações [8].

O grande número de conceitos e técnicas desenvolvidos e implementados têm possibilitado melhorias significativas, embora os progressos já alcançados mostrem-se insuficientes para atingir níveis satisfatórios de qualidade, prazo e custo exigidos pelas novas aplicações.

O cerne da questão econômica ligada ao desenvolvimento de software continua associado ao fato de que é necessário aumentar a quantidade de software produzida por unidade de mão-de-obra, capital e insumos necessários para sua elaboração. Fica desta maneira evidente, que o problema abordado é a questão clássica da produtividade econômica.

A ineficiência dos processos de produção utilizados correntemente pode ser observada

principalmente através de:

- Desperdício de insumos e recursos, tais como capital, mão-de-obra, materiais etc., utilizados para refazer várias vezes uma mesma tarefa;
- Emprego de tecnologias de produção inadequadas; e
- Demora em localizar e corrigir problemas ocorridos ao longo do processo de desenvolvimento.

A inexistência de indicadores técnicos e gerenciais satisfatórios, notadamente aqueles de caráter quantitativo, também conduz à utilização de processos de produção inadequados. Impede ainda a formação de uma base sólida para acompanhamento de projetos de novos sistemas e acarreta avaliações imprecisas dos produtos de software. Impossibilita também o desenvolvimento de estimativas confiáveis para futuros projetos de desenvolvimento, como se pode verificar através da análise das prescrições dos níveis de maturidade estabelecidos pelo CMM-Capability Maturity Model for Software [14].

Esses fatos evidenciam a necessidade de se dispor de ferramentas eficientes que permitam a gerência de desenvolvimento identificar, analisar criticamente e propor medidas que levem ao aperfeiçoamento do processo de produção de software.

## **2.2. A Procura por Indicadores Gerenciais do Processo**

Se os benefícios advindos da medição das várias etapas do processo de produção de software podem ser facilmente visualizados, a sua implantação encontra dificuldades operacionais significativas, decorrentes principalmente, das dificuldades em se encontrar medidas satisfatórias para os artefatos e/ou serviços que são produzidos ao longo do seu desenvolvimento.

Este fato, há tempos vem desafiando a comunidade de engenharia de software, pois como observou Jones, *“a medição da programação tem sido o elo mais fraco de toda a ciência de engenharia de software”* [11].

A implantação de métricas encontra dificuldades devidas principalmente ao caráter imaterial e intangível do software [7]. O cunho abstrato dos artefatos resultantes do seu processo de produção, acaba por impedir uma melhor visualização e contribui para dificultar o controle e acompanhamento pelos administradores do projeto. A grande variabilidade e despadroneização dos ambientes de produção também impede que a disseminação de métodos bem sucedidos de medição ocorram fora de limites bastante restritos, como os de uma equipe ou empresa de desenvolvimento. Isto ocorre em grande parte devido a dificuldade em adaptar esses procedimentos ao efeito de todas as variáveis que afetam o processo produtivo.

Com base nesses fatos pôde-se constatar que os indicadores usualmente utilizados não se mostram adequados à finalidade pretendida pelos autores, que consiste em subsidiar administrativamente o processo de melhoria da eficiência econômica na produção de software.

Ao analisar a chamada *“economia da qualidade”*, proposta por Juran, deparou-se com um método que permite avaliar a eficiência dos processos de produção na razão inversa do custo para se corrigir ou prevenir resultados inadequados quanto a qualidade [13]. Esta técnica, que recebe a denominação de custos da (má) qualidade, permite avaliar a eficiência econômica através da quantificação dos custos para encontrar e corrigir o trabalho defeituoso ou rejeitado, freqüentemente manifestado através das constantes necessidades de retrabalhos, modificações, correções, testes intensivos, inspeções, verificações, adiamento de prazos, ampliação de orçamentos, etc.

Por seu intermédio, a mensuração de alguns fatores intangíveis existentes ao longo do processo de desenvolvimento de software, torna-se possível através da quantificação dos seus efeitos, por meio da *“mais antiga e respeitada das medidas - o dinheiro concreto”* [5].

Assim, se a aplicação direta das medidas de produtividade econômica encontra sérias dificuldades ocasionadas pelas características intrínsecas do produto, verificou-se que era possível avaliá-la de forma indireta, através das perdas decorrentes das ineficiências do seu processo de produção. Esta métrica, por não estar associada aos aspectos tecnológicos do processo produtivo, mostrou-se adequada para utilização em qualquer ambiente de produção.

### 2.3. Os Custos da Qualidade de Software

O conceito de custo da qualidade pode ser definido como sendo o tempo e o dinheiro gastos por uma organização para prevenir, detectar e corrigir falhas e outros efeitos indesejáveis, no sentido de atender as necessidades (explícitas e implícitas) de seus clientes, e fazer com que seus produtos sejam apropriados ao fim a que se destinam. Sob o ponto de vista operacional, pode-se dizer que são os custos incorridos para garantir que o que se faz seja feito certo e, caso esta estratégia não funcione adequadamente, refazer, consertar ou substituir o que saiu errado.

Ao analisar esta questão, Crosby afirma que “*qualidade não custa dinheiro*” e sim a sua ausência [5]. Segundo Crosby, o custo da qualidade é a soma de todos os gastos incorridos quando as coisas não são feitas de maneira correta, logo da primeira vez. Assim, ao formular o seu “*quarto princípio absoluto da qualidade*”, afirma taxativamente: “*A medida da qualidade é o preço do não cumprimento, não os índices*” [6].

Sobre este enfoque, os custos da qualidade constituem-se no denominador econômico comum, por meio do qual a gerência e a produção podem estabelecer uma forma de comunicação nítida e efetiva, sobre investimentos em programas de melhoria da qualidade. Sua utilização como indicador gerencial, apresenta as seguintes características:

#### **Vantagens:**

- Permite avaliar a eficiência dos métodos de produção, em função da melhoria do desempenho econômico decorrente, expressando-os em dinheiro, que é a linguagem da alta administração;
- Possibilita priorizar os problemas, em função do impacto econômico causado;
- Permite converter em uma única unidade de medida, o dinheiro, as deficiências relativas a prazos, custos e qualidade; e
- Como é orientado para resultados financeiros, não está atrelado a nenhuma tecnologia de produção específica, que poderia desencorajar mudanças de caráter revolucionário.

#### **Desvantagem:**

- Indica quais os pontos que devem ser melhorados e não como fazê-lo.

Os custos da qualidade usualmente são classificados nas quatro categorias gerais propostas por Juran [13], como mostra a Figura 1. Os custos de controle são divididos em dois tipos: os custos de prevenção e os custos de avaliação. Já os custos provenientes das falhas nas atividades de controle, são separados em custos das falhas internas e custo das falhas externas.

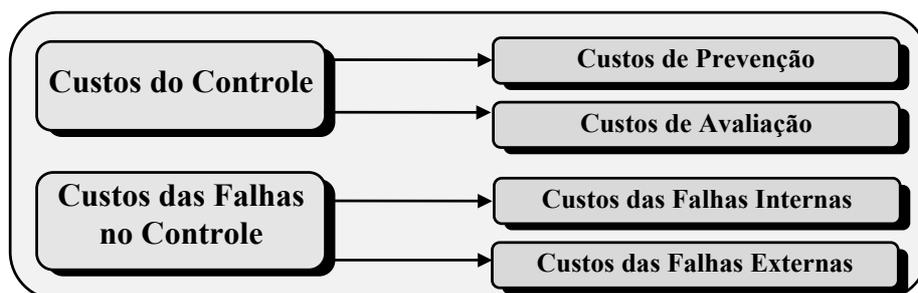


Figura 1. Taxonomia dos custos da qualidade.

Os custos do controle, também denominados de custos de conformidade, referem-se às despesas incorridas para garantir que um software atenda às suas especificações originais e esteja adequado ao uso pretendido. Divide-se em custos de prevenção, que são aqueles despendidos para evitar a ocorrência de defeitos e não conformidades, e custos de avaliação, que decorrem do esforço para identificar defeitos e não conformidades ocorridas nos sistemas desenvolvidos. Enquadram-se nesta última categoria os gastos para determinação do grau de conformidade com os requisitos de qualidade que tenham sido preestabelecidos.

Os custos das falhas no controle, também denominados de custos de não conformidade, ocorrem devido a correção de falhas e outros problemas detectados com relação ao desempenho do software produzido. Dividem-se em duas categorias. Os custos das falhas internas, estão associados aos defeitos e não conformidades encontradas nos sistemas desenvolvidos antes que eles tenham sido entregues aos seus usuários. São detectados durante a fase de desenvolvimento do software e não existiriam caso o processo de produção permitisse *“fazer certo, logo na primeira vez”*. Seus componentes usuais incluem o retrabalho, o tempo despendido para análise das causas das falhas, os recursos consumidos na sua correção, reinspeção, etc. Já os custos das falhas externas, estão normalmente associados a correção ou reparação dos defeitos e não conformidades encontradas nos sistemas desenvolvidos após eles terem sido entregues aos seus usuários. São detectados durante a fase de utilização do software pelo cliente, e englobam as despesas com garantia, suporte técnico, manutenção corretiva, perda de clientes em decorrência das deficiências de qualidade, etc.

É importante observar que o conceito de custo da qualidade, comporta sem maiores problemas os parâmetros de qualidade, custo e prazo, que podem ser facilmente convertidos em valores monetários. Desta maneira, o melhor conhecimento dos custos envolvidos com atividades de correção ou prevenção de não conformidades, apresentadas por um software em desenvolvimento, pode indicar os pontos críticos de seu processo produtivo, levando a melhores métodos de produção e vice-versa.

Deve-se salientar que nos projetos que envolvem atividade de pesquisa, fronteiras do conhecimento, ou mesmo prospecção tecnológica, como é o caso de muitos softwares complexos ou de aplicação crítica, provavelmente não será possível *“fazer tudo certo, logo na primeira vez”*. Mesmo assim, o conceito permanece válido, não no seu caráter operacional específico, mas como uma postura gerencial, no sentido de não repetir os caminhos infrutíferos já percorridos, servindo para impulsionar os trabalhos numa direção positiva.

Finalmente, é importante ressaltar que as decisões embasadas na apuração dos custos da qualidade deverão sempre focar sua repercussão ao longo de todo o ciclo de produção. Isto se deve ao fato de que é possível tomar providências de curto prazo, que levem à diminuição de determinado item de custo da qualidade sem, no entanto, impedir que o problema se amplie posteriormente. Um exemplo desta situação seria o caso em que a realização de testes de programas fosse totalmente suprimida, sem que tenha havido mudanças significativas no sentido de viabilizar a geração de código-fonte isento de erros. Num primeiro momento, os custos de inspeção diminuiriam drasticamente para aumentar os custos de retrabalho nas fases posteriores.

Pode-se denominar este aspecto do indicador como sendo o *“paradoxo das atividades não entregues”*, pois a execução de atividades como a realização de ensaios do produto, revisões técnicas e avaliações de performance, dificilmente são percebidas pelos clientes, embora contribuam para elevar os seus custos da qualidade. Porém, quando não são executadas, estas atividades concorrem significativamente para aumentar o grau de insatisfação com o produto e, por conseguinte, os seus custos referentes às falhas externas.

### 3. QUANTIFICAÇÃO DOS CUSTOS DA QUALIDADE

Como a mensuração dos custos da qualidade de software envolve aspectos operacionais complexos devido a grande variabilidade existente nos ambientes de produção, e a dificuldade em caracterizar com rigor o efeito de todas as variáveis que afetam seu desenvolvimento, torna-se difícil reunir características homogêneas aplicáveis a uma extensa gama de sistemas de produção. Verificou-se no entanto, que existem dois aspectos que se sobressaem no sentido de minimizar os esforços para sua implantação. São eles:

- A adequação da sistemática adotada para se contabilizar os custos de software;
- O acompanhamento das mudanças introduzidas nos diversos produtos intermediários que são desenvolvidos, de forma a contabilizar a grande incidência de retrabalho que é frequentemente observada.

#### 3.1. Adequação do Sistema Contábil

Com a evolução dos ambientes de produção de software, os sistemas tradicionais de apropriação dos custos de desenvolvimento, baseados na contabilidade por centros de custo departamentais ou funcionais, passaram a apresentar algumas impropriedades estruturais devido a quantidade crescente de despesas indiretas e de caráter geral, que necessitam ser rateadas por uma base de custeio direto cada vez menor, representada pelos componentes tradicionais tais como: materiais, mão-de-obra e equipamentos de uso exclusivo. Robles observou que, em certos casos, o item mão-de-obra deixou de ser relevante em termos de custos, quando comparado com os demais componentes do produto [15]. Em certas circunstâncias especiais, os gastos gerais e despesas indiretas chegam a representar o principal componente do custo de fabricação de um produto.

Estes fatos demonstraram a necessidade de se efetuar a avaliação dos custos de produção de software em novas bases, de modo a acomodar estas evoluções. Na indústria manufatureira em geral, a resposta para questões semelhantes foi a implantação dos sistemas de custeio baseados em atividades, como o ABC (Activity-Based Costing).

Nesta sistemática, os custos passaram a ser contabilizados por atividades, em vez de departamentos ou funções como nos sistemas tradicionais. Isto pode ser justificado pelo fato de que são as atividades que consomem insumos de produção, enquanto que produtos, serviços e departamentos consomem atividades para alcançar seus objetivos e finalidades.

Com isto, tornou-se possível contabilizar processos que envolvem vários departamentos e pode-se quantificar, além das despesas diretas, que usualmente incidem sobre cada produto, também a parcela das despesas indiretas efetivamente consumidas por eles. Além disso, como muitas destas atividades não estão relacionadas diretamente com o volume de produção, puderam ser custeadas dividindo-se o custo da atividade pelo número de vezes em que ela é executada. Já os custos relacionados ao volume continuaram a ser apropriados diretamente, de maneira análoga ao que era feito anteriormente.

Assim, a utilização do sistema ABC mostrou-se como a opção mais adequada para quantificar os custos de desenvolvimento de softwares produzidos através das tecnologias mais modernas. Isto se deve principalmente aos seguintes aspectos:

- Seguindo a tendência de integrar cada vez mais todas as tarefas num único ambiente de produção de software, vários recursos como os ambientes integrados de desenvolvimento e muitas das ferramentas CASE, são utilizados em várias das atividades e etapas do ciclo de vida, o que dificulta sua apropriação pelos sistemas tradicionais, muitos voltados para estratificação departamental ou funcional;
- Atividades como gerência de configuração, garantia da qualidade, certificação, etc. não conseguem ser alocadas adequadamente num único centro de custo, pois assumem

um caráter de despesas gerais, cujo critério de rateio não deve incidir igualmente sobre as várias fases, tendo em vista que a quantidade de recursos alocados em cada uma delas é diferente. Verifica-se também que com a evolução da tecnologia de produção de software, e a busca por maiores níveis de maturidade do CMM, a participação deste tipo de despesas sobre o montante total tem crescido significativamente;

- Existem atividades cuja execução freqüentemente ultrapassa os limites de uma única fase do modelo clássico de ciclo de vida, o que dificulta sua alocação em apenas um centro de custo definido segundo os critérios de apropriação tradicional;
- Diversas atividades, como a manutenção e atualização permanente dos requisitos do sistema, codificação e testes de programas, etc. ainda são refeitas várias vezes. Há portanto, grande incidência de retrabalho, que os sistemas tradicionais não conseguem contabilizar em separado, o que implica na implantação de um segundo sistema contábil para apurar os custos da qualidade; e
- Algumas atividades de acompanhamento e controle são refeitas durante praticamente todo o ciclo de vida. Isto além de gerar retrabalho, contrapõe-se aos sistemas tradicionais de apropriação, que apresentam dificuldades para separar custos que envolvem vários departamentos e/ou funções.

### **3.2. Acompanhamento das Mudanças Efetuadas**

Ao analisar as tarefas efetuadas pelos sistemas de gerência de configuração de software, verificou-se que existe grande afinidade entre o fato de que todos os itens e artefatos acompanhados devem ter algumas de suas características recadastradas cada vez que sofrem modificações, e a necessidade de serem contabilizadas as atividades que envolvem retrabalho.

Desta maneira, ao integrar-se as tarefas relativas a contabilização dos custos da qualidade com as de gerência de configuração, torna-se possível efetuar a sua medição sem causar grande impacto organizacional, tendo em vista que sistemas complexos, tradicionalmente já se utilizam de sistemas de gerência de configuração, em virtude da necessidade de dispor de grande visibilidade, rastreabilidade e controle sobre todos os componentes do sistema [4].

A seguir são analisadas as características dos principais componentes dos sistemas de gerência de configuração de software, de forma a verificar a adequabilidade desta proposição.

#### **3.2.1. Sistemas de Gerência de Configuração de Software**

Os sistemas de gerência de configuração surgiram como resposta à necessidade de lidar com a crescente complexidade observada no desenvolvimento de software ocasionada, em grande parte, pela maneira como ainda se processa a sua produção.

O fato de que uma mudança efetuada numa das fases do ciclo de vida, quase sempre se refletir sobre a execução das etapas de produção posteriores, apresenta-se como elemento adicional para aumentar essa complexidade. Este acréscimo ocorre, principalmente, nos casos dos sistemas que apresentam grande extensão, ou múltiplas funcionalidades, pois a dificuldade em gerenciar a produção cresce exponencialmente com o tamanho do sistema em linhas-de-código fonte [2].

A forma de operacionalizá-la, é através da introdução de procedimentos formais que permitam acompanhar, avaliar e verificar se cada mudança efetuada atingiu seus objetivos e, se as alterações decorrentes também foram implementadas a contento.

A seguir, procura-se analisar sua interação com a contabilização dos custos da qualidade de software.

### 3.2.1.1. Identificação dos Componentes

A primeira tarefa ligada ao gerenciamento da configuração de software, consiste em separar e identificar os diversos produtos intermediários que são gerados ao longo do processo de desenvolvimento, de forma que seja possível acompanhar sua evolução ao longo do tempo.

Conforme o desenvolvimento do sistema evolui, o desmembramento dos diversos componentes deve prosseguir até que seja possível obter um nível satisfatório de controle. A etapa seguinte consiste em descrever as interdependências entre os diversos itens, de forma que seja possível identificar a propagação dos efeitos das alterações efetuadas em cada um deles. O conhecimento dos relacionamentos existentes entre os diversos itens de configuração, também permite acompanhar sua evolução, e auditar o processo de efetivação das mudanças.

A partir da identificação dos componentes cuja configuração será acompanhada, torna-se possível controlar sua evolução [2], estabelecer “*baselines*” que representem o estágio de desenvolvimento do sistema em determinado instante do tempo, e ainda possibilitar o controle de versão de cada componente, conforme ocorra sua liberação.

### 3.2.1.2. Controle das Mudanças

Os sistemas de gerenciamento de configuração costumam empregar procedimentos formais, que possibilitam o controle das alterações e mudanças a serem executadas.

Normalmente este processo tem início com a identificação de uma necessidade de mudança em um ou vários componentes do sistema. Esta necessidade é expressa através de uma solicitação formal de mudança, freqüentemente denominada de solicitação de mudança de engenharia. A solicitação é então avaliada e submetida à apreciação de um comitê de gerência de configuração, que poderá aprová-la ou rejeitá-la. Caso seja aprovada, a mudança é implementada, e os itens de configuração que passarem por alterações têm suas novas características recadastradas em uma nova versão, que será implementada imediatamente ou aguardará a liberação de uma nova “*baseline*” do projeto.

O ciclo genérico de implementação de mudanças descrito resumidamente acima apresenta grande interesse dentro do escopo deste trabalho, pois pretende-se aprofundar a análise das causas que ocasionam as modificações nos itens acompanhados pelo sistema de gerência de configuração de software.

Ao analisar-se as causas básicas que acarretam a necessidade de efetuar estas alterações, pode-se verificar que muitas vezes elas decorrem das deficiências do processo produtivo, ou seja, da impossibilidade de “*produzir certo, logo na primeira vez*” proposta por Crosby [6]. Conseqüentemente, os recursos despendidos na sua execução constituem-se em custos da qualidade incorridos na elaboração de cada item.

Desta maneira, torna-se possível obter um levantamento preciso das principais deficiências de qualidade de um processo de produção que esteja sendo monitorado, ao se agregar ao sistema de gerência de configuração, uma sistemática que permita classificar estas ocorrências em custos devido a falhas na prevenção, na avaliação, nos testes, ou ainda, devido a correção das falhas que tenham sido detectadas.

### 3.2.1.3. Auditoria da Configuração

Segundo Bersoff, execução das auditorias de configuração têm como objetivo assegurar que a visibilidade e a rastreabilidade entre os diversos itens de configuração, permaneçam adequadas após terem sido realizadas diversas mudanças no seu conteúdo [2]. Segundo Bryan & Siegel, “*eliminar a execução das auditorias de configuração constitui-se em falsa economia*”, o que também corrobora com a classificação das despesas resultantes da execução destas tarefas como custos da qualidade pertencentes a categoria prevenção [4].

### 3.2.1.4. Registro do Status Atual

As tarefas ligadas ao registro do “*status*” atual de um sistema em desenvolvimento visam fornecer às pessoas que participam do projeto, uma posição detalhada sobre cada item de configuração que é acompanhado.

A importância de sua utilização é tanto maior, quanto for o tamanho e a complexidade dos projetos gerenciados. Nestes casos, para minimizar a ocorrência de retrabalho, torna-se necessário dispor de um conjunto homogêneo de informações, que possa ser compartilhado pelas diferentes equipes que atuam simultaneamente no desenvolvimento dos vários itens de configuração, que tenham sido implementados.

## 4. ESTRATÉGIA PROPOSTA

Como visto anteriormente, as ações gerenciais sobre as questões que envolvem a produção de software devem fundamentar-se em fatos e dados concretos. Quando isto não ocorre, verifica-se a reincidência dos problemas, ou então, a introdução de novos problemas, tendo em vista que a causa básica não foi neutralizada. Este fato coloca o sistema de produção num círculo vicioso, onde cada ação corretiva implementada pode originar novos problemas para corrigir. Uma das manifestações mais visíveis destes fatos ocorre quando a manutenção de um software é efetuada sem maior critério técnico.

A estratégia proposta a seguir não tem por objetivo abordar as questões específicas da tecnologia de produção de software, mas assegurar que os produtos resultantes de sua aplicação terão maior qualidade e menores custos. Seu objetivo maior é o de proporcionar aos gerentes de desenvolvimento um melhor conhecimento sobre quais deficiências da tecnologia de produção causam maior prejuízo, devendo portanto merecer maior atenção gerencial. Com isto procura-se o aperfeiçoamento do processo produtivo através da sua otimização ou da introdução de novas tecnologias específicas.

A solução proposta consiste em integrar em um único ambiente de produção, a gerência das características técnicas do produto, representada pelo controle de configuração, com as características administrativas ligadas à gerência da sua produção, enfocadas pelos controles de custos de desenvolvimento e dos custos da qualidade.

Esta sinergia entre os aspectos técnicos e gerenciais, possibilita não só um maior embasamento para as decisões gerenciais sobre a produção, mas também, uma redução no esforço administrativo necessário ao controle do projeto. Se a estes benefícios forem acrescentadas as melhorias de eficiência proporcionadas pela utilização dos sistemas de custos da qualidade, pode-se verificar que a sistemática proposta apresenta uma relação custo-benefício extremamente favorável à sua utilização.

Deve-se ressaltar o caráter inovador da sistemática apresentada, que também propõe que a contabilização dos custos de desenvolvimento seja feita por atividades, tendo em vista que os sistemas tradicionais mostram-se inadequados aos ambientes de desenvolvimento mais modernos, onde existe grande incidência de recursos tecnológicos para apoio ao processo.

Para que estes objetivos possam ser alcançados, no cerne desta abordagem encontra-se a coleta de dados sobre o processo de produção de software. Da relevância dos dados coletados dependerão todos os resultados obtidos. Para sua implementação, propõe-se que essas ações sejam consolidadas através da realização das seguintes etapas:

1. Identificação, organização e controle das mudanças introduzidas nos itens de configuração, produzidos ao longo de todas as fases do processo de produção do software, que esteja sendo desenvolvido;
2. Coleta de dados significativos sobre os custos da qualidade observados no desenvolvimento destes itens;

3. Priorização das ações visando a correção dos problemas decorrentes da ausência de qualidade, em função do seu impacto econômico;
4. Identificação das causas básicas destes problemas;
5. Desenvolvimento de soluções apropriadas, visando a remoção destas causas; e
6. Planejamento e execução das mudanças necessárias.

A seguir são detalhadas as atividades relacionadas com a execução dessas etapas.

#### **4.1. Identificação, Organização e Controle das Mudanças nos Itens de Configuração**

A primeira atividade relacionada com a implementação da estratégia proposta neste trabalho, prevê a implantação de controles técnicos e gerenciais sobre todos os artefatos e produtos intermediários gerados ao longo das etapas de produção. Com isto torna-se possível controlar e acompanhar a evolução de cada um desses componentes ao longo do tempo.

Normalmente essa evolução se manifesta através da necessidade de alterações no conteúdo estabelecido inicialmente, ou então, por meio do desdobramento de um item em diversos subitens, cujo teor apresenta em maior detalhe algumas características do componente hierarquicamente superior. Um exemplo de como essa decomposição se processa pode ser visualizado através do processo de codificação de programas, onde a partir de uma especificação, detalham-se os diversos subsistemas envolvidos, que resultarão em módulos, cuja decomposição irá produzir a estrutura dos programas a serem codificados.

Assim, a partir da identificação dos diversos componentes, deve-se passar a controlar sua evolução tendo em vista a obtenção de uma melhor gerência e controle sobre as etapas envolvidas no processo de produção.

#### **4.2. Coleta de Dados sobre os Custos da Qualidade**

Após a identificação e controle sobre os diversos produtos e artefatos intermediários que são gerados ao longo do desenvolvimento, deve-se implementar o controle dos custos envolvidos na sua produção e alteração.

Para sua execução, propõe-se que cada um dos itens de configuração identificados anteriormente seja considerado como um centro de custos. Assim, a partir de uma sistemática de custeio baseada em atividades, será possível classificar cada uma das despesas envolvidas na sua elaboração ou alteração, como custos de desenvolvimento ou como custos da qualidade.

A sistemática proposta para sua implementação permite integrar as funções de controle das características técnicas dos itens de configuração de software, com o controle individualizado dos seus custos de produção, bem como, com as despesas decorrentes dos custos da qualidade relativos a cada um de seus componentes.

#### **4.3. Priorização das Ações Visando a Correção dos Problemas**

A partir dos dados coletados, pode-se elaborar análises de tendência e gráficos de Pareto, de forma a estabelecer uma escala de prioridades para as ações gerenciais em função da magnitude do impacto econômico ocasionado. Desta maneira, devem ser prioritariamente abordados aqueles pontos onde ocorram as maiores deficiências, representadas pelos custos da qualidade mais elevados.

Este tipo de análise tem sido bastante utilizado na gestão da qualidade, pois permite identificar quais os itens que tem maior participação na composição de determinado efeito que se deseja eliminar. Sua utilização permite restringir as ações corretivas somente aos pontos capazes de causar maior impacto na redução dos efeitos indesejados, o que minimiza a dispersão de esforços e simplifica a sua gerência, contribuindo significativamente para a

melhoria da eficiência econômica do processo de produção.

Ao utilizar este instrumento gerencial para subsidiar o processo de melhoria da qualidade do sistema de produção de software, procura-se atingir os pontos desse processo onde as deficiências da tecnologia de produção resultam em maiores despesas, o que não necessariamente coincide com os pontos onde elas sejam maiores sob o ponto de vista técnico.

O fato desta análise indicar onde agir prioritariamente, e não como agir, pode ser visto como um dos pontos fortes da metodologia apresentada. Como não está atrelada a nenhuma tecnologia específica, ela permite comparar os valores observados antes e depois da introdução de novos métodos, técnicas e ferramentas, de forma a verificar sua efetividade com relação a resolução dos problemas analisados, manifestada através da diminuição dos custos da qualidade.

#### **4.4. Identificação das Causas Básicas dos Problemas**

Uma vez que tenham sido determinados os pontos onde a gerência deverá concentrar seus esforços no sentido de eliminar as principais deficiências de qualidade, devem ser averiguados os fatores responsáveis pela sua ocorrência, de modo a possibilitar sua neutralização.

O objetivo a ser perseguido deve ser a remoção da causa básica dos problemas verificados, e não dos seus sintomas [10]. Nem sempre ela consegue ser facilmente detectada. Nestes casos, torna-se necessário aprofundar o conhecimento sobre todas as variáveis que intervêm sobre o processo, de forma a possibilitar a análise da sua influência sobre o resultado final. Com isto, pode-se obter indícios suficientes sobre o ponto onde tem início as dificuldades observadas, possibilitando a sua remoção ou controle.

Nem sempre a neutralização total da causa básica é possível, pois muitas vezes implica em mudanças revolucionárias na tecnologia de produção, cujo conhecimento necessário não se encontra disponível. Um exemplo deste fato é a necessidade de realização dos testes de programas, que apenas conseguem provar a existência de erros e não a sua ausência.

#### **4.5. Desenvolvimento de Soluções Apropriadas**

A próxima etapa da abordagem proposta consiste em criar soluções que realmente resolvam os principais problemas detectados. Como visto anteriormente, a maneira definitiva de implementá-las é através da remoção das causas fundamentais responsáveis pela sua ocorrência.

As mudanças necessárias para sua efetivação nem sempre são óbvias, fáceis de implementar e isentas de efeitos colaterais. Frequentemente são necessárias mudanças substanciais, que necessitam de planejamento adequado para serem implementadas. Nem sempre as limitações existentes conduzem a soluções técnicas únicas, sendo a condição mais comum, a existência de diversas alternativas competindo entre si quanto a sua viabilidade.

#### **4.6. Planejamento e Execução das Mudanças**

Finalmente, deve-se implementar as mudanças necessárias à remoção ou minimização dos problemas detectados. Sua operacionalização deve ser suficientemente bem planejada, de forma a não introduzir novas fontes de custos da qualidade.

Para sua execução propõe-se a realização das seguintes etapas:

- Determinação dos objetivos a serem alcançados;
- Definição dos métodos e critérios necessários para sua implementação;
- Identificação das restrições e fatores limitantes;

- Avaliação das alternativas existentes;
- Seleção da solução mais adequada; e
- Implantação da solução adotada.

## 5. RESULTADOS PRELIMINARES

A implantação da estratégia proposta foi aplicada no desenvolvimento de um sistema destinado à administração de consórcios por uma montadora de automóveis estabelecida no país. A contabilização dos custos da qualidade foi feita através da ampliação dos sistemas de apropriação de mão-de-obra e controle de configuração que já eram utilizados pela empresa, de forma que eles pudessem contabilizar principalmente a incidência de retrabalho e testes em cada uma das tarefas desenvolvidas.

A contabilização de alguns tipos de custos da qualidade foi desprezada em função das dificuldades encontradas para aplicar critérios homogêneos para as três equipes que trabalhavam no projeto, uma vez que elas estavam localizadas em três cidades distintas. Esta simplificação não acarretou grande impacto sobre os resultados obtidos, pois a incidência de retrabalho, em função principalmente das deficiências nas especificações, foi da ordem de 5 vezes o esforço previsto, de forma que as demais variáveis perderam significativamente sua importância.

### 5.1. Escopo do Sistema

Para a execução do aplicativo solicitado foi utilizada a ferramenta CASE, *Genexus*, empregada para gerar programas na linguagem Cobol 400 interagindo com o sistema gerenciador de banco de dados SQL 400. Também foram produzidas rotinas *batch* empregando a Control Language do sistema AS-400.

Já no início do projeto verificou-se que era necessário realizar uma revisão para complementação e ajustes da documentação de requisitos do projeto. Esta revisão detectou alguns problemas com a qualidade das especificações tais como funções incompletas, omitindo tanto detalhes básicos de funcionamento dos processos, quanto informações sobre o relacionamento dos processos com o modelo de dados; especificações de dados com falhas de conceituação, de padrão de nomenclatura, de regras de consistência e de integridade. O modelo físico proposto também apresentava problemas graves de normalização e as especificações das interfaces com os usuários eram imprecisas e omissas.

### 5.2. Contabilização dos Custos da Qualidade

O sistema de contabilização dos custos da qualidade permitiu um levantamento da ocorrência de solicitações de alteração dos requisitos iniciais, por etapa do ciclo de vida, de forma a localizar os pontos críticos do processo de desenvolvimento. É interessante ressaltar, que pouco mais de 3% das solicitações de alteração referiram-se a problemas que não puderam ser localizados, tendo sido creditada a sua elaboração a dificuldades que ocorreram durante a utilização. Convencionou-se denominá-las de “não alteração”.

Na Tabela 1 apresenta-se um quadro contendo um resumo das solicitações de modificação requeridas pelos vários subsistemas envolvidos, bem como, a sua distribuição pelas etapas do ciclo de produção onde elas ocorreram, de forma a melhor caracterizar os pontos onde ocorreu maior concentração de problemas.

As ocorrências observadas na fase de codificação, são conseqüência da introdução de informações incorretas na ferramenta *Genexus* utilizada para geração de código. Durante o desenvolvimento do projeto, não foram detectados erros intrínsecos na ferramenta, ou seja, o código não apresentava erros, desde que o conteúdo desejado fosse corretamente especificado.

Cod.	Subsistemas	Qtde de Solicit. Alteração	Média de Solicit.por Programa	Def. dos Req.	Proj. Lógico	Proj. Físico	Geração Código	Não alteração
HB	Entrega de Bem	236	1,2	57	129	1	43	6
HC	Contabilidade	43	0,3	0	27	10	4	2
HE	Encerram.de Grupo	115	1,1	77	22	0	14	2
HF	Financeiro	1277	3,2	171	769	194	115	17
HG	Administrar Grupo	356	1,9	117	163	14	25	35
HK	Cobrança Atrasado	395	1,2	146	157	33	57	2
HL	Administrar Cliente	118	0,7	68	40	0	5	5
HP	Planejar Vendas	117	0,2	77	24	0	10	5
HQ	Quitação	11	0,2	1	8	0	0	2
HS	Segurança	48	0,8	4	1	0	10	1
HV	Vender Contrato	305	1,7	52	197	7	28	21
	<b>Total</b>	<b>3021</b>	<b>1,1</b>	<b>770</b>	<b>1537</b>	<b>259</b>	<b>311</b>	<b>98</b>

**Tabela 1 - Distribuição das solicitações de alteração por fase do ciclo de vida.**

Determinou-se também o esforço total, representado em homens-hora alocados para desenvolvimento de cada um desses sistemas, e os custos da qualidade ocorridos em cada um deles. Na Tabela 2 esses dados são apresentados reunidos por subsistema.

Cod.	Subsistemas	Custo de Desenvolvimento	Custo da Qualidade	Incidência de Custos da Qualidade
HB	Entrega de Bem	2116	1337	63 %
HC	Contabilidade	1356	244	18 %
HE	Encerramento de Grupo	1179	651	55 %
HF	Financeiro	4365	7233	166 %
HG	Administrar Grupo	2083	2016	97 %
HK	Cobrança Atrasado	3505	2237	64 %
HL	Administrar Cliente	1753	668	38 %
HP	Planejar Vendas	6691	663	10 %
HQ	Quitação	650	62	10 %
HS	Segurança	650	272	42 %
HV	Vender Contrato	2017	1728	86 %
	<b>Total</b>	<b>26365</b>	<b>17111</b>	<b>65 %</b>

**Tabela 2 - Distribuição dos custos de desenvolvimento e da qualidade por subsistema.**

Os valores apresentados estão expressos em homens-hora alocados diretamente às tarefas de produção. Para sua conversão em valores monetários utilizou-se o valor médio calculado pela empresa na execução de projetos de grande porte. As horas despendidas na produção dos diversos subsistemas não incorporam os valores alocados para atividades indiretas, tais como gerenciamento da produção, apoio técnico, suporte técnico e atividades administrativas.

### 5.3. Considerações sobre os Custos da Qualidade Ocorridos

A situação retratada acima não é usual pois, em função de uma decisão gerencial equivocada e de uma situação contratual bastante peculiar, o desenvolvimento do sistema

avançou para as etapas seguintes, mesmo sabendo-se que os problemas existentes na fase anterior não haviam sido solucionados.

O procedimento usual seria interromper o desenvolvimento, refazer os documentos da fase de projeto e, só então, prosseguir com a execução das etapas posteriores. Provavelmente alguns problemas seriam transmitidos de uma fase para outra, como faz ver o “*modelo de amplificação de defeitos*” da IBM, mas nunca com a intensidade verificada neste caso.

O impacto destas atitudes não terem sido tomadas pode ser melhor avaliado quando se verifica que 39% dos gastos totais com mão-de-obra direta na execução do sistema foram relacionados com a incidência de custos da qualidade.

A contabilização dos custos da qualidade de software possibilitou uma forma efetiva de controle sobre as etapas da produção, permitindo detectar os pontos problemáticos do processo produtivo. Por possuir esses controles, a empresa conseguiu demonstrar claramente aos seus contratantes a causa dos desvios ocorridos ao longo do desenvolvimento do projeto e, conseqüentemente, ser ressarcida dos prejuízos que teria de arcar.

## 7. CONCLUSÕES

O problema central abordado neste trabalho, refere-se a melhoria da produtividade econômica do processo de desenvolvimento de software, ou seja, a necessidade de aumentar a quantidade de software produzida por unidade de mão-de-obra, capital e insumos necessários para sua elaboração.

Esta questão se apresenta como um dos grandes desafios tecnológicos a serem superados nos próximos anos, em virtude do aumento contínuo da demanda por software pela sociedade moderna e, principalmente, da sua magnitude econômica crescente.

Sob o ponto de vista gerencial, verifica-se que as principais deficiências observadas residem na falta de informações adequadas e fornecidas em tempo hábil, de forma a embasar decisões técnicas e administrativas que resultem no aperfeiçoamento do processo de produção de software. Por este motivo, no cerne da abordagem proposta neste trabalho encontra-se a coleta de dados sobre o processo de produção de software, pois eles constituem-se na “matéria prima” fundamental para a elaboração de análises que possibilitem à gerência nortear suas ações em busca de maior eficiência produtiva.

A estratégia apresentada visa proporcionar aos gerentes de desenvolvimento um maior conhecimento sobre quais deficiências da tecnologia do produto e de seu processo de produção causam maior prejuízo, devendo portanto merecer maior atenção gerencial. Através da sua aplicação pode-se avaliar objetivamente a magnitude dos problemas existentes com relação às deficiências de qualidade e tomar medidas preventivas e corretivas que neutralizem o efeito das falhas existentes, antes que elas se propaguem. Os benefícios decorrentes dessa abordagem são imediatos, pois tornam possível a redução dos custos, ao mesmo tempo em que se obtém ganhos de produtividade e lucratividade.

Ao utilizar a contabilização dos custos da qualidade como instrumento gerencial para subsidiar o processo de melhoria da qualidade do sistema de produção de software, procurou-se atingir os pontos desse processo onde as deficiências da tecnologia de produção ocasionam maiores despesas, o que não necessariamente coincide com os pontos onde elas sejam maiores com relação aos aspectos técnicos.

Uma análise inicial dos resultados obtidos com sua aplicação foi capaz de verificar que no caso analisado 39% dos gastos totais com mão-de-obra direta na execução do sistema, foram relacionados com a incidência de custos da qualidade.

A estratégia apresentada apresenta-se como uma nova abordagem para a questão da gerência da produção de software, pois permite integrar em um único ambiente de produção,

os aspectos relacionados com :

- Contabilidade de custos de produção;
- Contabilidade dos custos da qualidade; e
- Gerência de configuração.

Sua principal característica inovadora consiste no fato de tratar essas questões de maneira integrada, pois usualmente, as sistemáticas utilizadas para gerência da produção de software costumam abordar essas questões de forma isolada.

Este objetivo é perseguido desde os trabalhos iniciais de Bersoff, que relatam o anseio por reunir num único ambiente, as informações gerenciais e de controle técnico sobre o processo de desenvolvimento [2].

### REFERÊNCIAS BIBLIOGRÁFICAS

- [1] BASILI, Victor R., MUSA, John D. *The future engineering of software : a management perspective*. IEEE Computer, p. 90-96, 1991.
- [2] BERSOFF, Edward H. *Elements of software configuration management*. In: REIFER, Donald J. *Tutorial: Software management (3. ed.)*. Catalog N° EHO-0243-6, IEEE Computer Society, 1986.
- [3] BOEHM, Barry W. *Improving software productivity*. IEEE Computer, p. 43-57, 1987.
- [4] BRYAN, William E.; SIEGEL, Stanley G. *Software configuration management - a practical look*. In: SCHULMEYER, G.Gordon; McMANUS, James I. *Handbook of software quality assurance (2. ed.)*. New York: Van Nostrand Reinhold, 1992.
- [5] CROSBY, Philip B. *Qualidade é investimento*. Rio de Janeiro: José Olympio, 1992.
- [6] CROSBY, Philip B. *Qualidade sem lágrimas: a arte da gerência descomplicada*. Rio de Janeiro: José Olympio, 1992.
- [7] FAIRLEY, Richard E. *Software engineering concepts*. New York: McGraw-Hill, 1985.
- [8] FREEMAN, Peter. *Software perspectives: the system is the message*. Reading, Massachusetts: Addison-Wesley, 1987.
- [9] HUMPHREY, Watts S. *Managing de software process*. Reading, Massachusetts: Addison-Wesley, 1990.
- [10] ISHIKAWA, Kaoru. *Controle de qualidade total: à maneira japonesa*. Rio de Janeiro: Campus, 1993.
- [11] JONES, Capers. *Produtividade no desenvolvimento de software*. São Paulo: McGraw-Hill, 1991.
- [12] JONES, Capers. *Applied software measurement : assuring productivity and quality*. New York: McGraw-Hill, 1991.
- [13] JURAN, J. M.; GRZYNA, Frank M. *Handbook de controle da qualidade (vol.2)*. São Paulo: McGraw-Hill/Makron Books, 1991.
- [14] PAULK, Mark C. et alli. *Capability maturity model for software, version 1.1*. (CMU/SEI-93-TR-024). Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 1993.
- [15] ROBLES JR., Antonio. *Custos da qualidade: uma estratégia para a competição global*. São Paulo: Atlas, 1994.