

# **Programa de Melhoria Contínua de Gerência de Configuração de Software Apoiado por Mensurações GQM: A Experiência do Tribunal Superior do Trabalho**

CLÁUDIO FONTES FEIJÓ<sup>1</sup>  
KÁTHIA MARÇAL OLIVEIRA<sup>2</sup>  
WALCÉLIO L. MELO<sup>3</sup>

<sup>1</sup>TRIBUNAL SUPERIOR DO TRABALHO – TST  
SAS – Pça. dos Tribunais Superiores – BL. D  
70.097-900, Brasília-DF, Brasil  
claudio.feijo@tst.gov.br

<sup>2</sup>UNIVERSIDADE CATÓLICA DE BRASÍLIA – UCB  
SGAN 916 – Módulo B – Asa Norte  
70.790-160, Brasília-DF, Brasil  
kathia@ucb.br

<sup>3</sup>UNIVERSIDADE CATÓLICA DE BRASÍLIA – UCB  
SGAN 916 – Módulo B – Asa Norte  
70.790-160, Brasília-DF, Brasil  
walcelio\_melo@yahoo.com

## **Resumo**

Este artigo visa apresentar os resultados da aplicação integrada de modelos de referência e indutivos de melhoria de processos de software no departamento de tecnologia da informação do Tribunal Superior do Trabalho. Serão comentados os fatores que conduziram à utilização de modelos de referência para avaliação da maturidade, e de modelos indutivos para geração de um programa de melhoria contínua baseado em mensurações. Serão mostrados os resultados obtidos com a aplicação dessa abordagem mista, e como, através de um investimento de baixo custo, conseguiu-se criar um ambiente de melhoria contínua de processos de engenharia de software, dentro do serviço público federal brasileiro.

**PALAVRAS CHAVE:** CMM, Gestão de Configuração de Software, GQM, Melhoria e Avaliação de Processos, Mensuração de processos, SPICE.

## 1 Introdução

O Tribunal Superior do Trabalho (TST) preocupado em melhorar a qualidade de seus serviços criou um programa de melhoria contínua de processos de engenharia de software. O primeiro passo deste programa consistiu em realizar avaliações sobre a maturidade e capacidade no desenvolvimento de software. Seguido a isso, foram conduzidas análises qualitativas e quantitativas sobre a maturidade dos seus processos de software.

Com o objetivo de dar sustentabilidade a esse programa de melhoria foi desenvolvido um plano de mensuração GQM [1], de maneira que as novas práticas, as novas tarefas e as ferramentas introduzidas pudessem ter a sua eficácia avaliada continuamente.

Para tanto, optou-se por construir uma solução combinando as abordagens baseadas em modelos de referência ou descendentes (*top-down*), e indutivas ou ascendentes (*bottom-up*) de melhoria de processos. Dessa forma, foi escolhido um modelo de referência para avaliar a capacidade dos processos de software da organização e outro indutivo para melhorá-los.

Esse relato de experiência mostra na seção 2 a estratégia e as metodologias utilizadas. A seção 3 apresenta os resultados da etapa de diagnóstico. Na seção 4 e 5 são descritos os planos GQM e os resultados do processo de mensuração, respectivamente. Por fim, a seção 6 relata as lições aprendidas.

## 2 Estratégia e metodologias para a melhoria

Na Figura 1 estão ilustrados os elementos chave da estratégia global de implementação da solução de melhoria. As elipses representam os aspectos extrínsecos de suporte, a caixa contém os requisitos.

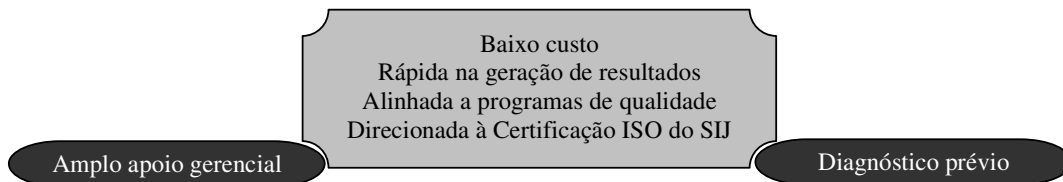


Figura 1 – Elementos da estratégia global de implementação.

O apoio gerencial estava de certa forma condicionado à implementação de um programa que não necessitasse de recursos financeiros adicionais aos já alocados pela administração do TST. Desse modo, deveria-se preferencialmente utilizar horas de trabalho dos próprios funcionários, ferramentas próprias e demais recursos já adquiridos pela unidade produtora de software (UPS).

O diagnóstico prévio partiu da auto-avaliação feita por meio do questionário de maturidade CMM [10]. Essa escolha se fundamentou pelo modelo de referência SW-CMM [7] encontrar-se consolidado, ser gratuito e disponível. Além do que, tal método permite avaliar periodicamente os efeitos das ações de melhoria.

Durante essa etapa de conhecimento pesquisou-se ainda sobre programas de qualidades implantados anteriormente, forças e fraquezas do ambiente, e relatos de defeitos em sistemas. A partir daí constatou-se as maiores necessidades, entretanto, não se sabia por onde começar a melhorar. Então, optou-se pelo método indutivo

AINSI [3], visto que ele atua diretamente na linha de produção de software (ascendente) sem se prender a um tipo específico de processo. Esse método consiste em um conjunto de diretrizes para a abordagem de melhoria contínua e progressiva definida pelo QIP [2]/GQM.

O TST encontra-se atualmente sob um programa de qualidade total que tem por objetivo certificar processos organizacionais com base na norma ISO 9001:2000 [5]. Assim, pensando a melhoria da qualidade de software também como meio de evoluir o “negócio”, foi escolhido como piloto para esse projeto o Sistema de Informações Judiciárias – (SIJ), dada a sua importância estratégica para o tribunal.

Dessa forma, a estratégia formulada considerou usuários, desenvolvedores, gerentes e metas organizacionais para construir uma solução de melhoria de software integrada aos níveis operacional, tático e estratégico da organização.

### 3 Resultados da etapa de conhecimento (diagnóstico)

Visando caracterizar o ambiente organizacional de produção de software do TST, foi feito um levantamento das atividades desempenhadas (Tabela 1).

Tabela 1 – Distribuição percentual de atividades de engenharia de software.

Atividade	Realização (%)	Acumulado
Projeto	35	-
Testes de integração	16	51 %
Melhoria de processo	16	67 %
Testes de codificação	14	81 %
Análise de requisitos	9	90 %
Garantia de qualidade	5	95 %
Gerência de configuração	5	100 %

Por meio de auto-avaliação, constatou-se que o TST está no nível 1 do CMM, pois realiza em menos de 50% cada processo chave do nível 2 (Tabela 2).

Tabela 2 – Resultado do alcance de ACPs do nível 2 de maturidade.

<i>Nível 2 – Repetível</i>		
Área-Chave de Processo	Realização processo-chave (%)	ACP Alcançada ?
Gerência de requisitos	4	Não
Planejamento de Projeto	18	Não
Controle e monitoramento de projeto	17	Não
Gerência de subcontratantes de software	12	Não
Garantia da qualidade de processo e produto	2	Não
Gerência de configuração	5	Não
Total de ACPs alcançadas		<b>0</b>
Total de ACPs não alcançadas		<b>6</b>

Para complementar esta avaliação foi utilizado o software Bootcheck [4], com “saída” pela abordagem SPICE [6]. Os resultados por essa metodologia convergiram em direção aos anteriormente obtidos pelo modelo CMM.

Depois dessas avaliações por modelos de referência CMM e SPICE, como recomendado pelos métodos indutivos, partiu-se para as análises qualitativas feitas por meio de entrevistas estruturadas aos líderes de projeto. Isso permitiu definir *baselines* de práticas de software, tal como a distribuição do esforço de trabalho entre as equipes de engenharia de software (Figura 2).

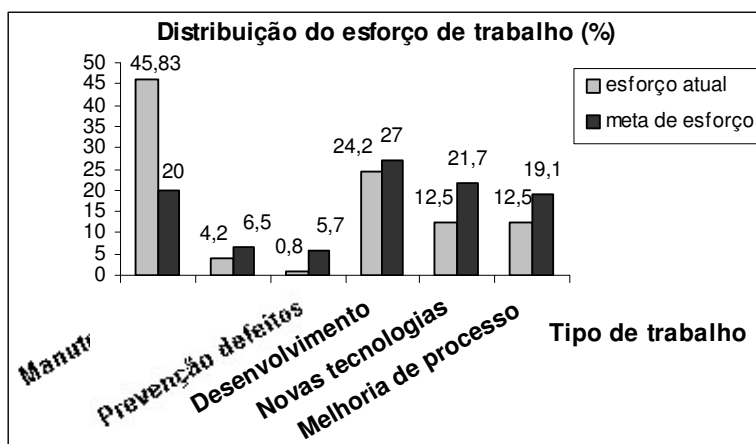


Figura 2 – Distribuição do esforço em atividades de engenharia de software.

Para complementar essa informação, como recomendado pelo GQM, listou-se as ocorrências de defeitos no SIJ registradas nos últimos 18 meses. Analisando cada uma delas, foi possível inferir as principais causas desses defeitos (Figura 3).

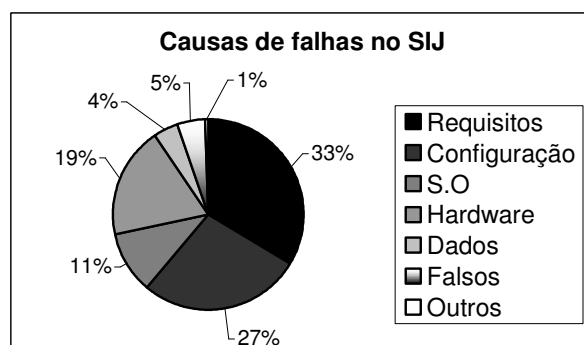


Figura 3 – Causas de falhas relatadas por usuários do SIJ.

Como se observa, a principal causa de defeitos no SIJ diz respeito a “Requisitos”, seguido por falhas de “Configuração de software”. Especificamente para esse processo, detectou-se que há falha no controle sobre a atualização de software básico, ocorre manutenção concorrente de arquivos e são disponibilizadas versões incorretas de programas no ambiente de produção.

Considerando todos esses indicadores e a caracterização do ambiente de software realizada, definiu-se como prioritário para melhoria o processo de Gestão de Configuração de Software (GCS).

Essa decisão foi reforçada devido ao fato desse processo ser requisito indispensável para o incremento da qualidade em software, segundo as principais abordagens de avaliação e melhoria de processos. E ainda, possui baixo custo de implementação, variando entre 2 e 4% do valor total do projeto do sistema e de 6 a 8% do gasto com desenvolvimento [8].

Por outro lado, a fase do ciclo de vida em que se encontra o sistema piloto pesou para essa escolha, pois, embora “Requisitos” apareça com 33%, estando o SIJ implantado há 5 anos e sob manutenção intensa, melhorar o processo de GCS seria mais apropriado do que o de requisitos.

## 4 Plano de Mensuração Goal/Question/Metric

A escolha da abordagem GQM de mensuração se justificou pela necessidade do programa de mensuração estar estritamente relacionado às metas organizacionais. Dessa forma, o objetivo de melhorar o processo de GCS deveria também aumentar a satisfação do usuário em relação a pedidos de mudança.

Assim foram formalizadas duas metas para o programa de mensuração. Uma enfatizava a melhoria do processo em si e a outra permitia avaliar a satisfação do usuário submetido ao novo processo. Ao todo foram definidas 17 questões e 25 métricas, mas devido a limitações de espaço só será possível reproduzir aqui parte delas (Figura 4). Convém ressaltar que a definição desse plano foi consequência do processo de diagnóstico da organização, descrito na seção anterior.

<p>G1. <b>Analisar</b> o processo de Gerência de Configuração, <b>com o propósito</b> de melhorá-lo, <b>com respeito</b> ao controle de versões e de mudanças, <b>do ponto de vista</b> da Gerência de Projetos de Software, <b>no contexto</b> da unidade produtora de software do TST.</p> <p><b>Q13. Há registros de manutenções simultâneas de código?</b> M20. N° de tentativas de manutenção simultânea/concorrente</p> <p><b>Q14. Recorre-se ao backup para recuperar código-fonte perdido ou corrompido?</b> M21. N° de solicitações de backup de módulos fonte</p>
<p>G2. <b>Analisar</b> a satisfação do usuário, <b>com o propósito</b> de aumentá-la, <b>com respeito</b> aos pedidos de manutenção, <b>do ponto de vista</b> da Gerência de Projetos de Software, <b>no contexto</b> da unidade produtora de software do TST.</p> <p><b>Q1. A capacidade de atendimento aos pedidos de manutenção tem aumentado?</b> M1. N° de pedidos de manutenção pendentes M2. N° de pedidos de manutenção em serviço M3. N° de pedidos de manutenção encerrados</p> <p><b>Q2. O tempo de atendimento a um pedido de manutenção tem diminuído?</b> M4. Tempo de permanência de um pedido na fila de "pendentes" M5. Tempo de permanência de um pedido na fila de "em serviço"</p>

Figura 4 – Plano de mensuração Goal/Question/Metric (parcial).

## 5 Resultados do processo de mensuração

Ao todo foram projetados 18 elementos de análise e interpretação GQM. A partir desses gráficos e tabelas será possível definir novas ações e corrigir desvios indesejados na busca pela melhoria contínua dos processos de software do TST.

### 5.1. Coleta de dados

Desde o início buscou-se fazer mensurações por meio da coleta automatizada de métricas, dispensando o uso de formulários em papel. Com efeito, optou-se pela abordagem de formulários eletrônicos e pela coleta semi-automática de métricas [9]. Dessa forma, os desenvolvedores puderam realizar as atividades dos novos processos, utilizar as ferramentas de apoio e coletar métricas de forma transparente, sem o inconveniente de preencher formulários em papel.

### 5.2. Capacidade de atendimento aos pedidos de manutenção

A partir desse elemento de informação, tomando-se por base a questão Q1 e as métricas M1, M2 e M3, além de testar a eficiência do novo processo de GCS, é possível projetar a capacidade de atendimento aos pedidos de manutenção (Figura 5), inferindo o grau de satisfação do usuário.

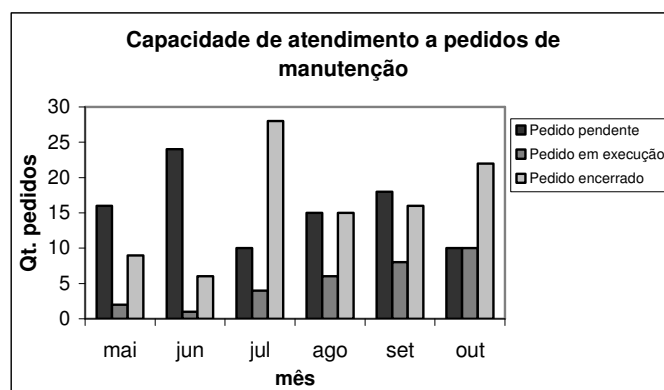


Figura 5 – Capacidade de atendimento a pedidos de manutenção.

### 5.3. Manutenções concorrentes

A eliminação de manutenções concorrentes, a partir de junho (Tabela 3), foi consequência da execução do novo processo de controle de versão e da ação efetiva dos mecanismos de *Checkin*, *Checkout* e *Locking*, presentes na ferramenta de apoio utilizada.

Tabela 3 – Número de atualizações simultâneas.

M20 Atualização Simultânea	Processo de GC antigo Jan/1997 à abr/2002	Meta (mensal)	Mês					
			Mai	Jun	Jul	Ago	Set	Out
Quantidade	8 por mês (média)	0	1	-	-	-	-	-

### 5.4. Reincidência de defeitos corrigidos anteriormente

Os resultados apresentados na Tabela 4 demonstram que as modificações nos controles de mudança e de versões, introduzidas pelo novo processo de GCS, surtiram os efeitos esperados a partir de julho.

Tabela 4 – Reaparecimento de defeitos de software já corrigidos.

M22 Reincidência de defeito corrigido	Processo de GC antigo Jan/1997 à abr/2002	Meta (mensal)	Mês					
			Mai	Jun	Jul	Ago	Set	Out
Quantidade	4 por mês (média)	0	1	2	-	-	-	-

## 6 Lições aprendidas

### 6.1. Ponto de vista tecnológico

A implantação de um novo processo de GCS não é trivial, considerando a complexidade de modelar processos de software em uma organização que não disponha de processos documentados e repetíveis. Some-se a isso, a dificuldade para integrar novos procedimentos e ferramentas de suporte com o ambiente corrente.

O programa de mensuração GQM é eficiente, mas requer planejamento e um número razoável de horas para implementá-lo. Também é necessário algum tempo para que várias pessoas possam dominá-lo e aplicá-lo corretamente. Na Tabela 5 são apresentados, em termos de pessoas e horas, a estimativa e o custo efetivo desse programa.

Tabela 5 – Custo de implementação do GQM.

Atividade	Pessoas		Horas	
	previstas	utilizadas	previstas	utilizadas
1- Preparar o programa de mensuração	6	1	120	480
2- Identificar e definir metas GQM	6	1	40	192
3- Preparar e conduzir entrevistas GQM	2	1	48	20
4- Desenvolver um Plano GQM	1	1	80	240
5- Desenvolver um Plano de Mensuração	1	1	72	185
6- Coletar dados	20	4	8	2
7- Interpretar os dados	7	2	24	16

Destaque-se que as atividades 3, 6 e 7 consumiram menos horas do que as previstas inicialmente. Acredita-se que esse fato decorreu do elevado grau de automatização conseguido para o processo de coleta de métricas, bem como, do grande detalhamento obtido para o Plano de Análise e Interpretação.

## 6.2. Ponto de vista organizacional

A direção de informática sensibilizou-se quanto a necessidade de designar servidores exclusivamente para o desenvolvimento desse programa. Um fator que pode ter contribuído para essa decisão é que esse tipo de consultoria, quando fornecida por pessoal externo, requer considerável soma de recursos e não garante melhores resultados.

Implantar programas de melhoria apoiado por mensurações constitui-se num verdadeiro antídoto contra as resistências internas que surgem dessas iniciativas. Nesse sentido, é mais fácil convencer as pessoas a seguirem rigidamente os novos processos se elas estiverem cientes de que a mensuração é instrumento capaz de confirmar ou extinguir uma nova prática de software.

Essas iniciativas, quando desenvolvidas de forma participativa e democrática, têm suas chances de sucesso aumentadas. Deve prevalecer a idéia de que a conscientização para as novas rotinas de trabalho, é diretamente proporcional ao grau de participação que tiveram as equipes na elaboração dos novos processos de trabalho.

## Referências

- [1] Basili, V. R.; Caldiera I. G.; Rombach H. D. *Goal Question Metric Paradigm*. Encyclopedia of Software Engineering (Marciniak, J.J., editor), Vol. 1, John Wiley & Sons, 1994.
- [2] Basili, V. R. *The Maturing of the Quality Improvement Paradigm in the SEL*. Presentation, Nokia Research Centre: Software Engineering Workshops, Helsinki, Finland, 1994.
- [3] Briand, L. ; El Eman, K. & Melo, W. L. *An inductive method for software process improvement: concrete steps and guidelines*. Los Alamos, California, in: elements of software process assesment and improvement. IEEE Computer Society, 1999.
- [4] European Software Institute. *Bootcheck Self Assessment Tool*. Ver. 0.5. Bootstrap Institute, 1999.
- [5] ISO 9001. *Sistemas de Gestão da Qualidade - Requisitos*. ABNT - Associação Brasileira de Normas Técnicas, 2000.
- [6] ISO/IEC TR 15504-1. *Information technology -- Software process assessment -- Part 1: Concepts and introductory guide*. ISO, 1998.
- [7] Paulk M. C. *et al. Capability Maturity Model for Software*. Version 1.1 (CMU/SEI-93-TR-24, ADA 263 403). Pittsburgh, Pa. : Software Engineering Institute, Carnegie Mellon University, 1993.
- [8] Samaras, T. T. *Configuration Management Deskbook: A Modern Approach for Assuring That Products Meet Customer Requirements*. 1988.
- [9] Solingen, R. van & Berghout, E. *The Goal/Question/Metric Method: a practical guide for quality improvement of software development*. McGraw-Hill Publishing Company, 1999.
- [10] Zubrow, D. ; Hayes, W. ; Siegel, J. & Goldenson, D. *Maturity Questionnaire*. Special Report, CMU/SEI-94-SR-7, 1994.