

# Uma proposta de organização e funcionamento da função de Garantia da Qualidade de Software em um contexto de implantação do SW-CMM<sup>i</sup>

Sabrina Marczak<sup>1</sup>, Luciana Sá<sup>1</sup>, Ilmari Ceccato<sup>2</sup>, Jorge Audy<sup>1</sup>, Dante Antunes<sup>2</sup>

<sup>1</sup>Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS  
Faculdade de Informática – Programa de Pós-Graduação em Ciência da Computação  
Av. Ipiranga, 6681 – Prédio 16 – Sala 106 – Porto Alegre, RS – CEP 90619-900  
{smarczak; audy}@inf.pucrs.br, puc\_luciana\_sa@dell.com

<sup>2</sup>*Brazil Global Development Center*  
Av. Ipiranga, 6681 – Prédio 95A – Porto Alegre, RS – CEP 90619-900  
{ilmari\_ceccato, dante\_antunes}@dell.com

## Resumo

A busca pela sobrevivência em mercados cada vez mais globalizados tem tornado a melhoria da qualidade uma necessidade evidente. Neste contexto, tem-se modelos de maturidade de processo como o CMM. Para a área de software, em específico, tem-se o SW-CMM, que é organizado em níveis, compostos por áreas-chave. Uma das áreas-chave do nível 2 trata a questão da garantia da qualidade de software (*Software Quality Assurance key practice area*). O objetivo deste artigo é apresentar uma proposta de organização e funcionamento da função de garantia da qualidade de software, visando atender os requisitos do SW-CMM nível 2. Este estudo pode ser caracterizado como exploratório, baseando-se em uma revisão teórica e nas lições aprendidas em um estudo de caso desenvolvido no *Brazil Global Development Center* (GDC) da *Dell Computer Corporation*. A proposta descrita neste artigo foi aplicada no GDC, recentemente avaliado e reconhecido como uma organização SW-CMM nível 2.

**Palavras-chave** – Qualidade de software, SW-CMM, garantia da qualidade, função de garantia da qualidade

## Abstract

*The search for survival in ever more globalized market has done the need for quality improvement evident. In this context, there are models for process maturity, such as CMM. To the software area, specifically, there is SW-CMM, organized in levels and composed by key areas. One of the key areas for level 2 deals with the software quality issue (Software Quality Assurance). This article objective is to present an organizational proposal and the software quality assurance activities, trying to achieve the SW-CMM level 2. This study can be characterized as exploratory, being based on a theoretical revision and the lessons learned in a case study developed in the Brazil Global Development Center (GDC) of the Dell Computer Corporation. The proposal described in this article was applied in the GDC, recently assessed and recognized as an SW-CMM level 2 organization.*

**Keywords** – Software quality, SW-CMM, quality assurance, quality assurance function

---

<sup>i</sup> Pesquisa realizada pelo Centro de Pesquisa em Processo e Qualidade de Software Dell/PUCRS, financiada pelos recursos da Lei de Informática (Lei Federal Brasileira nº 8.248/91). Este Centro está vinculado ao Programa de Pós-Graduação em Ciência da Computação da Faculdade de Informática/PUCRS.

## 1. Introdução

A qualidade tornou-se, nos últimos anos, um dos principais fatores de competição em todas as áreas de atividade econômica. A busca pela sobrevivência em mercados cada vez mais globalizados tem tornado a qualidade dos produtos uma necessidade evidente. Na área de software a realidade é a mesma. Além de ter-se como necessidade a melhoria da qualidade do produto final, resultante do processo de desenvolvimento de software, as empresas têm se preocupado em melhorar o próprio processo de desenvolvimento como forma de buscar garantir a qualidade do produto em si.

Neste contexto, surgem modelos de qualidade, como, por exemplo, o *Capability Maturity Model* (CMM), usados pelas organizações como guias para definir seus processos de software e a maturidade dos mesmos, tanto como para orientar um trabalho de melhoria destes processos. Mais especificamente, tem-se a função de Garantia da Qualidade de Software (*Software Quality Assurance – SQA*) com o objetivo de coletar e avaliar evidências para determinar o nível de aderência aos padrões e processos definidos na área de software.

O objetivo deste artigo é apresentar uma proposta de organização e funcionamento da função de SQA, visando atender os requisitos do SW-CMM nível 2. Este estudo pode ser caracterizado como exploratório, baseando-se em uma revisão teórica e nas lições aprendidas em um estudo de caso desenvolvido no *Brazil Global Development Center* da *Dell Computer Corporation*, doravante denominado GDC. A proposta descrita neste artigo foi aplicada com sucesso no GDC, recentemente reconhecido como uma organização SW-CMM nível 2.

O artigo está organizado em 8 seções. A seção 2 apresenta o conceito de qualidade de software e o modelo de maturidade de processo CMM. Na seção 3 discorre-se sobre a função de SQA. A seção 4 descreve o método de pesquisa utilizado para acompanhar a implantação da proposta de organização e funcionamento da função de SQA no contexto do SW-CMM nível 2 e a seção 5 apresenta a proposta em si. Na seção 6 é relatada a implantação da proposta na organização estudada, bem como também é apresentada uma breve descrição sobre a organização. A seção 7 apresenta as considerações finais. As referências bibliográficas são apresentadas na seção 8.

## 2. Qualidade de software

O conceito de qualidade é bastante antigo. Classicamente, a noção de qualidade tem sido a de que o produto desenvolvido deve cumprir com sua especificação [1]. Para Deming [2], um dos grandes líderes do gerenciamento da qualidade, a qualidade é definida como “Um grau previsível de uniformidade e dependência, baixo custo e satisfação no mercado”. Ou seja, qualidade é sempre aquilo que o cliente necessita e quer. O conceito de qualidade tem evoluído ao longo do tempo, acompanhando os novos conceitos da área de Administração. Este conceito foi incorporado a diversas áreas, como a de software.

Fisher e Light [3] definem qualidade de software como um conjunto de atributos que descrevem o grau de excelência de um sistema computacional. A ISO (*Internacional Standard Organization*) [4] define como “A totalidade de características que influenciam a habilidade de satisfazer necessidades explícitas ou implícitas para o desenvolvimento de um software”. Yourdon [5] entende como “Ter um sistema que funcione, faça exatamente o que o cliente espera, fique pronto no prazo, mereça confiança e possa ser modificado e mantido”. Pressman [6] define como “Conformidade com os requisitos funcionais e de performance explicitamente especificados, padrões de desenvolvimento explicitamente documentados e

características implícitas que são esperadas por qualquer software desenvolvido profissionalmente”.

Ainda, existem outros autores que defendem que a qualidade de software pode ser vista sobre duas diferentes abordagens, a qualidade dos produtos de software e a qualidade dos processos de desenvolvimento e manutenção dos mesmos [7]. Geralmente a primeira é caracterizada pela realização de testes e a segunda, de auditorias.

A ênfase na abordagem da qualidade dos produtos de software se dá, principalmente, em função das constantes mudanças de tecnologias e recursos que os profissionais da área de software vêm enfrentando nos últimos tempos. Estas constantes mudanças acabam não permitindo um pleno domínio sobre as novas tecnologias, em função da rapidez que devem ser estudadas e tornarem-se conhecidas e, como consequência, é preciso dar-se mais atenção à prevenção de erros no processo de desenvolvimento [8]. Desta forma, Kayani [9] e Molinari [8] acreditam que a qualidade de software é, antes de tudo, a realização de testes com o objetivo de prevenir erros.

Em contra-partida, a preocupação com a qualidade do processo de software tornou-se tão real e importante quanto a qualidade dos produtos entregues ao cliente. Isto se deve em função da importância estratégica que os softwares vêm assumindo na vida das organizações. Partindo do pressuposto que a qualidade dos produtos de software está diretamente relacionada com a qualidade do seu processo de desenvolvimento, muitas empresas passaram a investir na melhoria deste processo [10, 11]. Neste contexto, surgem modelos de maturidade, como, por exemplo, o CMM [12] e o SPICE [13]. Estes modelos são usados pelas organizações como guias para examinar as práticas dos seus processos de software e continuamente melhorá-las, buscando atender os padrões especificados [14].

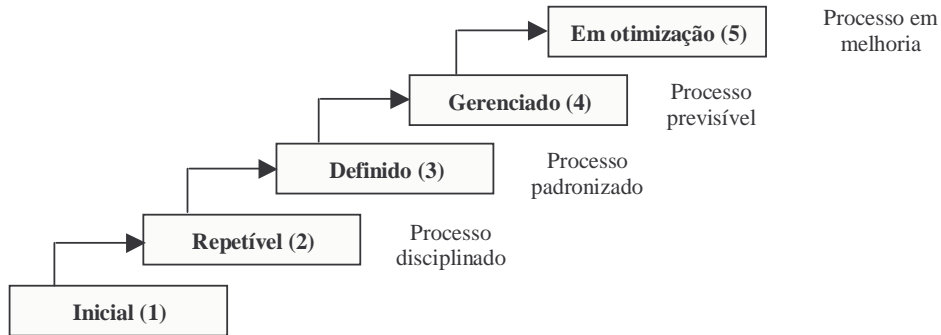
### *2.1 Capability Maturity Model*

O CMM foi criado pelo *Software Engineering Institute* – SEI em conjunto com o Departamento de Defesa dos Estados Unidos (*Department of Defense* - DoD) [12] e tem sido considerado o mais consolidado e aplicável modelo de qualidade de software [6, 15]. O DoD, após observar que a situação dos seus contratos de desenvolvimento de software tornara-se insustentável, patrocinou a criação do SEI, em 1984, visando proporcionar condições para a evolução das boas práticas de Engenharia de Software. O objetivo era que fosse alcançado, nos projetos de desenvolvimento de software dos fornecedores do DoD, o mesmo nível de repetibilidade e controle encontrado em outros setores da atividade industrial, tais como a manufatura e a construção civil [15].

O CMM nasceu sob influência das teorias e conceitos de qualidade propostos por Shewhart, Deming e Juran, e está baseado na crença de que é possível estender todos esses conceitos e ferramentas da qualidade para o setor de software [15]. O modelo é construído a partir do conceito de processo, o qual é integrado por pessoas, ferramentas e métodos para executar uma seqüência de passos com o objetivo definido de transformar determinadas entradas em determinadas saídas [12]. Uma premissa implícita no modelo é que a qualidade de um produto é determinada em grande medida pela qualidade dos processos utilizados na sua produção e manutenção. Na medida em que a maturidade dos processos de uma empresa evolui, estes passam a ser mais bem definidos e institucionalizados nas organizações e com maior equilíbrio entre os seus três componentes [15].

Quanto à sua estrutura, o CMM está baseado em uma série de práticas, organizadas em cinco níveis crescentes de maturidade. Cada nível define o grau de maturidade dos processos de uma organização e é composto por várias áreas-chave de processo. Cada área-

chave permite alcançar um conjunto de metas ou objetivos. A satisfação desses objetivos é que permite dizer se a organização atingiu um determinado nível de maturidade [12]. A Figura 1 apresenta os cinco níveis de maturidade do modelo.



**Figura 1 - Níveis de maturidade do CMM**

Baseados nesta estrutura geral foram criados alguns modelos específicos para atender diferentes áreas e aspectos de uma organização, tais como a área de recursos humanos (*People CMM*), de sistemas em geral (*SE-CMM*), de software (*SW-CMM*), entre outras. Além disto, existem ainda modelos para conduzir avaliações (*SCE*, *SCAMPI*) e planejar a melhoria dos processos (*IDEAL*). O *SW-CMM* abrange práticas de planejamento, engenharia e gerência para o desenvolvimento e manutenção de software, auxiliando uma organização a transformar seus processos de software *ad hoc* e imaturos em processos maduros e disciplinados.

O *SW-CMM* prevê que no Nível 1 a organização não tem repetibilidade dos seus processos e freqüentemente assume compromissos que não pode cumprir, seja em prazo ou em custo. Os processos, os procedimentos e o conhecimento pertencem às pessoas e não ao projeto de software. No Nível 2 o planejamento de novos projetos é baseado na experiência anterior em projetos semelhantes, de maneira formalizada e não intuitiva. Os projetos usam processos que são definidos, documentados, usados, disseminados, medidos e fiscalizados. Os processos podem ser repetidos com resultados previsíveis. A área-chave de *SQA* é prevista neste nível. No Nível 3 os processos utilizados são estabelecidos e padronizados em toda a organização. Os processos técnicos de engenharia de software passam a ser considerados ao lado dos processos gerenciais. Neste nível, passam a ser repetíveis tanto os processos gerenciais quanto os técnicos. Esses processos pertencem agora à organização e não a uma ou mais equipes. No Nível 4 a organização estabelece metas quantitativas para os seus produtos e processos. Medidas de qualidade e produtividade são coletadas em todos os projetos e armazenadas numa base de dados. A gestão em uma organização de nível 4 utiliza dados quantitativos para suportar o processo decisório. No Nível 5 a organização como um todo está engajada na melhoria contínua de seus processos, em constante otimização. A empresa é capaz de identificar seus pontos fracos e agir de forma preventiva, antes da ocorrência de problemas [12, 15].

### 3. Função de *SQA*

Conforme o conceito de qualidade de software adotado por uma organização, esta pode definir a função de *SQA* com diferentes finalidades e objetivos. As abordagens de qualidade de software mencionadas anteriormente, qualidade de produto e qualidade de processo, geram diferentes entendimentos sobre a função de *SQA*. A abordagem centrada no

produto visa a realização de testes e a centrada no processo visa a verificação da conformidade dos produtos e artefatos de um projeto em relação aos processos, procedimento e padrões especificados.

Sapijaszko e Weening [16] afirmam que muitas empresas adotam a função de SQA apenas para a realização de testes, o que acaba por caracterizar uma equipe de teste. Estes profissionais elaboram planos de teste, especificam casos de teste e aplicam técnicas específicas para verificar se os produtos desenvolvidos atendem os requisitos especificados pelo cliente. Esta perspectiva não será adotada neste artigo, por não condizer com o que é proposto pelo SW-CMM para a função de SQA.

Conforme o SW-CMM [12], o SQA tem como propósito proporcionar à gerência uma visibilidade apropriada sobre o processo que está sendo usado pelo projeto de software e pelos produtos que estão sendo desenvolvidos. Para tal, o SQA tem como objetivos revisar e auditar produtos de software e atividades para verificar se estão de acordo com os procedimentos e padrões definidos, e informar os resultados destas revisões e auditorias aos gerentes envolvidos. Para atingir tais objetivos, o SQA deve executar as seguintes atividades:

1. Preparar um plano de qualidade para o projeto;
2. Desempenhar as atividades planejadas de acordo com o plano definido;
3. Participar no planejamento do projeto e na definição dos procedimentos e padrões a serem adotados;
4. Revisar as atividades do projeto;
5. Auditar os produtos/artefatos de software;
6. Reportar periodicamente os resultados de suas atividades aos gerentes;
7. Documentar os desvios identificados;
8. Conduzir uma revisão das suas atividades com o SQA da área do cliente, quando apropriado.

É importante destacar que, quando o SW-CMM descreve as atividades do SQA, estas podem ser desempenhadas por uma única pessoa ou um grupo. Esta pessoa ou grupo de pessoas pode fazer parte de um setor ou área específica, normalmente denominada SEPG (*Software Engineering Process Group*). Cabe ressaltar que no nível 2 do SW-CMM este grupo não precisa ser um setor definido formalmente na empresa. Este pode ter suas funções atribuídas a pessoas de diversas outras áreas dentro da organização. Também, estas pessoas podem dedicar parte do seu tempo para exercer tal função ou ter exclusividade, dependendo da política estabelecida [17].

Por atuar na organização em uma função típica de auditoria, especialista na área de qualidade de software, deve-se atentar para conceitos próprios desta função. O auditor não possui função punitiva, devendo atuar como um assessor dos níveis estratégicos da organização no sentido de alertar sobre disfunções ou não aderência a padrões estabelecidos. Neste sentido, o SQA deve adotar dois princípios típicos da Auditoria, quais sejam: a administração por confronto e a administração por exceção. A administração por confronto significa verificar a aderência aos padrões e processos estabelecidos e a administração por exceção envolve o conjunto de mecanismos que permitem selecionar e avaliar o risco inerente aos processos e padrões a serem verificados [18].

Aprofundando estas analogias com a auditoria, pode-se definir a função de SQA como o processo de coletar e avaliar evidências para determinar o nível de aderência aos padrões e processos definidos na área de software. A função de SQA pode ser caracterizada como um processo contínuo de avaliação e validação dos processos de software em uma organização.

Um terceiro princípio da Auditoria diretamente aplicado a função de SQA diz respeito à independência do auditor. O SQA atua como um auditor interno na organização. Pelas

mesmas razões que as organizações que possuem auditorias internas devem periodicamente sofrer um processo de auditoria externa, o SQA também deve ser avaliado externamente de tempos em tempos. A questão da independência também pode ser abordada sob a perspectiva de distanciamento crítico com relação ao objeto de verificação. Neste sentido, o auditor de SQA atua de forma independente com relação às equipes de desenvolvimento de software que ele verifica, reportando-se diretamente à diretoria. Desta forma, tem condições de fornecer uma visão “independente” sobre a aderência aos padrões estabelecidos.

#### 4. Método de pesquisa

Esta pesquisa caracteriza-se como exploratória, ao ter como objeto essencial à familiarização com um tópico pouco conhecido pela equipe de pesquisa [19]. O principal método de pesquisa foi o estudo de caso conforme proposto por Yin [20]. Pode-se justificar o uso de métodos qualitativos nesta pesquisa pelo fato de envolver o estudo de um processo de implantação do SW-CMM nível 2 no seu contexto real, visando documentar e analisar os resultados de uma experiência prática. Também, por envolver o acompanhamento da implantação da área de qualidade em uma organização de desenvolvimento de software. Para a coleta de dados adotou-se uma solução multimétodo – questionário, entrevistas, *survey* e análise de documentos. Esta diversidade de fontes de dados propiciou a triangulação das informações apuradas, aumentando assim a confiabilidade dos resultados. A revisão teórica e as lições aprendidas no estudo de caso desenvolvido permitiram-nos tomar conhecimento do assunto estudado.

#### 5. Proposta de organização e funcionamento da função de SQA

Tendo-se realizado um estudo sobre o SW-CMM e, mais especificamente, sobre a área-chave de SQA, apresenta-se uma proposta para a organização e funcionamento da função de SQA em organizações que estão adotando as práticas propostas no nível 2 do SW-CMM. Esta proposta é composta de um conjunto de atividades, bem como os artefatos de entrada e saída relacionados a cada uma destas atividades. Os artefatos de entrada (AE) são usados como base para a realização das atividades e os artefatos de saída (AS) são aqueles resultantes da execução das mesmas. Cada uma das atividades que compõe a proposta é descrita a seguir, destacando-se os AE e AS. Estes artefatos devem estar disponíveis em uma área de comum acesso a todos os membros da equipe de projeto, definida segundo as práticas propostas na área-chave de Gerência de Configuração de Software (*Software Configuration Management – SCM*), outra área do nível 2 do SW-CMM.

- **Atividade 1:** *Apoiar o uso e compreensão dos processos, procedimentos e templates*

O uso dos processos, procedimentos e *templates* definidos no contexto de desenvolvimento de software de uma organização nem sempre é compreendido e de fácil utilização e adaptação em projetos. Para auxiliar a compreensão e utilização dos mesmos, propõe-se que o SQA esclareça dúvidas e oriente os desenvolvedores e demais profissionais envolvidos no processo de desenvolvimento de software quanto ao uso destes. Sendo assim, serão utilizados como AE para a realização desta atividade o **conjunto de processos, procedimentos e templates** definidos no processo de software da organização. **Os mesmos compreendidos** são considerados os AS desta atividade.

O SQA pode desempenhar esta atividade em qualquer momento do processo de desenvolvimento ou quando um novo profissional for contratado, interagindo-o sobre os

processos a serem seguidos. Esta atividade assemelha-se a uma orientação<sup>1</sup>, a qual é discutida e tratada na área-chave de Programa de Treinamento (*Training Program*) do SW-CMM nível 3. Apesar disto, recomenda-se que esta atividade seja conduzida pelo SQA no contexto do nível 2, uma vez que a organização está passando por um momento de institucionalização dos processos e pode haver algum tipo de dificuldade de compreensão de sua utilização. Considera-se que o SQA deve ser o responsável por fornecer esta orientação, pois é ele quem verifica a conformidade dos artefatos e produtos de software desenvolvidos em relação aos padrões estabelecidos pela organização. Isto exige do mesmo um domínio e conhecimento sobre os processos a serem seguidos.

Esta atividade não precisa ser organizada de forma burocrática ou formalizada. Sugere-se que seja desenvolvida sempre que algum dos profissionais envolvido solicite ajuda ou manifeste dificuldade de compreensão quanto ao uso e entendimentos dos processos, procedimentos e *templates*. Desta forma, é fundamental que o SQA possua a capacidade para identificar tais dificuldades e necessidades.

- **Atividade 2: Apoiar o planejamento do projeto**

O SQA deve apoiar o planejamento do projeto visando orientar e revisar se os padrões e requisitos estabelecidos pelo cliente, e os procedimentos e *templates* definidos estão sendo seguidos, bem como se está havendo respeito às políticas de software estabelecidas. Esta atividade deve ser realizada com o gerente do projeto (e algum representante da equipe responsável pelas especificações técnicas, caso exista esta função na equipe de projeto) assim que uma primeira versão do plano do projeto for elaborada. Ou seja, será utilizada a **versão preliminar do Plano do Projeto (PP)** como AE para esta atividade e, como saída, ter-se-á o **mesmo artefato revisado**. Isto permitirá que possíveis desvios e/ou interpretações errôneas quanto ao uso dos *templates* sejam identificadas em um estágio inicial.

- **Atividade 3: Planejar as atividades de qualidade para o projeto**

Tão logo inicie a fase de planejamento do projeto e já exista uma versão preliminar do PP, no qual estarão previamente definidas as atividades a serem desenvolvidas e os profissionais que irão compor a equipe de projeto, o SQA deve planejar quais atividades irá realizar no projeto. Este planejamento deve estar documentado em um Plano de Qualidade de Software (PQS), abordando os seguintes aspectos em relação ao projeto: recursos humanos a serem alocados e suas respectivas responsabilidades; recursos computacionais necessários; processos, procedimentos e padrões a serem utilizados como referência, produtos de software e artefatos que serão auditados; métricas que serão coletadas; cronograma previsto para realização das revisões (vide atividade 4) e auditorias (vide atividade 5), bem como a periodicidade, a forma e para quem os resultados (desvios) serão reportados e avaliados; e artefatos que deverão ser elaborados (relatórios, *checklists*, etc). Salienta-se a necessidade de observar as datas definidas no PP para definição do cronograma das atividades do SQA. Estas datas devem estar em concordância com as datas definidas para o projeto.

O SQA irá elaborar este plano baseado nos artefatos já existentes do projeto (AE), tais como o **PP** e a **definição de padrões pelo cliente**, caso esta definição exista. Depois de elaborado o **PQS (AS)**, este deve ser revisado e aprovado pelo gerente do projeto, para que possam ser discutidas as divergências.

---

<sup>1</sup> A área-chave de Programa de Treinamento (*Training Program*) utiliza o conceito de treinamento sob dois aspectos: 1) um informal, que ocorre no desenvolver de um trabalho, chamado de *orientação*; 2) e outro formal, organizado em formato de aulas ou estudo orientado, chamado de *treinamento*. Esta diferenciação é feita para indicar o nível de formalismo envolvido, mas ambos possuem como objetivo instruir e ensinar o grupo de profissionais da organização. Maiores detalhes em [12].

- **Atividade 4: Revisar as atividades do projeto**

Em conjunto com o gerente de projeto ou um representante da equipe de projeto, o SQA deve revisar se as atividades do projeto estão sendo conduzidas conforme os procedimentos e padrões estabelecidos. Ou seja, revisar se as atividades definidas para o projeto estão de acordo com as atividades previstas no processo de software e estão sendo realizadas. Esta atividade pode começar a ser desempenhada tão logo se iniciem as atividades do projeto, na fase de planejamento, e sua periodicidade deve respeitar o cronograma definido no PQS.

Para orientar a revisão das atividades, o SQA deve usar o PQS e o PP, onde estão definidos os padrões a serem seguidos e as atividades a serem realizadas. Também, sugere-se que seja usado como referência um *checklist*, descrevendo todos os itens importantes a serem observados e revisados. Caso não exista um *template* previamente definido, o SQA pode elaborá-lo, bem como no caso de já existir um *template*, este pode ser adaptado para atender as particularidades do projeto. Sendo assim, o SQA estará usando como AE o **PQS**, o **PP** e o **checklist de atividades** e, como AS, terá o **checklist preenchido**.

O *checklist* de atividades também tem como objetivo permitir que os desvios identificados sejam anotados, para futuramente serem descritos em um relatório a ser disponibilizado e discutido com o gerente de projeto (vide atividades 6 e 7). Caso não tenha sido definido no PQS a periodicidade para que o SQA verifique se os desvios foram solucionados, este deve acordar com o gerente do projeto uma data específica. Além disto, o SQA também pode oferecer auxílio e discutir propostas de solução com o gerente, considerando aspectos relacionados à não-conformidade em relação aos padrões estabelecidos e/ou processos e procedimentos que não foram seguidos.

- **Atividade 5: Auditar os artefatos do projeto**

Esta atividade pode ser realizada em paralelo com a atividade 4, pois não existe uma dependência entre ambas, apenas é necessário observar-se o cronograma definido no PQS do projeto, respeitando as datas definidas para os artefatos estarem prontos e poderem ser auditados. Também, a atividade só pode ser desempenhada no final da fase de planejamento, quando já houver uma versão final do PP, ou a partir do início da fase de desenvolvimento, quando os artefatos começam a ser desenvolvidos. Assim como na atividade 4, a periodicidade de realização das auditorias deve respeitar o cronograma definido no PQS.

A auditoria dos artefatos dos projetos possui o mesmo objetivo que a revisão das atividades, verificar se os artefatos estão sendo elaborados e desenvolvidos conforme os procedimentos e padrões estabelecidos para o projeto, mas porém se difere da mesma por poder ser realizada de forma independente da equipe de projeto (a outra deve ser realizada com a presença do gerente de projeto). Sugere-se que o SQA audite os artefatos utilizando como base um *checklist* especificando quais são os itens importantes a serem observados em cada artefato. Este *checklist* de artefatos deve ser adaptado para atender artefatos específicos de um projeto, para os demais previstos no processo de desenvolvimento ele deve ser padrão. Por exemplo, caso um projeto deva ter um manual do usuário e o *template* do *checklist* não preveja uma lista de itens para este tipo de artefato, o SQA poderá especificar os itens a serem observados sobre este manual no contexto deste projeto. Os desvios encontrados durante a auditoria devem ser registrados neste *checklist* e utilizados como referência na elaboração do relatório de não-conformidades a ser enviado ao gerente de projeto. Sendo assim, são usados como AE o **PQS**, o **PP** e o **checklist de artefatos**. O **checklist preenchido** será o AS.



- **Atividade 6: Documentar os desvios encontrados**

Baseado nos desvios indicados nos **checklists de atividades e artefatos** (AE), o SQA deve elaborar relatórios descrevendo estes desvios e as datas limites para estes serem corrigidos, ou seja, quando serão verificados quanto à sua correção. Sugere-se a elaboração de dois **relatórios** distintos (AS), um descrevendo os desvios relativos às atividades e outro, aos artefatos. Esta distinção poderá facilitar a verificação da correção dos desvios, uma vez que esta verificação poderá ocorrer em diferentes momentos ou realizada com diferentes pessoas, ou o gerente de projeto ou algum representante da equipe de projeto.

Os relatórios de não-conformidade devem ser elaborados conforme a periodicidade estabelecida no **PQS** (AE), geralmente a mesma definida para a realização das revisões e auditorias. Além disto, os relatórios podem ser elaborados para diferentes públicos-alvo, apresentando diferentes visões sobre as mesmas informações para cada um destes públicos. Por exemplo, caso se deseje informar o gerente organizacional sobre a quantidade de desvios encontrados, pode-se elaborar um relatório apenas indicando quantos desvios foram encontrados em relação às atividades e/ou artefatos. Também se pode usar como fonte de referência para a elaboração do relatório para o gerente organizacional as métricas coletadas (vide atividade 9). Já para os gerentes de projeto, é necessário descrever cada um destes desvios, pois o mesmo, provavelmente, será o responsável por solucioná-los ou eleger algum dos membros da equipe de projeto para tal.

- **Atividade 7: Reportar os desvios encontrados**

Uma vez elaborados os relatórios de não-conformidade, estes devem ser disponibilizados e/ou discutidos com os respectivos responsáveis, conforme acordado no PQS. O relatório a ser discutido com o gerente do projeto deve ser disponibilizado seguindo a política estabelecida e documentada no PQS. Por exemplo, a política pode estabelecer que o relatório deve ser disponibilizado na área comum ao projeto ou submetido por *e-mail*. Para aqueles relatórios a serem apresentados ao gerente organizacional, estes podem ser apenas disponibilizados, mas sugere-se que os relatórios também sejam discutidos, para que o gerente organizacional possa tomar conhecimento do andamento do projeto. Assim, tem-se como AE o **PQS** e os **relatórios de não-conformidades**, e como AS apenas os **relatórios**.

Quanto à discussão dos desvios, esta deve seguir a periodicidade estabelecida no PQS. Sugere-se realizar uma única reunião para discutir os desvios relativos às atividades e aos artefatos. Caso não seja a mesma pessoa responsável por responder ao SQA se os desvios foram ou não solucionados, pode-se realizar duas reuniões. Mas recomenda-se como o mais apropriado à realização de uma única reunião com as duas pessoas, buscando facilitar o replanejamento de datas para rever os desvios que não foram, eventualmente, corrigidos. Uma prática alternativa seria disponibilizar o relatório de não-conformidades de artefatos e, na data prevista, auditar novamente os artefatos visando verificar se os desvios foram solucionados. Esta prática não pode ser adotada para o relatório de não-conformidades de atividades, visto que estas não podem ser revisadas através de artefatos.

- **Atividade 8: Acompanhar a resolução dos desvios encontrados**

Seguindo a periodicidade estabelecida no PQS, o SQA deve reunir-se com o gerente de projeto (e mais uma representante da equipe, se necessário) para verificar se os desvios foram corrigidos. Àqueles que não foram solucionados na data acordada, devem ter as data redefinidas. Caso a não solução do desvio na data acordada implique em algum risco para o projeto, a questão deve ser discutida com o gerente organizacional. Para definir como se irá

proceder nestes casos, é necessário ter um procedimento especificando quais são as possíveis decisões a serem tomadas.

O SQA deve usar os relatórios de não-conformidades (de atividades e de artefatos) para verificar se os desvios foram corrigidos, bem como estabelecer e registrar no relatório as novas datas para os desvios serem corrigidos. Considera-se importante ter uma data para cada desvio encontrado, pois estes podem ter impactos distintos no projeto, possibilitando a determinação de diferentes datas. Logo, tem-se como AE os **relatórios de não-conformidades** usados como referência para a reunião e os mesmos **relatórios com as datas atualizadas**, no caso de ter-se identificado algum desvio não solucionado, como AS.

- **Atividade 9:** *Coletar as métricas de qualidade definidas para o projeto*

A coleta de métricas é muito importante para o processo de desenvolvimento com um todo. As métricas específicas definidas para o processo de SQA irão permitir que o SQA crie uma base de dados para futuramente ser consultada, buscando identificar, por exemplo, qual desvio teve o maior índice de ocorrência. Estas métricas também poderão ser usadas para contabilizar o custo das atividades relacionadas ao processo de SQA.

As métricas devem ser definidas no PQS e coletadas pelo SQA com base nas revisões e auditorias realizadas, bem como nas demais atividades que desenvolve no projeto. Ou seja, serão utilizados como AE desta atividade o **PQS**, no qual estão definidas as métricas a serem coletadas, e os **relatórios de não-conformidades** do projeto. Como AS, tem-se um **relatório de métricas** específico para o projeto. Eventualmente o conjunto de informações destes relatórios pode ser reunido e condensado em um único relatório para ser apresentado à gerência organizacional.

- **Atividade 10:** *Revisar as atividades de qualidade do projeto com os gerentes (de projeto e organizacional)*

Esta atividade tem como objetivo proporcionar aos gerentes, de projeto e organizacional, tomar conhecimento sobre o andamento das atividades do SQA, visando verificar se os prazos e procedimentos definidos estão sendo seguidos. Estas revisões devem ocorrer periodicamente, conforme definido no **PQS**. Este plano, bem como os **relatórios de não-conformidades**, pode ser usado como base para orientar os gerentes na revisão (AE). Como resultado da reunião, pode ser gerada uma **ata** para registrar-se as decisões acordadas entre o(s) gerente(s) e o SQA ou até mesmo ter-se o **PQS atualizado** (AS), caso seja considerado necessário.

- **Atividade 11:** *Revisar as atividades de qualidade do projeto com o SQA do cliente*

Caso exista uma equipe de qualidade de software estabelecida no cliente, ou mesmo a função de SQA, devem ser realizadas pelo SQA do cliente revisões periódicas das atividades do SQA no projeto. Estas revisões devem seguir o mesmo procedimento adotado nas revisões das atividades com os gerentes (uso dos AE e AS) ou seguir o processo do cliente, desde que esta opção esteja registrada no PQS. Também deve ser seguida a periodicidade definida no PQS.

- **Atividade 12:** *Revisar as atividades e artefatos que fazem parte do processo de SQA com um especialista*

Considerando que as atividades e artefatos utilizados pelo SQA são revisados apenas pelos gerentes, os quais são membros integrantes do projeto, e eventualmente pelo SQA do cliente (caso exista), sugere-se que um especialista em qualidade de software revise estas

**atividades e artefatos**, bem como o **processo** e os **procedimentos de SQA** definidos pela empresa (AE). O especialista pode ser contratado pela empresa ou ser um profissional de outro setor da mesma, desde que este não atue no mesmo setor que o SQA e tenha o domínio do assunto. Como resultado, espera-se que o especialista apresente uma série de sugestões de mudança visando a melhoria das atividades e dos artefatos utilizados através de um **relatório de melhorias** (AS). O objetivo destas revisões é obter um parecer de um profissional externo ao ambiente de trabalho, isento do dia-a-dia da empresa. As revisões devem ser periódicas e organizadas conforme a política estabelecida pela empresa.

## 6. Implantação da proposta apresentada: o caso do GDC da Dell Computer Corporation

O grupo de pesquisa aplicou a proposta apresentada (seção 5) em uma organização de desenvolvimento de software com o objetivo de verificar seu funcionamento em uma situação real e analisar sua aplicabilidade. Desta forma, apresenta-se a seguir a organização e o processo de implantação da proposta apresentada.

### 6.1 Caracterização da organização

O GDC é uma unidade de desenvolvimento de software *offshore* da Dell Computer Corporation, localizada no Parque Tecnológico da Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) – TECNOPUC, em Porto Alegre/RS. Esta unidade foi criada em dezembro de 2000 utilizando os incentivos da Lei Federal Brasileira nº 8.248/91 com o objetivo de atender as demandas de desenvolvimento de software da Dell mundial. O GDC possui mais de 110 colaboradores trabalhando nos quase 30 projetos, desenvolvidos para atender as diversas fábricas da Dell Computer Corporation. A Figura 2 apresenta a estrutura organizacional do GDC, destacando o SEPG. Este é composto pelas subáreas de Qualidade de Software (SQA), na qual a função de SQA está vinculada; Gerência de Configuração de Software (SCM), responsável por gerenciar as atividades de configuração dos produtos dos projetos (versões de código, aplicações, artefatos, entre outros); e Escritório de Gerência de Projetos (*Project Management Office* – PMO), responsável por controlar e acompanhar o andamento dos projetos desenvolvidos (horas de trabalho, riscos, esforço, tamanho, entre outros). Além do SEPG, o GDC ainda possui as áreas de Recursos Humanos (RH), prestação de serviços e desenvolvimento de software, formada por equipes voltadas para o desenvolvimento de sistemas legados e tecnologias *Web*.

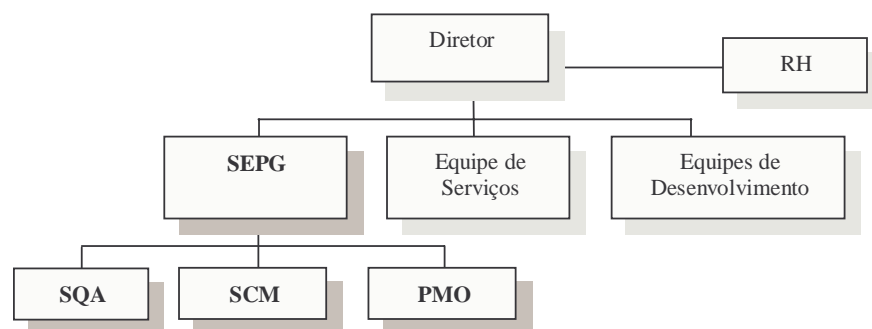


Figura 2 - Estrutura organizacional do GDC

O GDC desenvolveu e implantou um conjunto de processos baseados nas áreas-chave de processos do SW-CMM nível 2, com procedimentos escritos e detalhados. Para tal, ter implantado a função de SQA na organização foi de grande importância.

### 6.2 Implantação da função de SQA no GDC no contexto do SW-CMM nível 2

A implantação da função de SQA se deu ao longo do processo de definição e implantação dos processos de software do GDC no SW-CMM nível 2. O processo iniciou em março de 2001 e foi concluído em janeiro de 2003, após a avaliação e reconhecimento da organização como uma empresa SW-CMM nível 2 de maturidade, totalizando 23 meses. O mesmo pode ser organizado em fases, conforme pode ser observado no esquema cronológico abaixo (Figura 3). Durante estes 23 meses, é possível observar como a função de SQA foi definida e institucionalizada na organização. Também, como sua atuação tornou-se indispensável e importante no processo de implantação do SW-CMM nível 2.



**Figura 3 - Fases do processo de implantação do SW-CMM nível 2 no GDC**

Na Fase 1 (de março até agosto de 2001) foram contratados profissionais para conduzir o processo de implantação do SW-CMM na organização, o qual constituiu o SEPG. Um dos profissionais tinha como função específica assumir a criação e definição da área de qualidade no GDC. Estes profissionais fizeram diversos cursos sobre CMM, promovidos por um projeto de capacitação de empresas em modelos de qualidade de software de uma universidade da região. Este projeto também ofereceu consultorias para auxiliar as empresas na definição dos seus processos de software. Nesta fase ainda não havia sido criada a função de SQA na organização. O responsável pela área de qualidade apenas interagiu com as equipes buscando auxiliá-las na definição da documentação a ser gerada nos projetos. Durante estas interações surgiram as primeiras idéias para a elaboração dos *templates* utilizados hoje.

Na Fase 2 (de abril a setembro de 2001) os consultores do projeto com a Universidade fizeram um diagnóstico para identificar o nível de maturidade dos processos de software do GDC naquele momento. Com base nos resultados encontrados os integrantes do SEPG, juntamente com os consultores e as Equipes de Trabalho (*Process Action Teams – PATs*), formadas por gerentes de projetos e desenvolvedores, deram início a definição da primeira versão dos processos. Existia uma PAT para cada área-chave do nível 2, exceto para SQA. Em setembro de 2001 as atividades das PATs foram suspensas devido ao baixo rendimento apresentado. Desta forma, não houve nenhuma definição quanto ao processo de SQA nesta fase. No mesmo mês, encerraram-se as consultorias oferecidas pela Universidade.

Na Fase 3 (de outubro de 2001 a novembro de 2002) os processos foram disponibilizados e colocados em prática nos projetos e as atividades das PATs foram retomadas. Mesmo não tendo sido criada ainda uma PAT para a área-chave de SQA, em dezembro de 2001 o gerente do SEPG definiu uma primeira versão do processo de SQA, descrevendo as atividades e os artefatos que deveriam ser criados. Como o processo ficou muito extenso e difícil de ser utilizado, por não haver ainda a função de SQA definida no

GDC, em março de 2002 criou-se uma PAT para discutir o processo de SQA. Como não houve resultados relevantes e consideráveis, a mesma foi cancelada em maio do mesmo ano.

Em março de 2002 (Fase 3) o SEPG passou a ser uma das áreas de gerência do GDC e em junho uma nova pessoa, experiente na área de qualidade e que já havia trabalhado na implantação do nível 2 do SW-CMM, foi contratada para assumir esta área. Desta forma, o outro profissional passou a dedicar-se à gerência do SEPG e ao projeto de implantação. Também, foi contratada a filial brasileira de uma empresa de consultoria americana com larga experiência em processo de implantação CMM, com o objetivo de dar maior velocidade ao processo. Esta contratação deu-se em conjunto com um grande universidade brasileira. A partir deste momento a equipe de pesquisa passou a acompanhar o processo de implantação do SW-CMM nível 2, registrando-o e analisando-o com o objetivo de fornecer uma visão externa e isenta do andamento do mesmo. Os consultores auxiliaram o GDC a formalizar seu processo de implantação, através da elaboração de um Plano de Melhoria de Processo de Software, no qual foram previstas avaliações informais e treinamentos, e um Plano de Ação por Prática do Modelo. Em seguida, foram revisados os processos já definidos e no final de agosto de 2002 o SEPG disponibilizou a nova versão dos processos, denominada versão *alfa*. Esta foi a primeira vez que o processo de SQA tornou-se conhecido pela organização e foi colocado em prática. Neste momento o SQA passou a ser um dos profissionais mais procurados, pois os demais desejavam receber instruções de como deveriam proceder para utilizar os *templates* e seguir os procedimentos definidos. Desta forma, definiu-se uma nova atividade no processo de SQA, o apoio à compreensão dos processos existentes.

Oficialmente as atividades de SQA se iniciaram em setembro de 2002, ainda na Fase 3, através das primeiras revisões das atividades dos projetos realizadas em projetos piloto. Nas primeiras atuações do SQA este revisava se as atividades do projeto estavam sendo realizadas conforme previsto nos processos e também auditava os artefatos elaborados. As reuniões eram bastante longas e nem sempre conseguiam verificar todos os itens previstos no *checklist*. Os desvios eram reportados ao gerente do projeto através de um relatório de não-conformidades. Baseado neste relatório, o SQA acompanhava se os desvios haviam sido solucionados. No início, este acompanhamento ocorria em um prazo de tempo especificado genericamente para todos os projetos, mas conforme se foi colocando em prática a atividade, percebeu-se que seria melhor definir a periodicidade para acompanhar e revisar os desvios de acordo com as datas das demais revisões e aquelas definidas no cronograma do projeto. Também se percebeu que era necessário revisar o *checklist*, buscando apresentar os itens de uma forma mais clara e concisa. Ainda em setembro de 2002 foram elaborados os primeiros PQS para os projetos, aperfeiçoados conforme as sugestões apresentadas pelos integrantes das equipes de projeto. Conforme iam sendo realizadas as revisões das atividades dos projetos e os desvios iam sendo identificados e solucionados, o SQA passou a discutir periodicamente com os gerentes de projetos o andamento das atividades de SQA. Este se tornou o momento para os gerentes apresentarem suas opiniões sobre a atuação do SQA junto ao projeto. Além disto, também passaram a ser reportadas ao gerente organizacional o andamento das atividades de SQA. Este manifestou o interesse de tomar conhecimento da percentagem de desvios encontrados e solucionados, entre outras informações. A partir desta necessidade, criou-se um outro tipo de relatório baseado no relatório de não-conformidades de atividades. As métricas coletadas também passaram a contribuir para a elaboração destes relatórios.

No final de outubro de 2002, parte da equipe de pesquisa passou a atuar junto ao SEPG, buscando acompanhar de forma mais próxima o processo de implantação. Um dos membros da equipe dedicou-se a auxiliar o SQA, revisando a documentação gerada durante as atividades de SQA e colocando em prática as auditorias de artefatos dos projetos, realizadas

com o auxílio de um *checklist* de artefatos, criado partindo-se da mesma idéia do *checklist* das atividades. A partir deste momento os desvios identificados nas auditorias de artefatos passaram a fazer parte dos relatórios reportados aos gerentes.

A Fase 4 (junho de 2002 a janeiro de 2003) caracterizou-se pelas avaliações previstas no Plano de Melhoria de Processo de Software e pela *survey* aplicada pelo grupo de pesquisa, também considerada como uma das avaliações informais realizadas. As duas avaliações informais tiveram como objetivo verificar o *status* dos processos em relação às práticas do SW-CMM nível 2 e simular a avaliação oficial, respectivamente. Como resultado da primeira avaliação informal, em dezembro de 2002 a versão *beta* dos processos foi disponibilizada. Ainda em dezembro realizou-se uma avaliação independente das atividades e artefatos de trabalho do SQA, conduzida pelos consultores, com o objetivo de verificar a conformidade em relação ao modelo. Como resultado desta avaliação, alguns artefatos de trabalho foram reformulados e o processo de SQA foi atualizado, atendendo os diversos aspectos apontados. Por exemplo, melhorou-se o procedimento que define quais os critérios para comunicar-se ao gerente organizacional que um desvio não foi solucionado no prazo previsto. Na sua essência, o processo manteve o conjunto de atividades definidas pelo GDC. Uma última avaliação informal realizada nesta fase foi a *survey* definida pelo grupo de pesquisa. A *survey* teve como objetivo acompanhar a evolução dos conhecimentos adquiridos pelos profissionais do GDC ao longo do processo de implantação. Várias ações de treinamento e capacitação foram tomadas com base nos resultados apresentados. Neste contexto, durante a execução de suas atividades, o SQA buscava sempre identificar se as ações tomadas estavam sendo eficientes. Isto se evidenciava pelo aumento do comprometimento dos profissionais com a utilização dos processos e procedimentos propostos pela organização.

A Fase 5 envolveu a avaliação externa oficial da organização nas áreas-chave do nível 2 do SW-CMM, em janeiro de 2003. Durante uma semana, quatro projetos de software foram avaliados por meio de questionários, revisões de documentos, apresentações, entrevistas individuais e em grupo. Este material foi organizado pelos gerentes de projeto sob a orientação do SQA. Como nas cinco áreas-chave do nível 2 avaliadas foi encontrado grau zero de não-conformidade, o GDC foi reconhecido como uma organização SW-CMM nível 2.

## **7. Análise da implantação da função de SQA no GDC da Dell Computer Corporation**

A função de SQA foi institucionalizada e suas atividades foram colocadas em prática no GDC somente em setembro de 2002, 18 meses após ter-se dado início ao projeto de definição e implantação dos processos de software da empresa. Apesar disto, pode-se observar que as atividades desenvolvidas pelo grupo de SQA assemelham-se bastante àquelas sugeridas na proposta de organização e funcionamento da função de SQA apresentada. Apenas uma das atividades propostas não foi executada, aquela que sugere a revisão das atividades com o SQA do cliente (atividade 12). Esta atividade está prevista no processo do GDC, mas não foi realizada por não haver um contato mais próximo com o SQA do cliente (no caso, outras unidades da Dell Computer Corporation) até o momento. Acredita-se que deverá ocorrer a interação entre as equipes de SQA com o projeto de padronização mundial dos processos de desenvolvimento de software da empresa e, assim, esta atividade será realizada.

Como consequência de qualquer processo que é colocado em prática, o processo de SQA sofreu alterações e foi melhorado. Isto ocorreu em função das sugestões apresentadas pelos profissionais do GDC e pelos especialistas que conduziram a avaliação externa sobre as atividades e artefatos de trabalho do SQA. Estes artefatos, os quais são utilizados como apoio

para a execução das atividades, também foram melhorados no decorrer de sua utilização. Considera-se que as avaliações informais e externa dos processos em geral contribuíram consideravelmente para a estabilização e elaboração de uma versão final dos processos. Destaca-se a importância da avaliação realizada pelo especialista. Esta avaliação permitiu que o próprio SQA tomasse consciência de aspectos que não estavam sendo abordados na execução de suas atividades e que eram previstos no processo de SQA.

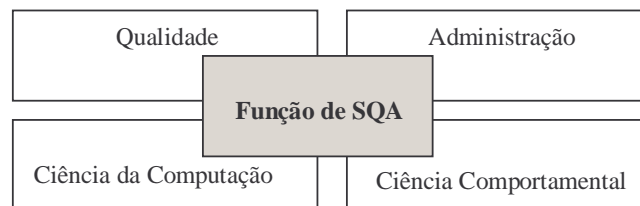
Além destas avaliações, outro fator importante e decisivo para a institucionalização da função de SQA, foi a contratação de um profissional experiente e conhecedor do modelo utilizado como referência. Este profissional adaptou-se rapidamente à empresa e pode, assim, auxiliar o gerente do SEPG na condução do projeto de implantação do SW-CMM nível 2.

## 8. Considerações finais

Considerando-se que cada vez mais as organizações vem adotando modelos de qualidade de software para buscarem garantir a qualidade de seus produtos, a função de SQA torna-se quase que indispensável, independente de ser exercida por um profissional com dedicação exclusiva ou parcial do seu tempo.

A experiência de aplicar a proposta apresentada e vivenciar as dificuldades da área de SQA no contexto de um processo de implantação do SW-CMM nível 2 mostrou que outros conhecimentos são importantes. Para constituir esta função é necessário adotar-se conceitos e princípios de diversas áreas do conhecimento, conforme pode ser observado na Figura 4. A área de Qualidade, através dos conceitos do gerenciamento da qualidade total (*Total Quality Management*) de Deming [2], deu origem ao CMM. A Administração contribui com aspectos de gestão e gerência de informações, planejamento e auditoria. A Ciência da Computação oferece suas técnicas e metodologias para a definição do processo de desenvolvimento de software, oriundas da subárea de Engenharia de Software. Por fim, a Ciência Comportamental trata aspectos de relacionamento interpessoal, fundamentais para uma comunicação e interação eficiente entre os profissionais envolvidos no processo de desenvolvimento de software e o pessoal da área de qualidade (SQA, SEPG, entre outros).

Apesar da importância reconhecida da área de qualidade de software atualmente e dos modelos de melhoria dos processos de software, faltam modelos e/ou propostas que permitam a organização e definição de funções de áreas como SQA e PMO. Este estudo avança no sentido de apresentar uma proposta de organização e funcionamento da função de SQA.



**Figura 4 - Áreas de conhecimento relacionadas**

Por ter-se aplicado a proposta em apenas uma organização, ainda não se pode chegar a conclusões mais definitivas sobre a validade da proposta. Mas esta aplicação demonstrou sua viabilidade e potencial para ser adotada por outras organizações que queiram implantar a função de SQA no contexto do SW-CMM nível 2.

## 9. Referências bibliográficas

- [1] Crosby, P., *Quality is Free*, New York, McGraw-Hill, 1979.
- [2] Deming, W., *Quality, Productivity and Competitive Position*, MIT Press, USA, 1982.
- [3] Fisher, M., Light, W., *Definitions in Software Quality Management*, New York, Petrocelli Books, 1979.
- [4] ISO 8402, *Quality – Vocabulary*, 1986.
- [5] Yourdon, E., *Declínio e queda dos analistas e dos programadores: a salvação e os novos caminhos para a produtividade e a qualidade no desenvolvimento de software*, São Paulo, Makron Books, 1995.
- [6] Pressman, R., *Software Engineering: A Practitioner's Approach*, New York, McGraw-Hill, 2001.
- [7] Sant'Ana, M.; Guerra, A., *Qualidade de Produto ou Qualidade de Processo de Software? Developers' Magazine*, São Paulo, n. 68, abr. 2002.
- [8] Molinari, L, *Os Principais Pontos da Nova Qualidade de Software*, **Developers' Magazine**, São Paulo, n. 68, abr. 2002.
- [9] Kayani, N., *Quality Assurance: QA Concepts and Implementation Guidelines*, 2000, On-line, Disponível em: <http://www.stickyminds.com>, Acessado em: 31 out. 2001.
- [10] Fuggetta, A., *Software process: a roadmap*, In: *The future of Software Engineering*, A. Funkelstein (ed), 2000.
- [11] Sommerville, I., *Engenharia de Software*, São Paulo, Addison Wesley, 2003.
- [12] Paulk, M. *et al*, *The Capability Maturity Model: Guidelines for Improving the Software Process*, USA, Addison Wesley, 1999.
- [13] ISO/IEC 15504, *Software Process Assessment*, Technical Report, 1998.
- [14] Oliveira, K.; Rocha, A.; Weber, K., *Workshop on Software Quality*, In: *International Conference on Software Engineering*, 2002, Flórida, USA, **Proceedings...** Flórida, 2002.
- [15] Côrtes, M., *Modelos de Qualidade de Software*, Campinas, UNICAMP, 2001.
- [16] Sapijaszko, G., Weening, D., *Software Quality Assurance*, Technical Report, Department of Computer Science, University of Calgary, Alberta, Canada, 1999. Disponível em: <http://sern.ucalgary.ca/~DWeening/SENG621/WebDoc/SQA-Report.html>. Acessado em: 29 out. 2001.
- [17] Schulmeyer, G., McManus, J., *Handbook of software Quality Assurance*, New Jersey, Prentice Hall, 1998.
- [18] Weber, R., *Information Systems Control and Audit*, New Jersey, Prentice Hall, 1999.
- [19] Sampieri, R. *et al*, *Metodología de La Investigación*, México, McGraw-Hill, 1998.
- [20] Yin, R., *Estudo de Caso: Planejamento e Métodos*, Porto Alegre, Bookman, 2001.