

# Gerência de Conhecimento de Qualidade de Software

Ana Candida Cruz Natali

Ricardo de Almeida Falbo

Departamento de Informática, Universidade Federal do Espírito Santo – UFES  
Av. Fernando Ferrari s/n, Centro Tecnológico  
CEP 29.060-900, Vitória – ES, Brasil.  
+55 27 3335-2167

anatali@inf.ufes.br

falbo@inf.ufes.br

## RESUMO

O desenvolvimento de software é uma atividade em expansão. Atender a essa demanda com eficiência, qualidade e menor custo requer um aprendizado evolutivo, pois um mesmo erro não deve ser cometido novamente. Uma maneira de alcançar este objetivo consiste em capturar e disponibilizar o conhecimento de uma organização de desenvolvimento de software, usando sistemas de gerência de conhecimento. Este artigo discute a importância da gerência de conhecimento no desenvolvimento de software. Mais especificamente, trata de como a gerência de conhecimento pode apoiar as atividades de planejamento e controle da qualidade.

## Palavras-chave

Gerência de Conhecimento, Qualidade de Software, Ontologia, Agentes.

## ABSTRACT

Software development is an expanding activity. Responding to its needs efficiently, with quality and low cost demands an intense learning, to prevent the repetition of the same mistake twice. One way to meet this goal consists of capturing organization's knowledge and makes it available to all their members through the use of knowledge management systems. In this paper, we discuss the importance of knowledge management in software development. Specifically, this paper discusses how knowledge management can support activities of software quality planning and software quality control.

## Keywords

Knowledge Management, Software Quality, Ontology, Agents.

## 1. INTRODUÇÃO

A demanda na área de desenvolvimento de software é crescente e traz novos desafios: melhor resposta ao mercado, grande produtividade e software com maior qualidade. O melhor atendimento a estes requisitos influencia diretamente a competitividade de organizações de software. Para atender estes objetivos, muitas organizações têm tentado obter melhor uso de seu recurso mais importante: o conhecimento.

Para ser reutilizado, o conhecimento não pode estar no nível do indivíduo. O conhecimento individual é perdido quando o indivíduo sai da empresa e não é facilmente localizado em grandes organizações. O conhecimento organizacional registrado

em papel também representa um problema, pois não pode ser facilmente acessado, compartilhado e atualizado [1]. Desta forma, o conhecimento adquirido com o desenvolvimento de software deve ser coletado e armazenado sistematicamente para que possa ser compartilhado através de uma memória organizacional. Neste contexto, sistemas de gerência de conhecimento podem ser bastante úteis.

A gerência de conhecimento envolve recursos humanos, organização e cultura, além de tecnologia de informação, métodos e ferramentas para o seu apoio [2]. A gerência de conhecimento facilita a criação, acesso e reuso do conhecimento e seu objetivo principal é promover o surgimento de conhecimento novo, seu armazenamento e compartilhamento por toda a organização.

No contexto da qualidade de software, a gerência de conhecimento pode ser utilizada para capturar o conhecimento e a experiência envolvidos no planejamento e controle da qualidade. Embora cada projeto de desenvolvimento de software seja único, experiências similares e comuns a eles podem ajudar os desenvolvedores a executarem suas atividades. O reuso deste conhecimento ajuda a evitar que falhas se repitam e auxilia na solução de problemas recorrentes. Mas, para ser efetiva a gerência de conhecimento deve ser integrada ao processo de software e suas várias tarefas, como o controle da qualidade.

Este artigo discute a gerência de conhecimento de qualidade de software e apresenta ControlQ, uma ferramenta de planejamento e controle da qualidade que apóia a gerência do conhecimento neste contexto. A seção 2 apresenta uma visão geral da gerência de conhecimento. A seção 3 discute o controle da qualidade de software e as dificuldades envolvidas nesta tarefa. A seção 4 apresenta uma proposta geral para a gerência de conhecimento, definindo requisitos essenciais. A seção 5 apresenta a ferramenta ControlQ. A seção 6 discute como a gerência de conhecimento da qualidade de software apóia as atividades de planejamento e controle da qualidade em ControlQ. A seção 7 discute trabalhos correlatos. O artigo termina com as conclusões, apresentadas na seção 8.

## 2. GERÊNCIA DE CONHECIMENTO

Antes de definir como gerenciar o conhecimento, é essencial entender o que é conhecimento. De acordo com Markkula [3], conhecimento é informação combinada com experiência, contexto, interpretação e reflexão. É uma forma de informação pronta para ser aplicada em decisões e ações.

Podemos classificar o conhecimento como sendo formal ou informal [4]. O conhecimento formal pode ser expresso de uma forma estruturada e é fácil de ser comunicado e compartilhado. Como exemplo de conhecimento formal podemos citar os artefatos de software, componentes, padrões, entre outros. O conhecimento informal é altamente pessoal e difícil de se formalizar, tornando difícil o seu compartilhamento. O conhecimento informal surge a partir das experiências individuais e envolve fatores como crença pessoal, perspectivas e valores. Exemplos deste tipo de conhecimento são as discussões e as lições aprendidas.

No contexto do desenvolvimento de software, as lições aprendidas podem ser consideradas um dos principais tipos de conhecimento. As lições são criadas como resultado do trabalho da própria organização. Devem descrever tanto os relatos de sucesso quanto oportunidades de melhoria. Lições de sucesso relatam boas respostas a situações ocorridas. Oportunidades de melhoria relatam o que ocorreu de errado em uma situação e formas potenciais para solucionar o problema [1]. O reuso de lições aprendidas em projetos passados promove boas práticas de desenvolvimento de software e previne a repetição de erros.

Uma gerência de conhecimento eficiente deve ser capaz de apoiar a criação, captura e utilização dos vários tipos de conhecimento. As atividades básicas de gerência de conhecimento incluem: identificação, captura, adaptação, integração, disseminação, uso e manutenção do conhecimento. No centro de um sistema de gerência de conhecimento está uma memória corporativa ou organizacional, apoiando o reuso e o compartilhamento do conhecimento organizacional, incluindo lições aprendidas [5].

Para apoiar as atividades envolvidas na gerência de conhecimento, várias tecnologias têm sido aplicadas como banco de dados e data warehouses, intranets e internet, máquinas de busca, entre outras [1]. Porém, duas tecnologias são particularmente importantes para a gerência de conhecimento: ontologias e agentes.

As ontologias podem ser vistas como a “cola” que mantém ligadas as atividades de gerência de conhecimento [6]. As ontologias definem um vocabulário comum utilizado pelo sistema de gerência de conhecimento e facilitam a comunicação, integração, busca, armazenamento e representação do conhecimento [1].

Agentes de software podem ser utilizados para ligar os membros de uma organização ao conhecimento disponível [1]. Agentes podem apoiar não só a busca e filtro, mas também, a disseminação do conhecimento. Se um processo de software tiver sido definido, agentes podem agir de forma pró-ativa, buscando e oferecendo itens de conhecimento que podem ser relevantes para a tarefa que o usuário está executando.

Os objetivos de uma organização determinam o tipo de conhecimento que ela deve capturar. Qualquer tipo de conhecimento pode ser utilizado para evitar retrabalho e melhorar a qualidade. Processos, modelos de qualidade, artefatos desenvolvidos, experiências e lições aprendidas são alguns exemplos de conhecimento reutilizáveis. Porém, para que o reuso de conhecimento seja eficiente em uma organização, é necessário um armazenamento adequado deste conhecimento. Por exemplo, um item de conhecimento gerado em um projeto deve ser

adaptado a futuras necessidades de futuros projetos, agregando informações que auxiliem seu reuso [7]. Estes objetivos podem ser alcançados através de um sistema de gerência de conhecimento eficiente.

### 3. QUALIDADE DE SOFTWARE

A qualidade deve ser uma característica fundamental de qualquer produto existente. Porém, desenvolver software de qualidade assegurada, dentro do prazo estabelecido e sem necessitar de mais recursos do que os alocados tem sido um grande desafio para a Engenharia de Software.

A preocupação com a qualidade de um produto de software é cada vez maior. E esta preocupação é causada principalmente pela dificuldade em se determinar meios através dos quais a qualidade possa ser planejada, avaliada e, enfim, alcançada.

No desenvolvimento de software, o controle da qualidade envolve monitorar resultados do projeto, verificando se estão de acordo com os padrões de qualidade estabelecidos. Como consequência, é possível obter a melhoria da qualidade através da identificação das causas de resultados insatisfatórios e da tomada de ações para aumentar a eficiência do projeto, provendo benefícios a todas as partes envolvidas.

Porém, assegurar o desenvolvimento de um software de qualidade não é uma tarefa simples. Primeiro é necessário identificar quais características determinam a qualidade desejada para um dado produto de software. Ou seja, primeiro é preciso fazer um planejamento da qualidade para que depois a avaliação possa ser efetuada. A qualidade de software deve ser avaliada ao longo de todo o processo de desenvolvimento de um produto. É necessário avaliar a qualidade, não apenas do produto final, mas também dos vários produtos intermediários gerados ao longo de seu desenvolvimento. Desta forma, é importante que, na definição de um processo, estabeleça-se, pelo menos em linhas gerais, como proceder para avaliar a qualidade. Procedimentos para avaliação da qualidade devem estar acoplados ao processo de desenvolvimento, definindo como serão avaliadas as características de qualidade do produto em pontos selecionados do processo.

Diante deste quadro de dificuldades, cresce, então, a necessidade de apoiar gerentes de projeto em tarefas como planejar, controlar e avaliar a qualidade do software desenvolvido. Este apoio pode ser construído a partir de duas bases complementares: o uso de ferramentas que orientem um desenvolvimento com qualidade e o reuso de conhecimento adquirido previamente na busca pela qualidade.

Uma ferramenta que se destine ao planejamento e controle da qualidade de software tem de apoiar a definição de questões essenciais, como *o que* deve ser avaliado, *quando* se avaliar, *como* avaliar e *quem* vai realizar esta avaliação. Quando estas questões estão definidas, um gerente de projeto sente-se mais seguro e guiado a desempenhar o controle da qualidade de uma forma muito mais eficiente. Porém, o papel desta ferramenta não deve se resumir a apenas automatizar certas tarefas, mas principalmente tornar o controle da qualidade uma atividade inserida no processo de desenvolvimento de software.

Outra importante forma de apoio ao controle da qualidade é o reuso de conhecimento. Já que a qualidade de software é uma

combinação complexa de fatores que variam de acordo com as diferentes aplicações e clientes que a solicitam, por que não se utilizar o conhecimento adquirido previamente nesta difícil tarefa? Vários passos do planejamento e avaliação da qualidade podem ser apoiados por experiências similares, o que auxilia o gerente de projeto a evitar erros anteriores e elimina algumas dificuldades do controle da qualidade.

Seja proveniente do uso de ferramentas ou do reuso de conhecimento, o apoio fornecido torna-se essencial para orientar que o desenvolvimento de software seja feito com qualidade, dado que a qualidade é uma característica que deve ser projetada junto com o sistema e não imposta após o fato.

#### 4. UMA PROPOSTA PARA GERÊNCIA DE CONHECIMENTO

Para apoiar as atividades que compõem a gerência de conhecimento, uma infra-estrutura deve ser provida. A memória corporativa ou organizacional deve estar localizada no centro desta infra-estrutura, apoiando o armazenamento e compartilhamento do conhecimento. Posicionados ao redor da memória organizacional, os serviços de gerência devem prover ativamente informações úteis a usuários que estejam trabalhando em atividade que exijam conhecimento [5]. Estes serviços de gerência de conhecimento correspondem a atividades como: criação, captura, recuperação, acesso, disseminação, uso e preservação do conhecimento existente na organização.

O primeiro requisito para o desenvolvimento de uma memória organizacional é promover a melhoria da acessibilidade ao conhecimento da organização, através de um repositório centralizado e bem estruturado. Já que os membros da organização estão freqüentemente muito ocupados para procurar uma informação, ou ainda, desconhecem que uma informação relevante existe, devem existir serviços pró-ativos que lembrem ativamente a estes usuários a existência de conhecimento útil. Desta forma, a distribuição do conhecimento pode ser ativa ou passiva, pois o usuário pode tomar a iniciativa de buscar uma informação ou o próprio sistema de gerência de conhecimento pode oferecer o conhecimento que julgar relevante à tarefa que o usuário está executando.

Para obter a aceitação do usuário, o sistema de gerência de conhecimento deve estar integrado ao processo da organização, permitindo a coleta e o armazenamento de conhecimento assim que eles são gerados durante o trabalho. Um sistema de gerência de conhecimento deve funcionar como um assistente para o usuário, oferecendo conhecimento organizacional relevante, mas, deixando a cargo do usuário a responsabilidade de interpretar o contexto desta informação e fazer sua avaliação. Sendo assim, para manter uma memória organizacional atualizada, é importante obter feedback de seus usuários, que são capazes de apontar deficiências e sugerir melhorias, integrando cada vez mais o sistema de gerência de conhecimento ao seu trabalho diário. Baseando-se no feedback do usuário, é possível, então, não

apenas a manutenção da memória organizacional, mas também a sua evolução.

#### 5. UMA FERRAMENTA DE APOIO AO PLANEJAMENTO E CONTROLE DA QUALIDADE DE SOFTWARE

Para apoiar o planejamento e o controle da qualidade, foi desenvolvido ControlQ, uma ferramenta integrada ao ambiente de desenvolvimento de software ODE [8]. Dentre as suas funcionalidades, podemos citar:

- a gerência de conhecimento de características de qualidade e de métricas;
- o planejamento da qualidade, permitindo a definição de atividades de avaliação da qualidade que serão executadas ao longo do projeto. Cabe ao gerente da qualidade definir para cada uma destas atividades: quando e o que será avaliado, quais características de qualidades avaliar e a partir de quais métricas estas características serão computadas;
- o controle da qualidade, permitindo o registro de resultados de medições efetuadas.

A ferramenta ControlQ foi desenvolvida baseando-se na ontologia de qualidade de software [9]. A arquitetura da ferramenta foi composta por dois pacotes: o pacote *Conhecimento*, mostrado na Figura 1, e o pacote *Controle da Qualidade*, mostrado na Figura 2.

O pacote *Conhecimento* reflete diretamente os conceitos da ontologia da qualidade de software, representando o conhecimento comum sobre este domínio. As classes deste pacote foram derivadas da ontologia de qualidade desenvolvida em [9] utilizando uma abordagem sistemática de derivação de um framework de objetos a partir de ontologias [10]. Porém, para apoiar o planejamento e controle da qualidade, são necessárias outras classes, além das classes de Conhecimento já modeladas. Para atender aos requisitos de ControlQ, foi criado o pacote *Controle da Qualidade*. As classes deste pacote representam conceitos específicos da aplicação, necessários para atender seus objetivos.

Como mostrado na Figura 2, um *plano de controle da qualidade* define todas as *atividades de avaliação da qualidade* de um projeto. Estas atividades definem não apenas *o que* será avaliado (um artefato), mas também *como* esta avaliação vai ocorrer, isto é, quais *características de qualidade* vão ser utilizadas para avaliar um artefato.

Uma *característica indiretamente mensurável* deve ser decomposta em *subcaracterísticas* para que possa ser computada pela agregação das medidas de suas subcaracterísticas. Para cada uma de suas subcaracterísticas, é necessário definir seu *peso* nesta agregação.

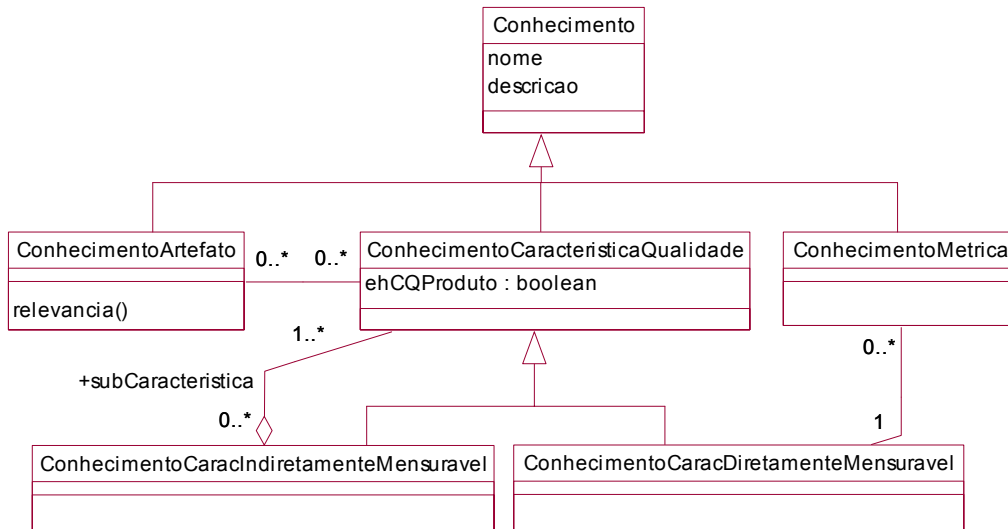


Figura 1. Parte do pacote Conhecimento.

Uma *característica diretamente mensurável* pode ser medida diretamente a partir da *métrica* escolhida para quantificá-la. Para cada *escolha*, que indica qual métrica irá avaliar uma determinada característica de qualidade, é necessário armazenar o valor correspondente desta medição.

Podemos reparar que o pacote *Controle da Qualidade* requisita serviço do pacote *Conhecimento*, e não é por acaso. Na verdade, este estilo de arquitetura em duas camadas faz com que o nível da aplicação concentre as classes utilizadas pela ferramenta e o nível de conhecimento concentre o conhecimento do domínio, que poderá ser utilizado por outras aplicações.

O planejamento da qualidade em ControlQ envolve os seguintes passos:

1. Selecionar o projeto para o qual o plano de controle da qualidade será criado e definir os artefatos de software deste projeto que serão avaliados;
2. Para cada artefato, identificar quais características de qualidade serão utilizadas para avaliá-lo;
3. Para cada característica de qualidade indiretamente mensurável escolhida, definir as subcaracterísticas nas quais ela será decomposta;

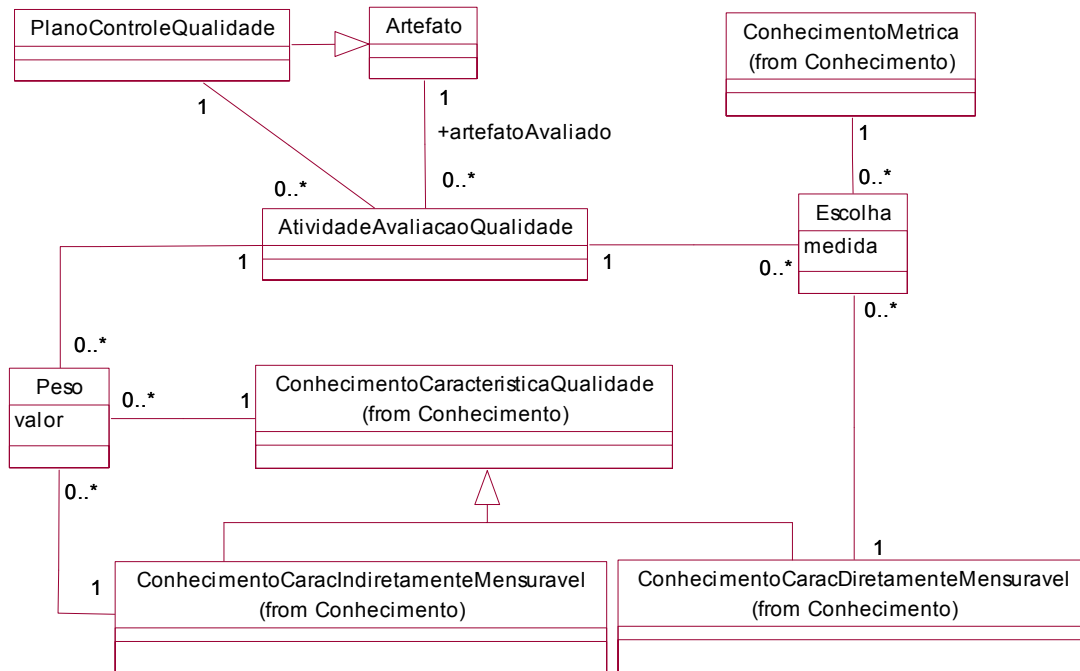


Figura 2. Parte do pacote Controle da Qualidade.

4. Para cada característica de qualidade diretamente mensurável, escolher métricas para medi-las;
5. Definir atividades de avaliação da qualidade, integrando-as ao processo de software. Para cada artefato definido no passo 1, é definido um conjunto de atividades de avaliação da qualidade.

As características de qualidade e métricas representam uma forma de conhecimento e devem ser cadastradas na ferramenta ControlQ pelo Gerente de Qualidade.

Após a definição de um plano de controle da qualidade, podemos seguir para o próximo passo: a *avaliação*, isto é, a *medição*. Na fase de medição, as métricas selecionadas devem ser aplicadas e para cada uma delas, devem ser informados valores que são registrados. A fase de medição é seguida da *apresentação de resultados*, mostrando um relato dos resultados obtidos. A análise destes resultados auxilia a definição de ações corretivas para atingir a qualidade desejada.

## 6. GERÊNCIA DE CONHECIMENTO DE QUALIDADE DE SOFTWARE

A gerência de conhecimento de qualidade de software pode auxiliar gerentes da qualidade a executarem tarefas similares do planejamento e controle de qualidade. O reuso de experiências já obtidas previamente pode evitar a repetição de erros nestas atividades.

### 6.1 Captura de Conhecimento de Qualidade de Software

A memória organizacional deve gerenciar todos os tipos de conhecimento necessários ao desenvolvimento de software. No contexto do planejamento e controle da qualidade realizado em ControlQ, há três tipos de conhecimento que são capturados e armazenados na memória organizacional: instâncias de ontologia, artefatos gerados pela ferramenta e lições aprendidas.

No contexto de ControlQ, considera-se lição aprendida tudo aquilo que foi aprendido, tanto pontos positivos ou oportunidade de melhoria, durante as atividades de planejamento e controle da qualidade. Ao lidar com lições aprendidas, é preciso considerar que o conhecimento em nível de projeto pode ser útil, mas nem sempre é o caso. Geralmente o conhecimento relativo a um projeto deve ser generalizado para se tornar um conhecimento do nível organizacional. Em ControlQ, há uma funcionalidade de aprovação de lições aprendidas. Primeiro, uma lição aprendida é cadastrada por um desenvolvedor, tipicamente um gerente de projeto ou gerente da qualidade. Neste momento este conhecimento ainda não está disponível para outros desenvolvedores. O gerente de conhecimento é o responsável por avaliar e adaptar a lição aprendida para que ela possa ser considerada um conhecimento de nível organizacional. Uma vez aprovada, a lição aprendida fica disponível como um item de conhecimento para todos na organização.

No caso de ControlQ, também se considera conhecimento as instâncias da ontologia de qualidade de software, tais como

características de qualidade e métricas usadas pela organização. O gerente de conhecimento da organização pode, então, alimentar o repositório de conhecimento com as instâncias utilizadas na organização. Uma vez criadas, estas instâncias de ontologia de qualidade de software ficam disponíveis como itens de conhecimento para reuso.

Outro tipo de conhecimento a ser capturado são os artefatos de software. No caso de ControlQ, o artefato mais importante é o plano de controle da qualidade, que define o que deve ser avaliado, a partir de quais características de qualidade e através de quais métricas estas serão medidas.

### 6.2 Busca por Conhecimento de Qualidade de Software

Em qualquer momento, um usuário de ControlQ pode buscar qualquer tipo de conhecimento: instâncias da ontologia de qualidade, artefatos (planos de controle da qualidade) e lições aprendidas.

Por exemplo, ao definir um plano de controle de qualidade, um gerente de projeto pode buscar itens de conhecimento existentes na memória organizacional relacionados a esta atividade. Esta busca é de iniciativa do usuário, já que cabe a ele definir quais as suas necessidades, ou seja, qual o tipo de conhecimento que ele deseja buscar.

Estas necessidades do usuário se tornam uma consulta à memória organizacional e os itens de conhecimento recuperados são apresentados. No contexto de ControlQ, por exemplo, um gerente de projeto pode procurar por lições aprendidas envolvendo a escolha de uma característica de qualidade para avaliar um tipo de artefato. Será, então, efetuada uma busca por lições aprendidas que relacionem a característica de qualidade e artefato especificados pelo gerente do projeto. O gerente pode, ainda, buscar outros tipos de conhecimento de qualidade de software, como instâncias da ontologia de qualidade e artefatos de ControlQ (planos de controle da qualidade).

### 6.3 Disseminação de Conhecimento de Qualidade de Software

Já que ControlQ baseia-se em uma ontologia de qualidade de software [9], instâncias da ontologia são utilizadas para apoiar atividades do planejamento da qualidade. O gerente de conhecimento é responsável por instanciar a ontologia. Estas instâncias são armazenadas na memória organizacional e são utilizadas para apoiar alguns passos do planejamento da qualidade em ControlQ como:

- Definição de quais características podem ser utilizadas para avaliar um determinado artefato (passo 2): como mostrado na Figura 3. Com base nas instâncias pré-definidas da ontologia, ControlQ apresenta apenas as características de qualidade que são consideradas adequadas para a avaliação de um determinado artefato. As características de qualidade exibidas na Figura 3 foram obtidas a partir da norma ISO 9126 e servem como um exemplo. Porém, como já ressaltado, cabe ao Gerente da Qualidade realizar este cadastro na ferramenta ControlQ;

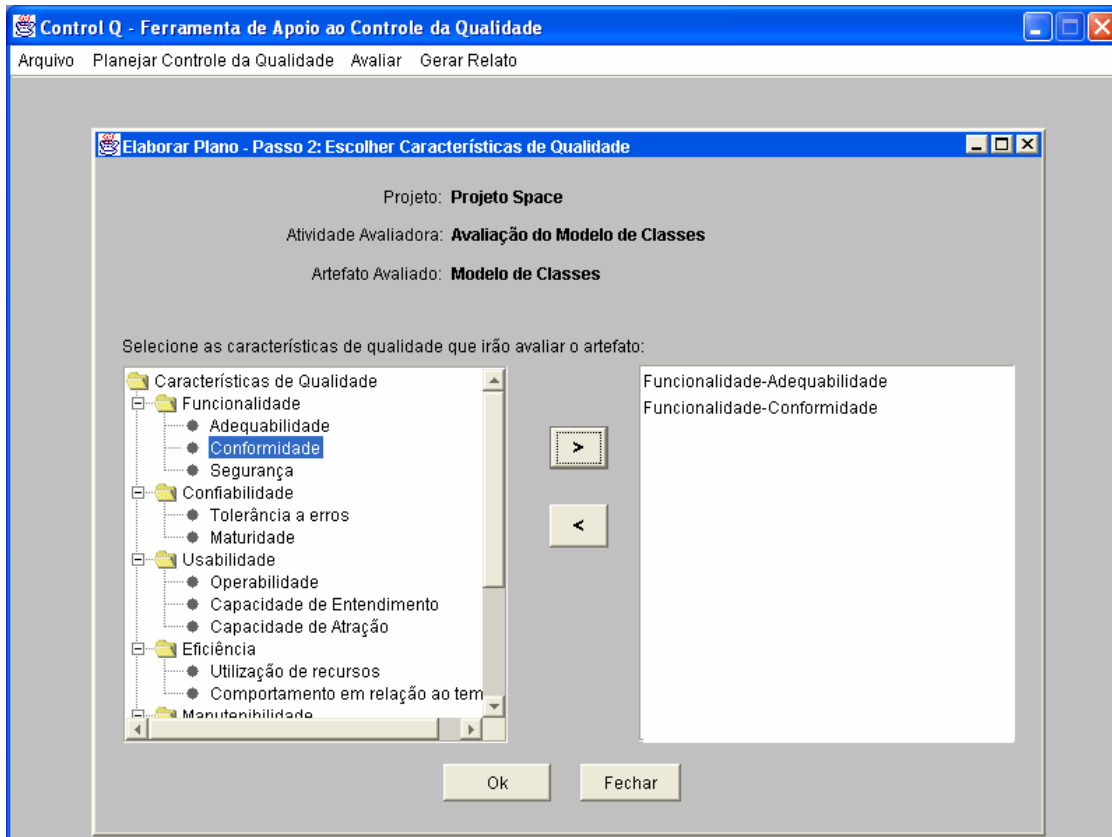


Figura 3. Definição de quais características de qualidade irão avaliar o artefato.

- Decomposição de uma característica de qualidade indiretamente mensurável em subcaracterísticas (passo 3): novamente, baseado no conhecimento existente na memória organizacional, ControlQ apresenta apenas aquelas características de qualidade que podem compor uma determinada característica indiretamente mensurável, como mostra a Figura 4.
- Definição da métrica para quantificar uma característica de qualidade diretamente mensurável (passo 4): como mostra a Figura 5, ControlQ apresenta apenas aquelas métricas que podem ser utilizadas para avaliar uma característica de qualidade específica. As métricas exibidas na Figura 5 foram obtidas a partir da norma ISO 9126 e servem como um exemplo. Porém, como já ressaltado, cabe ao Gerente da Qualidade realizar este cadastro na ferramenta ControlQ

Desta forma, o conhecimento existente na memória organizacional integra-se diretamente às atividades de planejamento e controle da qualidade, auxiliando e guiando os usuários de ControlQ.

Ainda, para cada um dos passos do planejamento descritos acima (2-4), o sistema de gestão de conhecimento da qualidade de software pode desempenhar um papel ativo na disseminação do conhecimento. Agentes de software monitoram as ações dos usuários enquanto eles utilizam ControlQ. Quando o usuário está executando um destes passos do planejamento, agentes de

software específicos agem, identificando a necessidade de conhecimento do usuário e recuperando experiências anteriores similares. Esta disseminação de conhecimento é particularmente importante quando os usuários não estão motivados a buscar uma informação ou não sabem da existência de conhecimento relevante à tarefa que estão executando.

Estes agentes disseminam lições aprendidas que relatam tanto sucessos quanto oportunidades de melhoria no planejamento e controle da qualidade. Além disso, eles disseminam outros planos de controle da qualidade já definidos e avaliados, ressaltando partes do plano relativas à tarefa que o usuário está executando. Desta forma, usuários podem tomar decisões baseadas não apenas em seu próprio conhecimento, mas baseadas, também, no conhecimento organizacional obtido em experiências similares.

Um dos agentes de ControlQ é o *Dr. QConselho*. Este agente auxilia na definição do plano de controle da qualidade fornecendo dicas em passos específicos da construção do plano. Por exemplo, no passo 2, quando é preciso escolher quais características de qualidade serão usadas para avaliar um determinado artefato, o agente *Dr. QConselho* busca por outros planos de controle da qualidade que já foram definidos e apresenta quais características de qualidade foram escolhidas em experiências anteriores para avaliar este mesmo artefato. Este conhecimento é apresentado sob a forma de uma sugestão, cabendo ao usuário da ferramenta avaliar e decidir se deve utilizar ou não a informação sugerida. A Figura 6 mostra uma sugestão do *Dr. QConselho*.

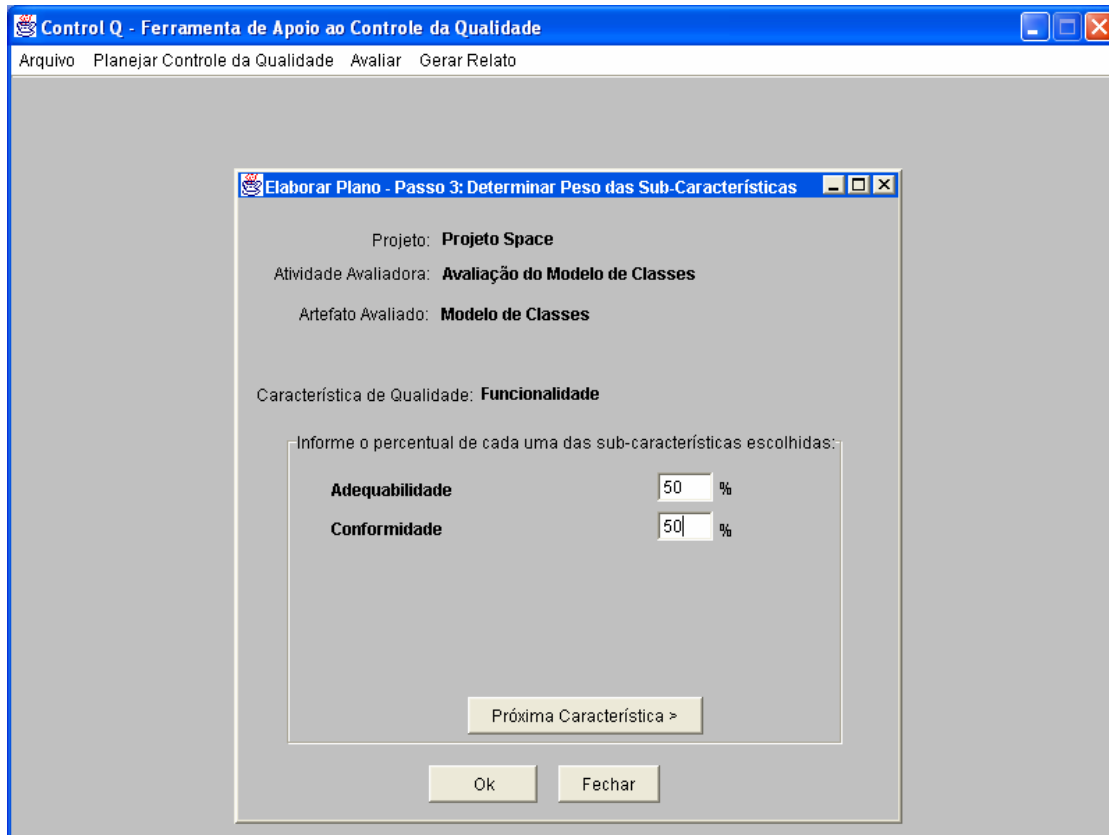


Figura 4. Decomposição de uma característica de qualidade indiretamente mensurável em subcaracterísticas.

## 6.4 Uso de Conhecimento de Qualidade de Software

Os itens de conhecimento de qualidade recuperados são apresentados ao gerente de projeto. Ele pode navegar por este conjunto de itens de conhecimento e escolher qual item deseja reutilizar. Se um plano de controle da qualidade for escolhido para reuso, ele deve identificar a parte deste plano que ele deseja utilizar em seu próprio artefato.

De um plano de controle da qualidade, um gerente de projeto pode reutilizar algumas de suas atividades de avaliação definidas ou uma escolha de características de qualidade e métricas para avaliar um determinado artefato do projeto. Se um item de conhecimento de qualidade é reutilizado, o usuário deve avaliar seu conteúdo para auxiliar a manutenção da memória organizacional. Esta avaliação inclui informações como: se o item foi útil, problemas que surgiram ao reutilizá-lo e as soluções aplicadas.

## 6.5 Manutenção de Conhecimento de Qualidade de Software

A manutenção do conhecimento de qualidade de software é realizada levando em conta o feedback do usuário. Baseado na

interação com o usuário, o gerente de conhecimento pode decidir quais itens de conhecimento estão obsoletos e quais deles nunca foram reutilizados. A partir desta identificação, o gerente de conhecimento pode decidir quais devem ser excluídos.

Esta exclusão pode ser realizada pelo próprio gerente de conhecimento ou ele pode requerer o apoio de um agente de software. Ao realizar as tarefas de manutenção da memória organizacional, o gerente possui uma interface para busca de itens de conhecimento, para exclusão de itens e para configuração de um agente de apoio. O agente de manutenção pode ser configurado para alertar o gerente de conhecimento que ele precisa realizar a manutenção da memória organizacional em intervalos de tempo definidos ou quando a memória atingir um determinado tamanho. O agente de software pode, ainda, sugerir alguns itens de conhecimento a serem excluídos, baseando-se nos critérios definidos pelo gerente de conhecimento.

## 7. TRABALHOS CORRELATOS

Muitas organizações concordam que o conhecimento é um fator crucial para o sucesso e sobrevivência em um mercado cada vez mais competitivo. Esta preocupação é uma das principais razões para o grande crescimento da pesquisa em gerência de conhecimento.

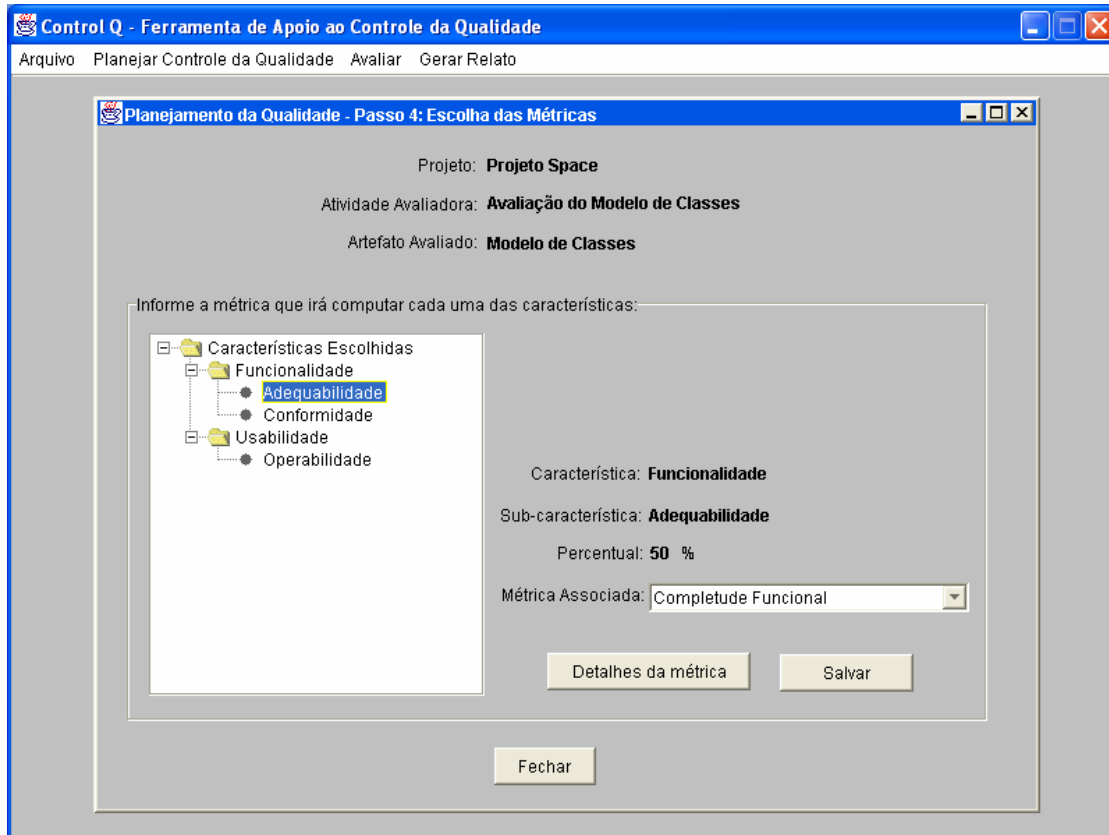


Figura 5. Definição da métrica que irá quantificar a característica de qualidade.

Vários trabalhos exploram o uso de sistemas de gerência de conhecimento para apoiar tarefas da engenharia de software, tais como [3, 4, 11]. Borges et al. [4] armazenam e compartilham experiência obtida na definição de processos de software. Para compartilhar este conhecimento, um repositório de experiências foi construído, contendo o processo padrão da organização, assim como os artefatos e o conhecimento informal obtidos ao longo dos projetos.

Markkula [3] descreve uma iniciativa realizada na ICL Finland para promover o compartilhamento e reuso do conhecimento obtido na engenharia de software. A fonte mais importante de conhecimento são as experiências de projeto, que devem ser identificadas e compartilhadas. Um framework foi desenvolvido para criação, captura, armazenamento e aplicação de conhecimento formal e informal nos projetos da organização.

Althoff et al. [11] defendem que o reuso contínuo da experiência em engenharia de software pode ser apoiado por uma memória organizacional capaz de gerenciar todos os tipos de experiências obtidas nas tarefas da engenharia de software. Os autores propõem uma arquitetura genérica e uma metodologia embutida para o reuso destas experiências.

A partir destes trabalhos, podemos observar vários pontos em comum. Todos os trabalhos citados, incluindo o nosso, se baseiam no conceito de Fábrica de Experiência [12]. A fábrica de experiência é uma unidade organizacional que apoia o reuso de experiência e conhecimento coletados no desenvolvimento de software. Porém, nenhum dos trabalhos citados anteriormente se

destina ao apoio ao planejamento da qualidade, como objetivo principal. Uma característica importante de nosso trabalho é que a abordagem de gerência de conhecimento proposta está ativamente integrada ao processo de trabalho. Ou seja, nossa preocupação é capturar informação diretamente do processo de desenvolvimento de software sem necessitar de esforço extra, apoiando desenvolvedores em suas tarefas através de uma memória organizacional ativa.

Observando aspectos estruturais de um sistema de gerência de conhecimento, também podemos encontrar trabalhos relacionados na literatura. Abecker et al. [5] definem uma abordagem de gerência de conhecimento com uma memória organizacional no centro do sistema de gerência de conhecimento. Posicionados ao redor desta memória, estariam os serviços da gerência de conhecimento provendo conhecimento ativamente a seus usuários. Nossa abordagem compartilha de muitas definições propostas por estes autores. Desta forma, nosso sistema de gerência de conhecimento também possui uma memória organizacional posicionada no centro e ao seu redor localizam-se serviços de captura, busca, disseminação, utilização e manutenção do conhecimento.

Ontologias foram apontadas como sendo cruciais em sistemas de gerência de conhecimento [13, 6, 14]. Benjamins et al. [13], por exemplo, apresentam uma abordagem para gerência de conhecimento baseada em ontologias e utilizam ontologias para a organização e estruturação do conhecimento. Ontologias também possuem um papel de extrema importância em nossa abordagem,



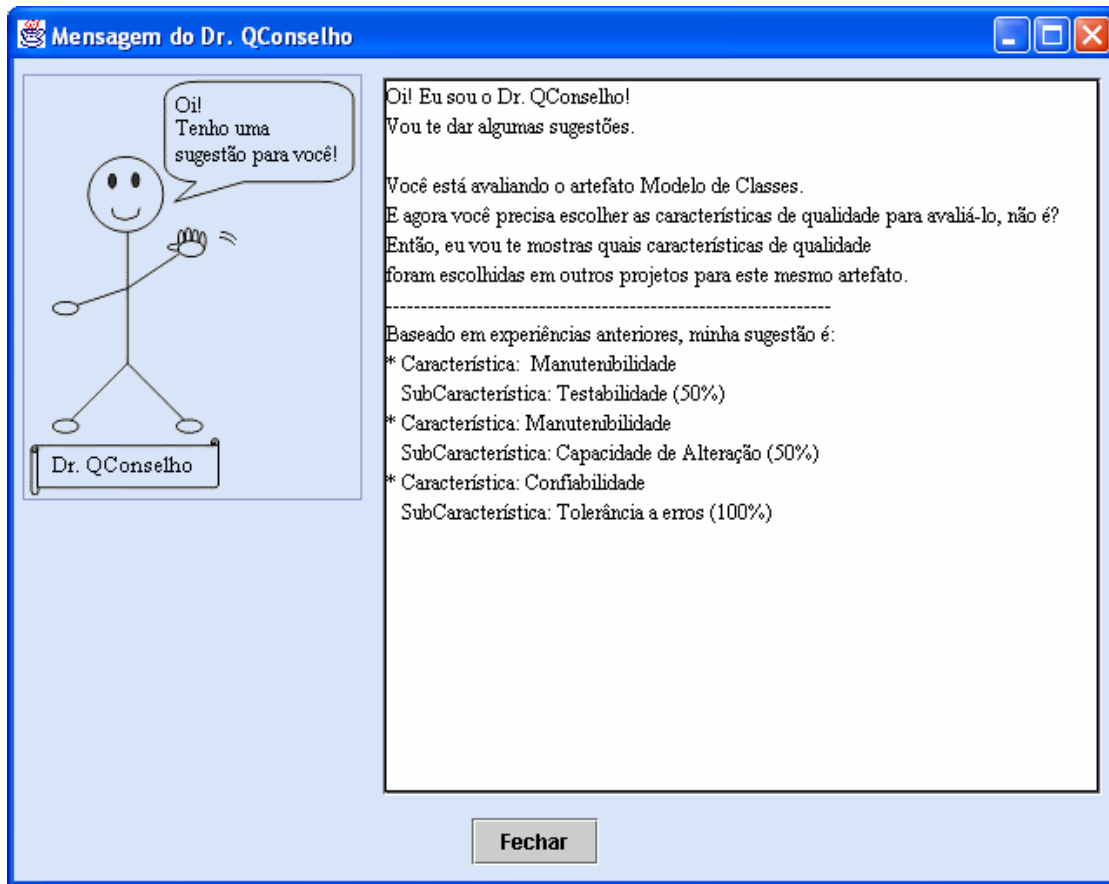


Figura 6. Disseminação do conhecimento de qualidade através de um agente de software.

já que são utilizadas para estruturar a memória organizacional, contendo o conhecimento de qualidade de software. Porém, em nossa abordagem, instâncias da ontologia também desempenham o papel de itens de conhecimento, já que a ontologia de qualidade de software pode ser instanciada e reutilizada.

Finalmente, várias pesquisas apontam os benefícios do uso de agentes de software na gerência de conhecimento. Rabarijoana et al. [14] sugerem o uso de agentes para a recuperação de conhecimento. Staab et al. [15] apresentam uma abordagem para a disseminação pró-ativa de conhecimento. Agentes se baseiam no conhecimento criado na execução de tarefas usuais e oferecem ao usuário conhecimento que pode ser relevante à tarefa que está desempenhando no momento. Em nossa abordagem, agentes também disseminam conhecimento, tendo por base a necessidade do usuário. Porém, nosso apoio de agentes está baseado em passos específicos de uma atividade, baseando-se em definições da ontologia que apóia esta atividade, para disseminar conhecimento de forma mais efetiva.

## 8. CONCLUSÕES

O planejamento e controle da qualidade não são tarefas simples e requerem um suporte baseado em experiência. Qualquer tipo de experiência pode ser reutilizado com intuito de evitar retrabalho e melhorar a qualidade, entre os quais conhecimento formal - ontologias e artefatos do desenvolvimento - e conhecimento

informal - melhores práticas em engenharia de software e lições aprendidas. Para tornar efetivas essas atividades tão importantes para a garantia da qualidade, é necessário oferecer apoio automatizado.

Neste trabalho, discutiu-se o compartilhamento de conhecimento e experiência como mecanismo de apoio ao planejamento e controle da qualidade de software e foi apresentada ControlQ, uma ferramenta de apoio ao planejamento e controle da qualidade. É importante ressaltar que ControlQ permite a criação de lições aprendidas e artefatos, que são armazenados como conhecimento na memória organizacional. Esta memória organizacional, funcionando como um repositório, propicia a disseminação das soluções bem sucedidas e das oportunidades de melhoria de uma organização, permitindo o aprendizado coletivo da mesma no contexto do controle da qualidade.

A abordagem de gerência de conhecimento apresentada neste trabalho pode ser reaproveitada para a gerência de outros tipos de conhecimento, não se limitando apenas ao conhecimento de qualidade de software. Para tal é importante que sejam desenvolvidas ontologias, a partir das quais novos itens de conhecimento sejam estruturados e atividades sejam apoiadas.

## 9. AGRADECIMENTOS

Os autores agradecem à CAPES pelo apoio financeiro a este trabalho.

## 10. REFERÊNCIAS

- [1] O'Leary, D.E. Enterprise Knowledge Management. IEEE Computer Magazine, March, 1998.
- [2] O'Leary, D.E., and Studer, R. Knowledge Management: An Interdisciplinary Approach. IEEE Intelligent Systems, January/February, Vol. 16, No. 1, 2001.
- [3] Markkula, M. Knowledge Management in Software Engineering Projects. In: Proc. of the 11th International Conference on Software Engineering and Knowledge Engineering, Kaiserslautern, Germany, 1999.
- [4] Borges, L.S., and Falbo, R.A. Managing Software Process Knowledge, Proc. of the CSITeA'2002, June 2002.
- [5] Abecker, A., Bernardi, A., and Hinkelman, K. Toward a Technology for Organizational Memories. IEEE Intelligent Systems, Vol. 13, No. 3, pp. 40-48, May/Jun, 1998.
- [6] Staab, S., Studer, R., Schurr, H. P., and Sure, Y. Knowledge Processes and Ontologies. IEEE Intelligent Systems, January/February, Vol. 16, No. 1, 2001.
- [7] Broomé, M., and Runeson, P. Technical Requirements for the Implementation of an Experience Base. . In: Proc. of the 11th International Conference on Software Engineering and Knowledge Engineering, Kaiserslautern, Germany, 1999.
- [8] Mian, P. G., Natali, A. C. C., and Falbo, R. A. Ambientes de Desenvolvimento de Software e o Projeto ADS. Revista Engenharia Ciência Tecnologia, Vitória, ES, v. 04, n. 04, p. 3-10, 2001.
- [9] Duarte, K.C., and Falbo, R. A. Uma Ontologia de Qualidade de Software. Anais do VII Workshop de Qualidade de Software, WQS'2000, João Pessoa, Brasil, Outubro 2000.
- [10] Guizzardi, G., Falbo, R.A., and Pereira Filho, J.G. Using Objects and Patterns to Implement Domain Ontologies. In: Proc. of the 15th Brazilian Symposium on Software Engineering, Rio de Janeiro, Brazil, 2001.
- [11] Althoff, K., Birk, A., Hartkopf, S., Muller, W., Nick, M., Surmann, D., and Tautz, C. Managing Software Engineering Experience for Comprehensive Reuse. In: Proc. of the 11th International Conference on Software Engineering and Knowledge Engineering, Kaiserslautern, Germany, 1999.
- [12] Basili, V., Caldiera, G., and Rombach, H. The Experience Factory, Vol. 1 of Encyclopedia of Software Engineering, Chapter X, John Wiley & Sons. 1994.
- [13] Benjamins, V. R., Fensel, D., and Pérez, A. G. Knowledge Management through Ontologies, Proc. of the 2<sup>nd</sup> International Conference on Practical Aspects of Knowledge Management (PAKM98), Switzerland, 1998.
- [14] Rabarijoana, A., Dieng, R., and Corby, O. Building a XML-based Corporate Memory, Workshop on Knowledge Management and Organizational Memories In the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI99), Sweden, 1999.
- [15] Staab, S., and Schurr, H. P. Smart Task Support through Proactive Access to Organizational Memory. Knowledge-based Systems, 13(5): 251-260. Elsevier, 2000.