

Um Caso de Integração de Gerenciamento Ágil de Projetos à Metodologia CommonKADS

Eduardo S. Estima de Castro¹, Felipe I. Victoreti², Sandro R. Fiorini¹, Mara Abel¹, R. Tom Price¹

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

²ENDEEPER - Centro de Empreendimentos em Informática
Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Av. Bento Gonçalves, 9500 – 91.501-970 – Porto Alegre – RS – Brazil

{esecastro, srfiorini, marabel, tomprice}@inf.ufrgs.br,
felipe.victoreti@endeeper.com

Abstract. *This article presents the results of the integration between some agile project management techniques with knowledge systems project management process as defined in the CommonKADS methodology. This integration allows the use of CommonKADS in small sized companies, without productivity losses and it allows the use of processes in small teams for the development of large knowledge systems. The results were extracted from Petroledge project, whose objective is the development of a knowledge system to petroleum exploration.*

Resumo. *Este artigo apresenta os resultados da integração de algumas técnicas de desenvolvimento ágil de projetos à gerência de projetos de sistemas de conhecimento definida na metodologia CommonKADS. Essa integração permite aplicar CommonKADS em empresas de pequena porte, sem redução de produtividade e permite o uso de processos por pequenas equipes no desenvolvimento de sistemas de conhecimento de grande porte. Os resultados foram extraídos do projeto Petroledge, cujo objetivo é o desenvolvimento de um sistema de conhecimento para a área de exploração de petróleo.*

1. Introdução

Empresas de pequeno porte da área de software, freqüentemente, têm dificuldade em desenvolver uma cultura de gerência de projetos devido a restrições de orçamento e recursos humanos. Além disso, a aplicação de todas as práticas definidas em metodologias tradicionais de gerência de projetos pode reduzir, significativamente, a produtividade da equipe envolvida. Por exemplo, o PMBOK [PMI 2004] organiza um conjunto de melhores práticas para gerência de projetos em 44 processos compostos por um grande número de etapas. Desse modo, uma equipe pequena, 5 a 7 pessoas, para utilizá-lo, normalmente precisa selecionar um sub-conjunto de processos que seja compatível com a baixa disponibilidade recursos humanos. Entretanto, corre o risco de

selecionar um subconjunto de processos inadequados para suas atividades e também comprometer sua produtividade.

Projetos de sistemas de conhecimento necessitam, durante o projeto, gerenciar constantemente a evolução do escopo, pois sistemas de conhecimento se caracterizam por uma contínua evolução da base de conhecimento, refletindo o estado da área ao qual se aplica, com conseqüente evolução dos requisitos do sistema. A incorporação de novos conhecimentos exige, na maioria das vezes, a modificação de modelos e componentes já desenvolvidos no sistema. Essas constantes modificações dificultam o uso de metodologias em cascata ou com longos ciclos, mas favorecem o uso de metodologias ágeis. Além disso, esse tipo de projeto, para ter resultados satisfatórios, precisa não somente produzir um sistema que suporte a gestão de conhecimento e a tomada de decisão, mas um conjunto de componentes reutilizáveis e integráveis aos demais sistemas de informação corporativos das organizações.

Nas próximas sessões, serão apresentados os passos realizados para atingir a integração de práticas ágeis à metodologia CommonKADS e os resultados obtidos. Na seção 2, serão explicitados os objetivos gerais do projeto Petroledge. Na seção 3, será detalhada a integração entre o gerenciamento ágil de projeto e o ciclo tradicional proposto pela metodologia CommonKADS [Schreiber et al. 2000].

2. Objetivos do Projeto Petroledge

O projeto Petroledge tem como objetivo central transformar um sistema de conhecimento, desenvolvido inicialmente no ambiente acadêmico, em um produto comercial estruturado para suportar sua constante evolução. O sistema Petroledge, produto do projeto, auxilia na tarefa de descrição e gerencia análises detalhadas de descrições petrográficas de rochas-reservatório de petróleo. Sobre as informações capturadas o sistema aplica métodos de raciocínio para extrair interpretações geológicas úteis para a avaliação da qualidade de reservatórios de petróleo. A base de conhecimento do sistema contém a descrição de mais de 120 conceitos (definem as classes de rochas, minerais e seus qualificadores) e seus relacionamentos, sobre os quais são aplicados treze diferentes métodos de classificação de rocha, além de outros três métodos de interpretação geológica com modelos de inferência próprios [Abel et al. 2004]. A formalização da ontologia de descrição petrográfica, garante a padronização da nomenclatura utilizada e o formato das descrições, reduzindo os efeitos da subjetividade do petrógrafo na descrição e permitindo a extração de correlações geológicas de forma automática por outros sistemas [De Ros et al. 2007]. A arquitetura e modelo de funcionamento do sistema foram concebidos para o ambiente corporativo, onde grandes volumes de dados são capturados de forma geograficamente distribuída, requerendo alto nível de sigilo, segurança de dados e confiabilidade nos sistemas.

Pesquisas na área da Computação relacionadas a sistemas complexos, tal como o Petroledge, envolvem a participação de um número significativo de desenvolvedores. Entretanto, o desenvolvimento no ambiente acadêmico utiliza uma abordagem baseada em protótipos e experimentações, o que dificulta o controle gerencial e, também, inibe a obtenção de resultados aplicáveis diretamente no ambiente empresarial, pois os resultados normalmente são aplicações experimentais e instáveis para o uso corporativo.

Além disso, a heterogeneidade dos desenvolvedores e a ausência de padrões de desenvolvimento colaboram para o desenvolvimento de protótipos funcionais voltados para a validação das idéias de cada pesquisa, mas sem consideração ao reuso dos componentes de software gerados para o tratamento desse tipo de conhecimento.

No caso do projeto acadêmico do Petroledge, que durou sete anos, um número significativo de desenvolvedores participou do desenvolvimento e, mesmo com o estabelecimento de uma arquitetura base na fase inicial do projeto [Abel et al. 2004], o sistema obtido apresentava alto custo para manutenção. As principais causas do alto custo eram as seguintes: a ausência de uma arquitetura central; ausência de padrões formais de desenvolvimento; alto índice de acoplamento e, por conseguinte, baixo índice de reuso de componentes. Em adição, devido à dinamicidade das pesquisas e à própria característica de constante de evolução desse tipo de sistema, existia uma grande dificuldade para gerir as evoluções no modelo de conhecimento do sistema e a implementação dessas mudanças no sistema.

Além da complexidade, naturalmente envolvida no desenvolvimento de sistemas de conhecimento, o produto comercial necessitava atender aos seguintes requisitos: funcionar em um ambiente multiplataforma (Windows, Linux e MacOS); suportar o armazenamento das informações em diferentes gerenciadores de banco de dados e, também, em bases de dados heterogêneas; permitir a customização de seus componentes de acordo com a necessidade do cliente; apresentar desde versões voltadas para usuários individuais até versões voltadas para o ambiente corporativo de grandes petrolíferas (Figura 1) e garantir a segurança dos dados, pois informações petrográficas são extremamente valiosas.

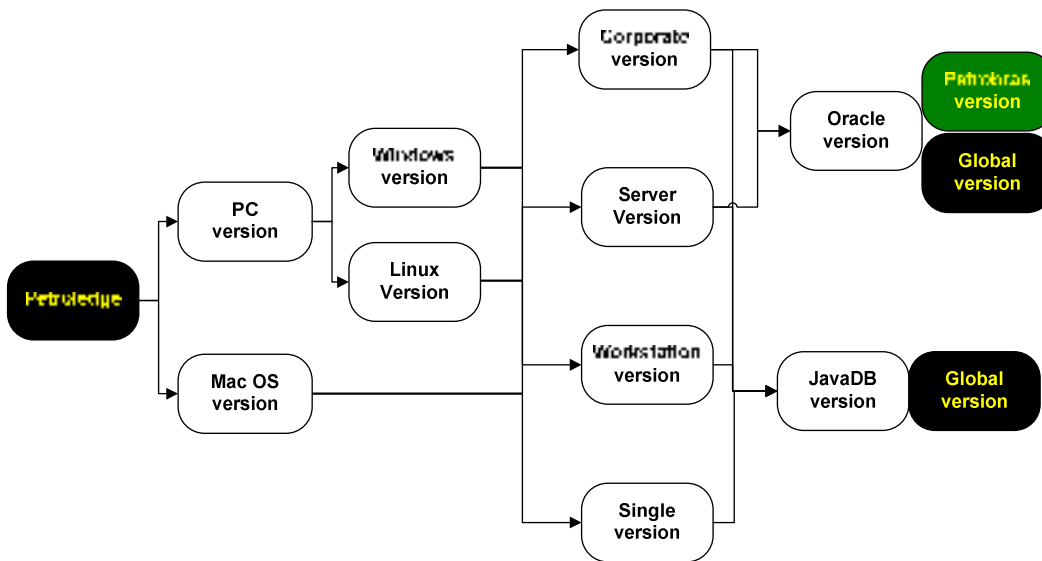


Figura 1. Petroledge – Versões comerciais

Esses novos requisitos para a versão comercial indicavam a necessidade de uma nova arquitetura que fosse aplicável em um sistema de conhecimento de grande porte. O padrão arquitetural adotado, baseado em [Fiorini 2006], se adapta a esse perfil pois incorpora conceitos de sistemas de conhecimento, como preservação de estruturas e

divisão entre modelos de conhecimento declarativo e procedimental, em uma arquitetura de componentes voltada a sistemas de informação de grande porte. Nessa arquitetura (Figura 2), os componentes de conhecimento (raciocínio e modelo de conhecimento) são separados dos componentes de sistema (como segurança e controle) seguindo o padrão de arquitetura de software *Model-View-Controller* (MVC) e mapeados para base de dados por um componente de abstração. O padrão MVC estrutura o software em três camadas [Buschmann et al. 1996]:

- *Model* – camada de dados do domínio da aplicação;
- *View* – camada de apresentação, frequentemente corresponde a interfaces gráficas;
- *Controller* – camada de processamento lógico. Trata interações com a *view* e gerencia alterações nos dados modelados na camada *model*.

Essa nova arquitetura, planejada para permitir um alto índice de re-uso de componentes e suportar expansões gradativas do sistema, levou a uma diminuição significativa no tempo para manutenção e expansão do sistema Petroledge.

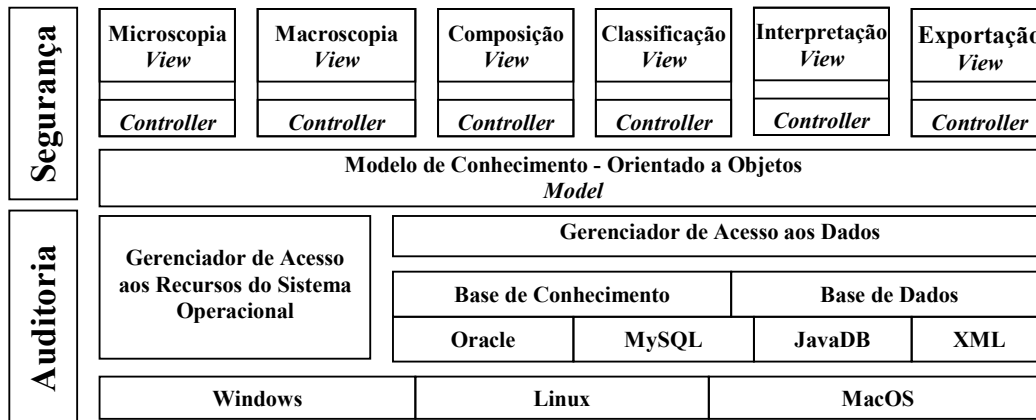


Figura 2. Organização Geral do Petroledge – Arquitetura e Componentes

Após a definição de todos os requisitos fundamentais para a obtenção de um produto comercial oriundo de um sistema desenvolvido no ambiente acadêmico, havia a necessidade de desenvolver uma cultura de gerenciamento de projetos compatível com as características de uma empresa de pequeno porte, mas que garantisse o atendimento aos requisitos e ao desenvolvimento do projeto com o alto nível de qualidade exigido pelas companhias de petróleo.

3. Gerenciamento Ágil de Projetos de Sistemas de Conhecimento

A metodologia CommonKADS estabelece um conjunto de processos para a realização de projetos de sistemas de conhecimento e propõem o ciclo de vida de gerência de projeto (Figura 3) baseado no modelo espiral [Schreiber et al. 2000]. Cada ciclo visa desenvolver progressivamente o conjunto de modelos propostos em CommonKADS para a elaboração de um sistema de conhecimento. Esses modelos são organizados em três níveis [Schreiber et al. 2000]:

- Nível contextual: modelos desse nível definem o contexto organizacional e buscam mapear as demandas da Gestão de Conhecimento da organização onde o sistema irá se inserir. O objetivo é identificar as tarefas intensivas em conhecimento estratégicas para a organização e os agentes (pessoas e sistemas) afetados por elas.
- Nível conceitual: tem como foco a captura dos objetos do conhecimento e os métodos de raciocínio utilizados para resolver problemas gerando o modelo de representação. Esse modelo é o componente principal dos sistemas e também o que mais sofre atualizações como uma evolução natural da compreensão do domínio. As tarefas intensivas em conhecimento mapeadas no nível conceitual são analisadas detalhadamente para identificar quais são os insumos de conhecimentos necessários, o raciocínio envolvido em cada uma e como a comunicação ocorrerá entre os diferentes agentes.
- Nível de artefatos: modelos desse nível detalham a estrutura do software a ser desenvolvido. Especificam a arquitetura do software, projeto de algoritmos, estruturas de dados, plataforma de hardware/software e linguagens de implementação.

Os níveis não existem apenas para especificar em que nível de abstração as especificações são feitas, mas definem uma real identidade entre os objetos abstratos identificados na organização e representados no modelo conceitual com os componentes de software especificados no nível do projeto. Este princípio é descrito em CommonKADS como *Manutenção de Estrutura*, e tem por finalidade permitir agilidade na atualização do conhecimento e raciocínio do sistema sem causar efeitos colaterais indesejáveis em seu funcionamento.

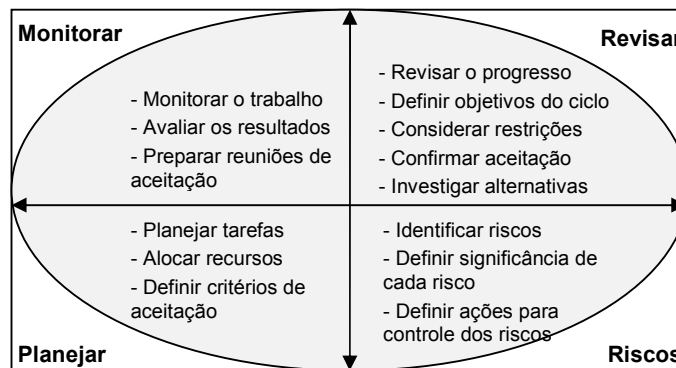


Figura 3. Ciclo de Vida do CommonKADS para Gerenciamento de Projetos de Sistemas de Conhecimento [Schreiber et al. 2000]

Devido a quantidade de modelos propostos e o nível de detalhamento exigido, cada ciclo do projeto demanda a gerência de uma grande quantidade de documentação e a execução de uma série de atividades que reduzem o tempo de resposta às constantes mudanças nesse tipo de projeto. Desse modo, para que uma empresa de pequeno porte consiga utilizar essa metodologia, foram aplicadas técnicas ágeis no ciclo espiral proposto em CommonKADS.

Inicialmente, para a definição da adaptação de cada etapa do ciclo de acordo com as necessidades do projeto comercial, foi realizada uma análise para determinar o estado geral do sistema no final do projeto acadêmico. A análise post-mortem do projeto acadêmico permitiu a identificação de uma série de problemas e conseqüências dos mesmos: aplicação individual pelos desenvolvedores, de suas idéias para estruturação e desenvolvimento de módulos levou à múltiplos modelos de módulos; falta de comunicação entre os desenvolvedores ocasionou duplicação de componentes que poderiam ser compartilhados e a falta de estilo de codificação tornava difícil a compreensão do código. Dessa forma, o projeto comercial, para evitar os problemas citados acima, precisou construir um modelo para todos os módulos que viessem a compor o sistema. A equipe de seis pessoas, que já tinha participado do projeto acadêmico, foi reunida para reconstruir um módulo chamado de *golden prototype*.

O desenvolvimento desse protótipo teve dois objetivos principais: validar o ciclo proposto a partir da integração entre metodologias ágeis e do ciclo proposto por CommonKADS e definir o modelo de organização, construção e codificação de todos os outros módulos. O ciclo utilizou um conjunto de idéias baseadas na integração entre *Extreme Programming* e *Scrum* chamada de *xP@Scrum* [Kane 2002]:

- Reuniões diárias de 20 minutos com todos os membros da equipe foram realizadas para definir pequenas partes do padrão de desenvolvimento à medida do necessário;
- Desenvolvimento em duplas dos requisitos críticos. Cada dupla era formada por um membro com maior experiência e um com menor experiência para garantir a evolução de conhecimento da equipe e reduzir erros de desenvolvimento ou arquiteturais;
- Uso intensivo de *refactoring*. Essa atividade foi importante para suportar novos requisitos, evitar duplicação de código e, principalmente, melhorar o código já disponível do projeto acadêmico com foco na redução de complexidade;
- Revisão diária de código voltada para verificação da aplicação dos padrões de codificação estabelecidos pela equipe;
- Testes desenvolvidos em paralelo para validar o protótipo ao longo do desenvolvimento. Os testes foram tanto unitários, quanto funcionais. Os testes funcionais foram realizados pelos especialistas em petrografia, usuários já experientes do sistema acadêmico;
- Disponibilidade total dos gerentes: os gerentes do projeto buscavam estar sempre disponíveis para resolução de dúvidas ou conflitos. Isso evitou que atrasos ocorressem devido a espera por resolução de dúvidas ou conflitos.

Essas práticas permitiram que o desenvolvimento fosse incremental, facilitaram o estabelecimento de um ambiente de aprendizagem constante para todos os membros da equipe e contribuíram para que a equipe aceitasse e aprendesse gradativamente os padrões estabelecidos em conjunto.

O ciclo ágil aplicado, diferentemente do ciclo de quatro estágios definido em CommonKADS, foi dividido em três estágios: revisão, planejamento e monitoração. O estágio relacionado a riscos foi incorporado continuamente aos demais estágios.

O estágio de revisão proposto por CommonKADS estabelece cinco atividades principais: revisar o progresso, definir os objetivos do ciclo, considerar restrições, investigar alternativas e confirmar a aceitação dessa revisão pelos envolvidos nesse estágio. A execução convencional pela primeira vez dos ciclos deve resultar em um plano completo para o projeto; entretanto, no ciclo ágil, optou-se por construir um plano minimalista. Esse plano apenas identificou os módulos a serem desenvolvidos, definiu sucintamente os casos de uso de cada módulo e o controle de qualidade foi estabelecido. Esse plano tratava três principais restrições: a disponibilidade de mão-de-obra, o orçamento e o prazo a ser cumprido. A aceitação do plano proposto ocorreu através de reuniões dinâmicas envolvendo os patrocinadores e os desenvolvedores, fator que colaborou significativamente para a integração da equipe e permitiu que cada membro identificasse claramente sua importância para o sucesso dos ciclos.

No estágio de planejamento, CommonKADS estabelece três atividades principais: definir as tarefas do ciclo, alocar recursos e definir os critérios para aceitação. Segundo essa metodologia, deve-se elaborar um plano de projeto tradicional (estrutura analítica de projeto, cronograma, plano de comunicação, planejamento de qualidade, entre outros); entretanto, a elaboração dessa documentação detalhada é muito custosa para uma empresa de pequeno porte. Desse modo, para alinhar esse estágio a práticas ágeis, optou-se apenas por registrar as tarefas, sem detalhamento, a serem desenvolvidas, definir o prazo limite para a realização de cada uma e identificar os responsáveis pelo desenvolvimento de cada uma. A definição dos responsáveis não era feita por imposição hierárquica, mas através de uma reunião em que cada participante identificava a sua preferência, isso colabora para um bom rendimento, pois o trabalho era desenvolvido de acordo com preferências pessoais [Terra 2001]. Não houve detalhamento de tarefas devido a existência de uma versão prévia do sistema, que permitia identificar o conjunto de sub-tarefas envolvidas e os respectivos requisitos.

No estágio de monitoração, CommonKADS estabelece três atividades principais: monitorar o desenvolvimento do trabalho, preparar reuniões com os patrocinadores para a aceitação do trabalho realizado e avaliar os resultados do ciclo. Apesar de a metodologia definir essas tarefas em um estágio específico, no modo ágil adotado, o monitoramento também passou a ser feito continuamente, isso permitiu identificar rapidamente melhorias possíveis nos processos envolvidos e acrescentou uma alta reatividade às mudanças. No projeto Petroledge, a monitoração dos ciclos de desenvolvimento foi realizada através de reuniões semanais/diárias e da análise de relatórios gerados pela integração entre um software de controle de versão, um sistema para envio automático de e-mails conforme evolução no desenvolvimento e um sistema para registro de problemas de qualidade. Isso facilitava a avaliação de rendimento, a verificação da evolução do projeto e a preparação de apresentação de resultados aos patrocinadores.

O estágio de avaliação/mitigação de riscos foi incorporado aos demais para aumentar a capacidade de resposta da equipe à constante evolução de escopo. As

principais atividades definidas em CommonKADS (identificação de riscos, definição de significância de cada um e proposição de medidas de controle) foram realizadas continuamente nos três outros estágios. Sempre que um risco era identificado, ele era imediatamente analisado pela equipe e medidas para reduzi-lo ou eliminá-lo eram tomadas. Isso evitava que o risco fosse identificado somente após os efeitos do mesmo, no estágio específico para isso. Por exemplo, o risco de mudança de requisito devido à evolução de conhecimento é constantemente alto nesse tipo de projeto, desse modo, ele era continuamente monitorado e, qualquer dúvida que surgisse relacionada à possibilidade de mudança de requisitos era rapidamente solucionada, evitando assim re-trabalho.

4. Conclusão

A abordagem de integração entre metodologias ágeis de gerenciamento de projetos e a metodologia CommonKADS, apresentou resultados bastante satisfatórios. O projeto Petroledge atingiu os prazos planejados, os requisitos foram atendidos, um padrão de desenvolvimento e de gerência de projetos foi estabelecido e o software produzido teve uma taxa de 2% de erros em relação a 3000 testes aplicados por uma empresa de qualidade de software após o desenvolvimento de todos os módulos. Atualmente, ele está em uso pela Petrobras, uma das 10 maiores empresas de petróleo do mundo.

A integração realizada permitiu extrair as vantagens das propostas de CommonKADS e das propostas ágeis. Em uma empresa de pequeno porte, a aplicação de todos as propostas definidas em CommonKADS pode comprometer a produtividade; entretanto, a ausência de uma metodologia de base para o desenvolvimento desse tipo de sistema pode comprometer o resultado do projeto. Desse modo, as práticas ágeis integradas à CommonKADS foram essenciais para permitir o uso dessa metodologia.

A próxima etapa dessa proposta de integração será formalizar os processos utilizados e estudar como definir estimativas de custos viáveis comercialmente para o desenvolvimento de sistemas de conhecimento, pois a dinamicidade do escopo pode comprometer planejamentos iniciais de custo e resultar em projetos que excedem o custo estimado.

Agradecimentos

Esse trabalho recebe suporte do Edital MCT/CNPq 02/2006 - Universal No. 475597/2006-0. Agradecemos a todos os membros da ENDEEPER pelo apoio em validar a proposta inovadora de gerência de projeto descrita nesse artigo.

Referências

- Abel, M., Silva, L., De Ros, L., Mastella, L., Campbell, J. and Novello, T. (2004). PetroGrapher: managing petrographic data and Knowledge using an intelligent database application. Em *Expert Systems with Applications* 26 (1), p.9-18.
- Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M. (1996). Pattern-Oriented Software Architecture Volume 1: A System of Patterns. Wiley.

- De Ros, L., Goldberg, K., Abel, M., Victoreti, F., Mastella, L. and Castro, E. (2007). Advanced Acquisition and Management of Petrographic Information from Reservoir Rocks Using the PETROLEDGE® System. Em *AAPG Annual Meeting*.
- Fiorini, S. R. (2006). Uma proposta de arquitetura de componentes para sistemas de conhecimento para avaliação de reservatórios de petróleo. Trabalho de Conclusão, Universidade UFRGS.
- Kane, M., Schwaber, K. (2002). Scrum with XP, <http://www.controlchaos.com/about/xp.php>, Março 2007.
- Project Management Institute – PMI (2004). PMBOK Guide: Um guia do conjunto de conhecimentos do gerenciamento de projetos. Pennsylvania, Project Management Institute, 3. ed., p. 11.
- Schreiber, G., Akkermans, H., Anjewierdem, A., Hoog, R., Shadbolt, N., Velde, W. e Wielinga, B. (2000). Knowledge Engineering and Management : the CommonKADS methodology. Cambridge, MIT Press.
- Terra, J. (2001). Gestão do Conhecimento: O Grande Desafio Empresarial. São Paulo, Negócio Editora, p. 136.