

PROCESSOS ÁGEIS E GESTÃO EM PROJETOS DE SOFTWARE

José Augusto Fabri²²

Resumo: a área de desenvolvimento de software é carente de uma gestão de projetos concisa e consistente. Diversas pesquisas apontam que a referida as empresas produtoras de software não praticam conceitos chave ligados a processo e gestão de projetos de software. Dentro deste contexto este trabalho tem como objetivo apresentar alguns de relatos sobre gestão em projetos e processo ágeis que possam ser incorporados pelas empresas de software.

Palavras Chaves: Processo, Gestão de Projetos, Engenharia de Software.

Abstract: the software development area needs of a concise and consistent projects management. Several studies indicate that such companies producing software don't practice key concepts related to software projects process and management. Within this context, this paper presents some reports about project management and agile process that can be incorporated by software companies.

Keywords: Process, Management Project, Software Engineering.

Introdução

Atualmente poucas empresas brasileiras produzem software com padrões de qualidade reconhecidos por algum modelo. Esta afirmação pode ser constatada ao ser analisada a quantidade de empresas certificadas no Modelo Referencial baseado em Maturidade, Capacidade e Integração (CMMI), CMMI-DEV e CMMI-SVC, o Brasil possui 221 empresas certificadas. Para efeitos comparativos, a China possui 3.316, os Estados Unidos da Américas 2186 e a Índia 959 (Fonte Instituto de Engenharia de Software).

A atividade de gestão em projetos é um diferencial nas empresas produtoras de software que possuem um processo de produção institucionalizado. É uma pena que o número de empresas que encaram tal atividade como relevante é pequeno. Esta afirmação pode ser constatada nos números publicados pelo Standish Group, percebe que o título do relatório é bem sugestivo (vide Tabela 1).

A pesquisa classifica os projetos finalizados em três categorias.

Sucesso: Projeto que terminaram no prazo estipulado, dentro do orçamento e com escopo completo.

Mudaram: Projetos que atrasaram, estouraram o orçamento e/ou tiveram o escopo reduzido.

Falharam: Projetos que foram cancelados ou nunca foram usados.

²² Tecnólogo em Processamento de Dados. Instituto Municipal de Ensino Superior de Assis. Fundação Municipal de Ensino Superior de Assis. Atualmente: Professor da Área de Engenharia de Software. Universidade Tecnológica Federal do Paraná Campus Cornélio Procopio. Líder Técnico do Grupo de Pesquisa em Tecnologia da Informação - GTI - UTFPR. E-mail: fabri@utfpr.edu.br

	2004	2006	2008	2010	2012
Successful	29%	35%	32%	37%	39%
Failed	18%	19%	24%	21%	18%
Challenged	53%	46%	44%	42%	43%

Tabela 1: CHAOS Report 2012

Percebam que 39% dos projetos, em 2012, tiveram uma gestão efetiva. É importante salientar que o *report* publicado pela *Standish* generaliza este número em um contexto global.

Principais fatores que ajudam no sucesso dos projetos:

- Envolvimento dos *stakeholders*;
- Apoio irrestrito;
- Requisitos claros concisos e consistentes;
- Planejamento apropriado;
- *Milestones* pequenos;
- Realismo quanto às expectativas;
- Visão e objetivos claros;
- Trabalho duro e equipe focada;
- Equipes competentes;
- Propriedade.

Principais fatores que ajudam no fracasso dos projetos:

- Falta de planejamento;
- Requisitos Incompletos;
- Falta de envolvimento do usuário;
- Falta de recursos;
- Expectativas não realistas;
- Falta de apoio executivo;
- Mudanças de requisitos;
- Não precisa mais daquilo;
- Falta de gestão da TI;
- Analfabetismo tecnológico.

Dado o contexto apresentado, os números apresentados nesta introdução, aliado a alta carga tributária e a deficiência na formação da mão obra na área de tecnologia da informação, mais precisamente na área de engenharia de software, caracteriza-se como fonte inibidora no processo de expansão externa do Brasil neste setor. Universidade, empresa e governo devem desenvolver mecanismos que alterem este cenário.

Este trabalho tem como objetivo apresentar alguns de relatos sobre gestão em projetos e processo ágeis que possam ser incorporados pelas empresas de software com o objetivo de minimizar problemas relatados nesta seção. Este relatos mostram como o GTI possibilita a institucionalização da gestão em projetos de software nas empresas da região metropolitana de Londrina.

Para atingir o objetivo traçado o artigo foi dividido em seções. A seção 2 apresenta uma visão sobre a teoria de gestão em projetos de software. Uma das

formas de gestão em projetos é apresentada na seção 3 - a gestão de expectativas. E as práticas ágeis alinhadas ao *Scrum* são apresentadas nas seções 4 e 5. Por fim a seção 6 apresenta as conclusões do trabalho.

Gestão DE projetos de software ou Gestão EM projetos de software?

Um dos pontos predominantes na gestão EM um projeto é a organização e institucionalização de um processo de produção. É justamente neste item que se localiza o “calcanhar de Aquiles” das empresas de software. Nós, analistas de sistemas e engenheiros de software, temos a capacidade à técnica para organizar o processo nos mais variados domínios do conhecimento, porém somos medíocres quando se trata da organização do processo de produção de nossa própria empresa. Com um pensamento ordenado e estruturado as suas chances de sucesso em um projeto de software são muito maiores. Seu trabalho é menor. Sua capacidade produtiva aumenta.

A gestão EM projetos de software possui uma amplitude conceitual que foge aos olhos da maioria dos gerentes. A gestão deve transpor a “simples” capacidade cumprir os custos e prazos pré-estabelecidos na atividade planejamento, ou seja, não devemos nos ater somente na gestão DE projetos, é necessário implementar a gestão EM projetos.

A capacidade gerencial EM um projeto de software deve levar em considerações outros fatores:

1 - Gestão da qualidade: Alicerçada sobre dois pilares: qualidade do produto e qualidade do processo. No primeiro é necessário verificar se as necessidades pré-estabelecidas pelos clientes estão implementadas no produto. No segundo, é importante analisar se o processo está organizado, estruturado para produzir subprodutos de qualidade. Lembre-se que a institucionalização de um processo de qualidade leva, na maioria das vezes, a qualidade do produto.

2 - Gestão de configuração: Conjunto de ações (ou tarefas) que possibilitam identificar os produtos, em nosso caso todos os artefatos gerados pelo processo de software, estabelecendo um relacionamento entre eles. A gestão de configuração também é responsável por subsidiar a atividade de gerenciamento EM projetos, pois esta necessita controlar as diferentes versões geradas para um mesmo produto. Em suma, a referida gestão tem como objetivo responder as seguintes perguntas: Qual artefato do processo foi mudado? Quando ocorreu a mudança? Quem foi o responsável pela mudança? Qual o impacto que a mudança irá gerar?

a - Gestão de expectativa: Caracterizada pelo estabelecimento de uma relação harmoniosa entre clientes internos e externos. Definir ciclos curtos para entregar os subprodutos gerados durante o projeto de software, respeitar os prazos, surpreender os clientes, gerar novas expectativas, realimentando assim, o ciclo deste tipo de gestão, são pontos importantes e merecem destaque.

b - Gestão de risco: Minimizar as condições que comprometam ou impendam a realização de alguma parte projeto, é uma das prerrogativas embutidas neste tipo de gestão.

c - Gestão de escopo: Escopo é a linha que une o ponto em que você está ao ponto que você deseja chegar. Delimitar, de maneira recorrente, as necessidades, os produtos e as pessoas que pertencem a esta linha contextualizam este tipo de gestão.

d - Gestão de aquisição: Caracterizada pelo estabelecimento de um conjunto de atividades e critérios que possibilita a obtenção de bens e serviços fora dos limites organizacionais.

A grande dificuldade das empresas produtoras de software é alinhar, de maneira eficaz, as formas gestão descritas. Elas praticam (quando praticam...) a gestão DE projetos, ou seja, atentam somente para os aspectos temporais e custodiais. O EM é o ponto comum a todas as formas de gestão delineadas neste post.

É importante ressaltar que existem diversos modelos que possibilitam a troca do DE pelo EM, MPS-BR, CMMI e o PMBOK são exemplos claros.

Que os gerentes de projetos DÊEM o melhor de si para transpor as preposições maximizadas neste texto.

Após apresentar uma visão sobre a gestão EM projetos a próxima seção irá aprofundar uma das forma DE gestão - a gestão de expectativa.

A gestão de expectativa em um projeto de software

Em quanto tempo estaremos utilizando a primeira versão do software? Quando podemos treinar nosso pessoal, para que as primeiras informações sejam automatizadas?

As questões acima ocorrem com frequência em vários projetos de software. Frequentemente, respondo além destas, várias outras que têm como foco principal o tempo de entrega do produto.

É importante salientar que durante a gestão em qualquer tipo projeto, devemos desenvolver uma boa gestão de expectativa junto ao nosso cliente. Neste texto apresento algumas dicas que podem ser aplicadas, sistematicamente, na gestão de qualquer projeto. Nós do Grupo de Tecnologia da Informação da Universidade Tecnológica Federal do Paraná Campus Cornélio Procópio - PR - estamos aplicando tal sistematização em vários projetos.

a - Estabeleça ciclos curtos para entrega de artefatos.

Fatie o seu projeto em atividades e tarefas. Cada tarefa deve gerar artefatos (ou subprodutos) que agregam valor para o cliente. Entregue esses artefatos em ciclos curtos de produção. A expectativa do cliente sempre será controlada.

b - Ao estabelecer um prazo... CUMPRA.

Quando dizemos, no próximo dia 10 entregaremos os subprodutos A, B e C, expectativas são criadas junto ao cliente. O descumprimento de um compromisso quebram-nas, fator este que denigre sua credibilidade como gerente de projetos. Caso ocorra algum imprevisto e você não consiga honrar

o compromisso, convoque o cliente para uma reunião, exponha o problema e equalize uma solução.

c – Surpreenda o cliente.

Inove ao apresentar um subproduto. Destaque a importância do que está sendo apresentado para o projeto. Justifique o motivo pelo qual aquilo foi desenvolvido daquela forma. Cite os pontos fortes e fracos que você encontrou no desenvolvimento. Diga ao cliente que o desenvolvimento daquele subproduto contribuiu para o crescimento de sua equipe. Seu cliente é seu parceiro.

d – Descubra as “sub-expectativas” do cliente.

Lembre-se que um gerente nunca conseguirá mapear todas as expectativas do cliente no início do projeto. Geralmente, mapeamos apenas as expectativas globais, o produto terá o seguinte escopo, deve funcionar de tal forma. Ao apresentar os subprodutos gerados, expectativas locais ou “sub-expectativas” com certeza vão surgir. Essas “sub-expectativas” são de extrema importância na satisfação do cliente perante o projeto. Verifique a possibilidade de incorporá-las no escopo do produto, mesmo que seja em uma segunda ou terceira fase de execução.

e – Gere novas expectativas junto ao cliente.

Às vezes a visão do cliente é limitada e o mesmo não enxerga as potencialidades do produto junto ao mercado. Apresente as potencialidades do produto junto ao cliente.

Para finalizar saliento que os recursos necessários para o desenvolvimento de alguns projetos podem ser duradouros. Novos produtos sempre vão gerar novas ideias e novas ideias sempre vão gerar novos produtos. A gerência de expectativa do cliente é de fundamental importância neste contexto. O algoritmo para a gestão de expectativa é relativamente simples: 1 – Atender as expectativas. 2 – Gerar novas expectativas. 3 – Volte ao passo 1.

Apresentada uma das formas DE gestão EM projetos, a próxima seção irá propor a inserção das práticas XP – com o objetivo de garantir a qualidade de um produto de software – no *Scrum* – framework que possibilita uma empresa de software a gerir os seus projetos.

Práticas XP dentro do Scrum

O *Scrum* é um processo de produção iterativo e incremental para gerenciamento de projetos e desenvolvimento ágil de software (e de qualquer outro projeto). Geralmente o *Scrum* é adotado por uma equipe de desenvolvimento de produtos.

O *Scrum* pode ser encarado como um *framework* com objetivo de organizar um conjunto de tarefas a ser realizadas objetivando a execução de um projeto.

No *Scrum* todas as tarefas são aglutinadas na *Product Backlog* – este artefato é caracterizado como uma lista de tudo aquilo que é necessário para executar o projeto.

A *Product Backlog* é fracionada, gerando listas menores – as *Sprints Backlog*. Estas listas são inseridas nas *Sprints*.

As *Sprints* caracterizam como ciclos temporais de trabalho, a duração de cada *Sprint* é de 2 a 4 semanas.

Diariamente, os envolvidos com o projeto realizam as *Daily Scrum Meeting* – reuniões diárias com o objetivo de verificar, junto aos membros da equipe (*Scrum Team*):

○ que você fez ontem?

○ que você vai fazer hoje?

Existe algum impedimento?

Importante: Todos os membros *Scrum Team* devem participar destas reuniões.

Após o término das *Sprints*, o proprietário do produto (*Product Owner*) recebe parte do produto pronto gerado pelo projeto. É importante salientar que esta parte deve possuir um valor agregado.

Outro fator a ser destacado, o *Scrum* prevê também que todo o projeto seja gerenciado pelo *Scrum Master* (“responsável por garantir que o *Scrum Team* se orienta pelos valores e práticas do *Scrum*”).

Programação extrema (do inglês *eXtreme Programming* – (XP)), é uma metodologia ágil para equipes pequenas e médias e que irão desenvolver projetos com requisitos vagos e em constante mudança. A estratégia de constante acompanhamento e realização de vários pequenos ajustes durante o desenvolvimento de software é uma constante na metodologia.

○ XP possui práticas interessantes, este texto destaca:

Produção em Pares: O colaborador nunca trabalha sozinho. Sempre existem dois colaboradores (o piloto e o navegador) trabalhando, ao mesmo tempo, em um mesmo problema.

Padronização: A utilização de um padrão é uma prática em empresas que possuem um processo institucionalizado.

Refabricar: Esta prática recomenda tudo aquilo mal formulado sofra a nova fabricação, isto é, o artefato do projeto deve ser reescrito.

Metáfora: Formalmente, metáfora é uma figura linguística, em que há a substituição de um termo por outro, criando-se uma dualidade de significado. A metáfora é utilizada para explicar a ideia central do projeto de forma simples e objetiva. A utilização de metáforas aumenta a comunicabilidade com o cliente.

Design Simples: Desenvolva somente aquilo que foi solicitado. Não especule, a produção de especulada, na maioria das vezes, não é utilizada pelo cliente.

Jogo do Planejamento: Nas práticas ágeis as necessidades dos clientes são mapeadas em cartões de histórias. O cliente recebe um cartão, daqueles que você compra em uma livraria e nele é escrito tudo que uma determinada funcionalidade deve conter. O cliente juntamente com a equipe irá mapear o sequenciamento para o desenvolvimento das funcionalidades definidas nos cartões.

Reuniões em pé. Para agilizar o processo, todas as reuniões devem ser realizada em pé e possuírem horário de início e término. As reuniões não devem ultrapassar 30 minutos.

Após apresentar uma visão do *Scrum* e do XP, o texto propõe por meio de uma figura genérica a inserção das práticas XP dentro do *Scrum*.

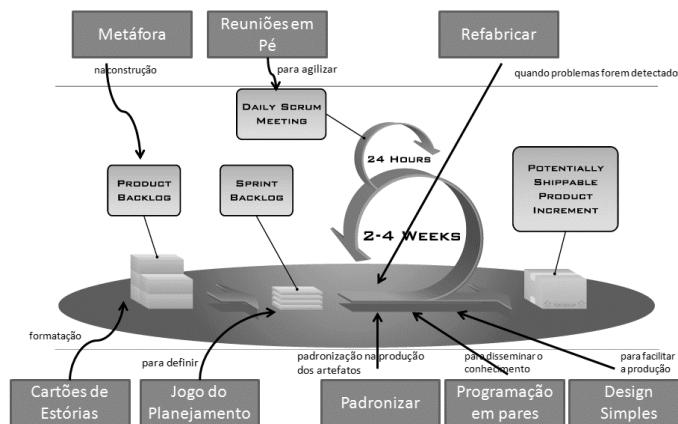


Figura 1: Inserindo práticas XP no Scrum.

Fonte: Scrum Process.

Instanciadas as Práticas XP aliadas ao *Scrum* em um projeto

Pessoal, na seção anterior o *Scrum* incorporou algumas práticas do XP. Neste texto iremos instanciar o modelo genérico (vide Figura 1) em um projeto de software. O projeto tem como meta desenvolver um software para controle de uma clínica médica.

Para apresentar o modelo instanciado vou assumir que a minha *Product Backlog* está configura e conta com os seguintes itens:

Agendar consultas; Cadastrar Cidades; Cadastrar Convênios; Cadastrar Exames; Cadastrar Laboratórios; Cadastrar Médicos; Cadastra Pacientes; Consultar Pacientes; Emitir Relatório de Pacientes por Clientes; Emitir Relatório Gerado na Consulta; Habilitar Convênios Médicos; Habilitar Exames; Realizar Pré-Consultas.

Iremos utilizar os Cartões de Estórias (vide Quadro 1) para formatar um dos itens da *Product Backlog*. Importante, em nosso projeto teremos um total 13 cartões como este. Nota: É possível utilizar as Metáforas para definir o conteúdo das Estórias (não iremos retratar como construir uma Metáfora neste texto).

Item da *Product Backlog*: Emitir Relatório Gerado na Consulta

A emissão do relatório gerado na consulta tem como objetivo apresentar ao paciente os exames que ele necessita realizar, quais são os laboratórios habilitados a realizar esses exames, os medicamentos que ele deve consumir, informações sobre o exame clínico: peso, altura, pressão, circunferência abdominal. O relatório é emitido

logo após a consulta. Os dados que constituem o relatório são gerados pelo médico durante a execução da consulta.

Quadro 1: Cartão de Estória

De posse dos cartões é necessário definir quantos (e quais) irão compor a *Sprint*. Neste momento a prática *Jogo do Planejamento* é instanciada (Quadro 2).

Jogo do planejamento
Minha equipe é capaz de produzir 4 Estórias a cada *Sprint* (duração de minha *Sprint* é de 2 semanas).
O Product Owner priorizou 5 Estórias: Emitir Relatório de Pacientes por Clientes; Emitir Relatório Gerado na Consulta; Habilitar Convênios Médicos; Habilitar Exames; Realizar Pré-Consultas.
Durante a negociação o Scrum Master argumentou que não era possível emitir qualquer tipo de relatório sem antes armazenar as informações de médicos e pacientes. O mesmo ocorre com as consultas, pré-consultas e habilitação de exames.
Scrum Master e Product Owner chegam a um acordo, a *Sprint* terá 4 Estórias: Cadastrar Cidades; Cadastrar Convênios; Cadastrar Exames; Cadastrar Laboratórios.

Quadro 2: Jogo do Planejamento

Realizado o *Jogo do Planejamento* teremos a nossa *Daily Scrum Meeting* (vide a ilustração desta reunião no quadrinhos - Figura 2). Perceba que a *Daily Scrum Meeting* é formatada seguindo as prerrogativas do *Stand-up Meeting* (reunião em pé).

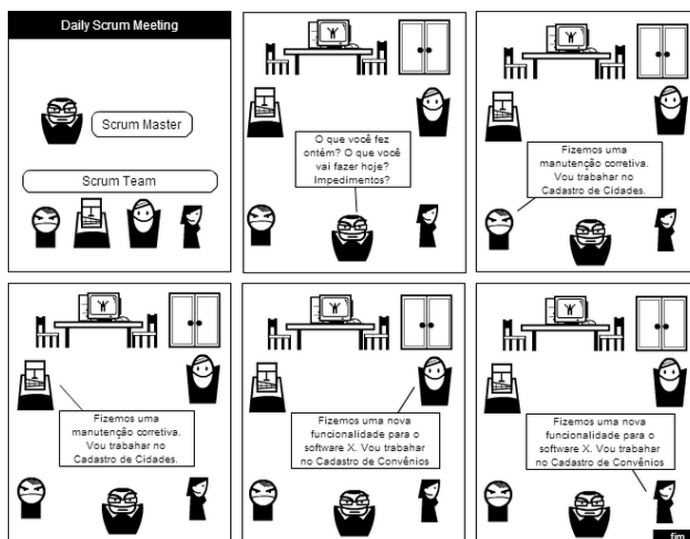


Figura 2: Quadrinhos apresentando *Daily Scrum Meeting*

Percebam (nos quadrinhos) que as respostas dos pares são as mesmas. E nenhum integrante do *Scrum Team* aponta dificuldades.

Após a realização da *Daily Scrum Meeting* os pares (*Pair Programming*) iniciam a construção das funcionalidades (*Pair Programming*) espelhadas nos Cartões Estórias, utilizando o *Java Code Convention*. Um dos Pares observou que algumas funcionalidades desenvolvidas em outro produto não respeitavam a Padronização de Código e partiram para um processo de Refabricação da referida funcionalidade. Esta percepção ocorreu porque o Par utilizou a prática de reuso para construir o Cadastro de Cidades.

Outra prática importante delineada pelo par é o *Design Simples* (esta não será ilustrada neste texto).

Ao término da *Sprint* um incremento do produto foi entregue ao *Product Owner*. Todo o ciclo de produção proposto do *Scrum* foi percorrido.

Conclusões

Este manuscrito apresentou alguns conceitos sobre gestão em projetos de software. A gestão em projetos deve incorporar a gestão da qualidade, a gestão de configuração e a gestão de expectativa entre outras. Um aprofundamento em uma das formas de gestão foi delineado – a de expectativa.

Uma proposta para incorporar as práticas do *eXtreme Programming* junto ao *framework Scrum* também foi delineada neste trabalho. Esta proposta é aplicada de forma sistemática pelo Grupo de Tecnologia da Informação – GTI do Campus de Cornélio Procópio de Universidade Tecnológica Federal do Paraná.

As formas de gestão também são trabalhadas de forma constante pelos pesquisadores o GTI, grupo liderado pelo autor deste trabalho. A união das práticas XP ao *Scrum* vem sendo institucionalizados em diversas empresas da região metropolitana de Londrina.

Enfim, existem diversas formas de focar a melhoria da gestão de projetos unindo *framework Scrum* e o *eXtreme Programming* este artigo apresentou uma, é de extrema importância que a referida união seja estabilizada dentro de um contexto empresarial específico.

Referências

Silveira Duarte, A. L´Erário; Domingues, A. L. S.; Fabri, J. A. **Proposal of a model to classify software residency environments**. Information Systems and Technologies (CISTI), 8th Iberian Conference on. Lisboa, Portugal. 2013

International Standard 15504, **Information Technology - Process Assessment**. Reference No. ISO/IEC 15504, 2004.

SCAMPI - **Standard CMMI Appraisal Method for Process Improvement**. Software. 2006.

MPS.BR - **Melhoria de Processo do Software Brasileiro**. Guia Geral. Versão 1.1, 2006.

Nonaka, I., Takeuchi, H., 1995, **The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation**. NY: Oxford University Press, 1995

- Tavares, Edval S. **Uma Contribuição para os Processos de Gerência de Projetos através da Gerência do Conhecimento**. Tese de doutorado apresentada ao Departamento de Engenharia de Produção da Escola Politécnica da USP. São Paulo. 2004.
- Terra, José C. C. **Gestão do Conhecimento: Aspectos Conceituais e Estudo Exploratório sobre as Práticas de Empresas Brasileiras**. Tese de doutorado apresentada ao Departamento de Engenharia de Produção da Escola Politécnica da USP. São Paulo. 1999.
- Ramasubramanian, Shivram; Jagadeesan, Gokulakrishnan. **Knowlodeg Management at Infosys**. in IEEE Software. USA: IEEE. may/june 2002.
- Trindade, André Luiz Presente. **Uma Contribuição para o Entendimento da Ensinagem na Preservação do Conhecimento em Ambientes de Fábrica de Software**. Tese de doutorado apresentada ao Departamento de engenharia de Produção da Escola Politécnica da USP. São Paulo 2006.
- Rus, I. & Lindvall, M. **Knowledge Management in Software Engineering**. IEEE Software, May-June 2002.
- Davenport, T., Long, D, and Beers, M. **Successful knowledge management projects**. Sloan Management, Review Winter (1998) 43-57.
- Polanyi, **Personal Knowledge: Towards a Post-Critical Philosophy**. The University of Chicago Press: Chicago, IL, 1966.
- Cohen, M. D., Burkhart, R., Dosi, G., Egidi, M., Marengo, L., Warglien, M., and Winter, S. (1996). **Routines and other recurring action patterns of organizations: Contemporary research issues**. *Industrial and Corporate Change*, (5:3), pp. 653-698.
- Kogut, B. (1991). **Country capabilities and the permeability of borders**. *Strategic Management Journal*, (12), pp. 33-47.
- H. Tsoukas, **Do We Really Understand Tacit Knowledge?** in *The Blackwell Handbook of Organizational Learning and Knowledge Management*, M. Easterby-Smith and M. A. Lyles Eds.. Oxford: Blackwell Publishing Ltd., 2003, pp. 410-427.
- T. Haldin-Herrgard, **Difficulties in Diffusing of Tacit Knowledge in Organization**. *Journal of Intellectual Capital*. Vol. 1, No. 4, pp. 357-363, 2000.
- R. M. Grant, **The Knowledge-Based View of the Firm**. in *The Strategic Management of Intellectual Capital and Organizational Knowledge*. C. W. Choo, and N. Bontis, Eds. New York: Oxford University Press, 2002, pp. 133-148.
- T. A. Stewart, **The Wealth of Knowledge: Intellectual Capital and the Twenty-first Century Organization**. Random House Inc.: New York, 2001.
- Lynn, G. S.; AKGÜK, A. E. **A new product development learning model: antecedents and consequences of declarative and procedural knowledge**. *Int. J. Technology Management*, v. 20, n.5/6/7/8, p. 490-50, 2000.
- Fabri, J.A; L´Erário, A.; Domingues, A. L. S.; Trindade, A. L. P.; Pessoa, M. S. P. **Knowledge Management and Concept Maps applied to Software Process Improvement**. 6th Iberian Conference on Information Systems and Technologies (CISTI), . Chaves, Portugal 2011.
- BATISTA, Fabio Ferreira. **Modelo de gestão do conhecimento para a administração pública brasileira: como implementar a gestão do conhecimento para produzir resultados em benefício do cidadão**. – Brasília: Ipea, 2012
- HEISIG, P. **Harmonisation of knowledage management – comparing 160 KM frameworks around the globe**. *Journal of knowledge Management*, v. 13, n. 4, p. 4-31, 2009.

SAMPAIO, A. et. al. Software Test Program. **A Software Residency Experience**. Proceedings of 27th International Conference on Software Engineering (Educational Track). St. Louis, USA. 2005.

SAMPAIO, C. A. S.; LIMA, J. M. **Residência em Software**. *Revista PROQUALIT - Qualidade na Produção de Software*. Editora UFLA. Volume 2. Número 1. Maio de 2006.