

Caracterização de Métodos Ágeis de Desenvolvimento de Software

José Fortuna Abrantes, Guilherme Horta Travassos

Universidade Federal do Rio de Janeiro – COPPE/Sistemas
Caixa Postal 68.511 – CEP 21.941-972 – Rio de Janeiro – RJ – Brasil

{jfa, ght}@cos.ufrj.br

Abstract. *This research intends to investigate agility characteristics in agile software development methods. A research protocol has been formalized and performed in order to conduct a systematic review. The data obtained were analyzed and a basic characterization for agile methods has been proposed.*

Resumo. *O objetivo deste trabalho é investigar quais são as características de agilidade no contexto de métodos ágeis de desenvolvimento de software. Um protocolo de pesquisa foi formalizado e executado para conduzir uma revisão sistemática de literatura. Os dados obtidos foram analisados e uma proposta para caracterização básica de métodos ágeis é apresentada.*

1- Introdução

A redução do ciclo de desenvolvimento foi considerada uma das principais metas de desenvolvimento de software, a partir da década de 1990. Neste cenário, Aoyama (1998) definiu agilidade em processos de software como sendo a capacidade de adaptações rápidas a mudanças, nos requisitos e no ambiente que envolve o software. Aoyama propôs um processo ágil a partir de experiências com desenvolvimento concorrente e distribuído, de lições aprendidas em fábricas de software japonesas e de conceitos em processos de produção de hardware. Foi então “cunhada” a idéia de processo ágil de software, que não significa simplesmente desenvolvimento rápido de aplicações, mas principalmente capacidade de adaptação com rapidez e flexibilidade a mudanças nos processos, nos produtos e no ambiente.

Segundo Abrahamsson et al. (2002) o que os métodos ágeis buscam não é como conter as mudanças mais cedo em um projeto de software, mas a melhor maneira de tratar mudanças inevitáveis ao longo de seu ciclo de vida. Para alcançar seu objetivo, os métodos ágeis são projetados, a princípio, para (1) produzir a primeira entrega em semanas e alcançar *feedback* rápido e mais cedo; (2) criar soluções mais simples de modo que se houver mudanças que haja mais facilidade e menor volume de alterações a serem feitas; (3) melhorar continuamente a qualidade do projeto, fazendo com que a iteração seguinte tenha menor custo de implementação; (4) testar constantemente, para detectar defeitos mais cedo e removê-los com menor custo.

O entendimento do significado de agilidade no contexto de métodos ágeis é importante para guiar a evolução de idéias e pesquisas relacionadas com práticas de teste de software compatíveis com os métodos ágeis. Tal entendimento poderá apoiar também, trabalhos relacionados com a verificação do alinhamento de características de projetos de software e alternativas de métodos ágeis para desenvolvê-los.

A questão básica de pesquisa é determinar o que caracteriza um método de desenvolvimento de software como sendo um método ágil. Pretende-se chegar a um conjunto básico de características que são necessárias para que um método possa ser classificado como método ágil, investigando através de revisão sistemática de literatura (estudo secundário) [Biolchini et al., 2005], quais são as características de agilidade no contexto de métodos ágeis. O que significa “ser ágil” para um método de desenvolvimento de software? Será adotada uma abordagem que estrutura a questão de pesquisa em 4 elementos básicos: população, intervenção, comparação e resultado [Pai et al, 2004]. Tendo em vista ser o objetivo deste estudo realizar uma caracterização da área, não haverá comparação e nem será possível aplicação de meta-análise. Desta forma, podemos definir este tipo de estudo secundário, apesar de sistemático, como uma *quasi-revisão* sistemática.

Não se pretende investigar características de métodos ágeis específicos, mas de uma maneira geral, identificar quais são as propriedades ou características desse grupo de métodos para desenvolver software e assim obter um conjunto de características desejáveis para um método ser considerado ágil. A motivação desta pesquisa é servir de apoio à busca de entendimento e solução a outras questões envolvendo atividades de processo com métodos ágeis. Este artigo contempla apenas uma síntese¹ dos resultados da revisão e está estruturado da seguinte forma: na seção 2 é apresentado um protocolo elaborado especificamente para esta revisão sistemática; na seção 3 descreve-se a execução do protocolo apresentado na seção 2; na seção 4 são apresentados e analisados os dados obtidos; na seção 5 apresenta-se uma proposta de caracterização básica de métodos ágeis de desenvolvimento de software e na seção 6 são apresentadas as conclusões.

2- Protocolo de Revisão Sistemática de Literatura

O **objetivo** é identificar as características dos métodos ágeis, de uma maneira geral. A **pergunta formulada** é: quais são as propriedades ou características dos métodos ágeis de desenvolvimento de software? O **problema** considerado é encontrar propriedades ou características de métodos ágeis. A **aplicação** da revisão sistemática é servir de base ou apoiar pesquisas envolvendo (1) atividades de processo em métodos ágeis e (2) critérios para seleção de métodos ágeis a serem aplicados em projetos de software.

A **população** considerada são os projetos de software. A **intervenção** considerada são os métodos ágeis. Não há **comparação**. E o **resultado** considerado é uma lista de propriedades ou características de agilidade de métodos de desenvolvimento de software. Foram utilizados como **controle** os seguintes artigos: [Miller, 2001; Abrahamsson et al, 2002; Lindvall et al, 2002].

As **fontes** selecionadas foram as bases de dados eletrônicas, disponíveis no portal CAPES, incluindo documentos indexados por *Compendex EI*, *IEEEExplore*, *Inspec*, *Web of Science* e *ACM digital library*. O **idioma** escolhido foi o Inglês, por ser maioria nas bases de dados pesquisadas. Além disso, textos em português, embora reconheça-se a sua importância, muitas vezes não se encontram indexados, o que aumenta o esforço ou impede sua busca. Considerou-se qualquer **tipo de trabalho** ou artigo que fizesse abordagem sobre características de métodos ágeis.

¹ Relatório completo disponível em http://www.cos.ufrj.br/~jfa/agil_caracterizacao.pdf

As **palavras-chave** escolhidas para a população foram *software, development, project, system, application, engineering, building e implementation*; para intervenção foram *agile, method, adaptive, rapid, approach, technique, environment, process, practice, methodology*; e para resultado foram *characteristic, attribute, property, feature, characterization, aspect, idea, factor, dimension, driver, perspective, requirement*.

Como **critérios de inclusão e exclusão** considerou-se que os documentos devem estar disponíveis na *web*, e contemplar características de agilidade em métodos ágeis. Como **estratégia de extração de informações**, considerou-se que para cada artigo selecionado, serão extraídas as seguintes informações: título do documento, autor(es), fonte, ano de publicação e propriedades ou características de agilidade.

Na medida do possível, a **string de busca** será a mesma para todas as máquinas de busca. Contudo, poderá haver adaptações para se adequar a restrições de máquinas de busca específicas, observando-se as seguintes diretrizes: (1) a *string* derivada deverá ser logicamente equivalente à *string* original, ou (2) na impossibilidade de se manter equivalência exata, deverá a *string* derivada ser mais abrangente para evitar perda de documentos potencialmente relevantes. A figura 1 mostra a *string* de base utilizada nas máquinas de busca. Nesta *string*, o operador NEAR é um recurso oferecido pelas máquinas de busca para estabelecer que uma palavra-chave deve ocorrer próxima à outra palavra-chave.

```
(software NEAR0 development OR software NEAR1 engineering OR software NEAR0 building OR software NEAR0 implementation OR software NEAR0 projects OR software NEAR0 systems OR software NEAR0 application OR system NEAR0 development OR system NEAR0 engineering OR system NEAR0 building OR system NEAR0 implementation OR system NEAR0 project OR application NEAR0 development OR application NEAR0 engineering OR application NEAR0 building OR application NEAR0 implementation OR application NEAR0 project) AND ((agile OR adaptive OR rapid) AND (method OR process OR practice OR methodolog OR approach OR technique OR environment)) AND ((agile) AND (characteristic OR attribute OR propert OR feature OR characterization OR aspect OR idea OR factor OR dimension OR driver OR perspective OR requirement))
```

Figura 1 String de base utilizada nas buscas

3- Execução de Buscas

As buscas foram realizadas utilizando máquinas de busca de editoras ou bibliotecas digitais disponíveis no portal CAPES. As buscas, em todas as máquinas, com exceção da *ACM digital library*, foram efetuadas com a opção *autostemming* ligada e com utilização do operador *NEAR* parametrizado. Na *Compendex EI* a busca não apresentou problemas, a *string* foi processada integralmente e não precisou ser ajustada. Nas demais editoras/bibliotecas digitais as *strings* de busca apresentaram problemas e tiveram que ser reformuladas.

Foram recuperadas 1016 referências, sendo 303 para *Compendex EI*, 299 para *IeeeXplore*, 250 para *Inspec*, 45 para *Web of Science* e 119 para *ACM digital library*. Para apoio à manipulação e tratamento dos itens recuperados foi utilizado o gerenciador de referências JabRef versão 2.0.1 ©2006.

As réplicas foram eliminadas, mantendo-se o artigo remanescente contabilizado para a biblioteca digital com maior quantidade de itens recuperados. Todos os controles foram recuperados (a *Compendex EI* recuperou dois deles e a *Inspec* recuperou o outro).

Em uma primeira avaliação superficial (título e *abstract*) foram excluídas as referências que nitidamente tratavam de outros assuntos não pertinentes à pesquisa.

Posteriormente, em uma avaliação mais apurada e detalhada, foram selecionados os documentos candidatos a fazer parte da revisão sistemática: total de 252 referências, sendo 86 para *Compendex EI*, 113 para *IeeeXplore*, 42 para *Inspec*, 4 para *Web of Science* e 7 para *ACM digital library*. Das 252 referências incluídas, 204 eram da área de software, 42 da área de negócios/manufatura e 6 da área de eletrônica.

Dos 252 documentos a serem lidos, foram identificados como prioritários, 106 documentos, com a seguinte distribuição por área: eletrônica com 6 documentos, negócios/manufatura com 42 documentos e métodos ágeis de desenvolvimento de software com 58 documentos. A priorização se deu a partir da leitura minuciosa dos resumos e da parte introdutória, identificado-se os artigos que apresentavam maior probabilidade de contribuir para a revisão sistemática. No processo de priorização dos artigos a serem lidos utilizou-se como apoio o gerenciador de referências JabRef.

No total foram lidos e analisados 182 documentos. A grande maioria deles não trazia uma contribuição direta para caracterizar métodos ágeis ou não traziam significado para as características neles mencionadas. Apenas 12 documentos puderam ser aproveitados para a revisão sistemática. Considerou-se 3 *clusters* de documentos, sendo cada *cluster* computado uma única vez na incidência das características nos artigos, para evitar a repetição de contagem de características similares.

Na formação dos *clusters* para fins de contagem, foram incluídos no mesmo *cluster* dois documentos tratando da mesma característica sendo que o autor ou alguns autores de um deles faz parte do conjunto de autores do outro. Também foram incluídos no mesmo *cluster*, dois documentos tratando das mesmas características, sendo que um deles referencia o outro.

As propriedades ou características de métodos ágeis de desenvolvimento de software capturadas em 12 dos documentos incluídos [Abrahamsson et al, 2003; Boehm e Turner, 2004a; Meso e Jain, 2006; Holmstrom et al, 2006; Miller, 2001; Aoyama, 1998; Coram e Bohner, 2005; Hansson et al, 2006; Cockburn, 2002; Abrahamsson et al, 2002; Lindvall et al, 2002; Miller, 2003;] e estão descritas na Tabela 1.

Tabela 1 – Características Identificadas para os Métodos Ágeis

CARACTERÍSTICA	INTERPRETAÇÃO
Incrementalidade	Não tentar construir o sistema todo de uma só vez; o sistema é partido em incrementos (pequenas <i>releases</i> com novas funcionalidades) que podem ser desenvolvidos em paralelo em ciclos rápidos; quando o incremento é completado e testado, ele é integrado ao sistema.
Cooperação	Interação aberta e com proximidade entre os vários <i>stakeholders</i> (especialmente entre cliente e desenvolvedores); o cliente deve tomar parte ativa no processo de desenvolvimento e prover <i>feedback</i> de forma regular e freqüente.
Transparência ou Clareza	O método é fácil de aprender e modificar e é suficientemente documentado.
Adaptabilidade	Habilidade e capacidade de adaptação rápida do processo para atender e reagir a mudanças de última hora nos requisitos e/ou no ambiente, ou a situações ou riscos não previstos inicialmente.
Iteratividade	Envolve vários ciclos curtos, dirigidos por características do produto, nos quais certo conjunto de atividades é completado em poucas semanas; estes ciclos são repetidos muitas vezes para refinar as entregas.
Auto-organização	As equipes determinam o melhor modo de trabalhar; a equipe tem autonomia para se organizar da melhor forma para completar os itens de trabalho.
Emergência	Os processos, princípios, estruturas de trabalho são reconhecidos durante o projeto ao invés de serem pré-determinados; permite-se que tecnologia e

	requisitos emergem ao longo do ciclo de vida do produto.
Períodos de reflexão e introspecção	Reuniões no fim de cada subprojeto ou iteração para os membros da equipe discutirem o que eles estão fazendo bem e o que precisa ser mudado.
Incorporação de <i>feedback</i> rápido	Equipes capazes de procurar e receber continuamente, <i>feedback</i> de modo mais freqüente e com mais rapidez (teoria dos sistemas adaptativos complexos).
Modularidade	Característica que permite que um processo seja quebrado em componentes chamados de atividades; modularidade permite que atividades sejam adicionadas ou removidas de um processo quando necessário.
Restrição de Prazo	Estabelecimento de limite de tempo para cada iteração programada. Grandes volumes de desenvolvimento são quebrados em múltiplas entregas que possam ser desenvolvidas incremental e concorrentemente de modo previsível.
Parcimônia²	Eliminação de perdas ou habilidade de fazer mais com menos recursos; característica que o processo ágil tem, de requerer o mínimo necessário de atividades para mitigar riscos e alcançar metas; deve-se remover todas as atividades desnecessárias no processo de desenvolvimento.
Convergência	Ataque efetivo a todos os riscos que devem ser considerados; como resultado o sistema se torna mais próximo da realidade buscada a cada iteração; à medida que os riscos são atacados pró-ativamente, o sistema está sendo entregue em incrementos.
Orientação a pessoas	Favorecimento de pessoas sobre processos e tecnologias; desenvolvedores são encorajados a aumentar sua produtividade, qualidade e desempenho; a comunicação e a cooperação dentro das equipes de desenvolvimento são consideradas fundamentais e necessárias. As reuniões diárias em pé e os <i>workshops</i> de reflexão dão às pessoas a chance de manifestar suas preocupações.
Colaboratividade	É uma atitude entre membros da equipe de desenvolvimento, entre os quais se encoraja a comunicação para disseminar informação e apoiar integração rápida de incrementos.
Equipes pequenas	O pequeno número de equipes por projeto é necessário para promover o ambiente colaborativo e por requerer menos planejamento para coordenar as atividades dos membros das equipes.
Testes constantes	Para prevenir a degradação da qualidade devido a entregas muito curtas, dá-se alta ênfase a testes do produto ao longo do ciclo de vida. Métodos ágeis requerem testes de integração ao longo do processo de desenvolvimento. Automação dos testes é importante para que as “ <i>builds</i> ” diárias passem por testes de regressão.
Equipes Locais	Para algumas metodologias significa trabalhar na mesma sala ou em salas adjacentes, o que só funciona para equipes de 8 a no máximo 14 pessoas. Todas as metodologias são sensíveis à localização da equipe, pois estão fortemente fundamentadas em canais de comunicação rápidos e ricos, que permitem reduzir a documentação externa a ser construída e mantida.
Cortesia	Utilização de atividades encadeadas para validar e melhorar os produtos de trabalho das atividades anteriores; atividades de cortesia (<i>complimentary</i>) são atividades que trabalham juntas para produzir um resultado melhor do que produziriam individualmente (ex.: escrever <i>user stories</i> , criar testes de aceitação).

4- Análise dos Dados Obtidos

Computou-se a quantidade de artigos em que a característica foi abordada, sendo que nesta contagem cada *cluster* foi computado uma única vez. Após interpretação do significado descrito para cada ocorrência de característica, foram identificadas as similaridades e contadas as respectivas presenças nos documentos ou artigos incluídos na revisão sistemática. As revisões de código consideradas por Coram e Bohner (2005) foram interpretadas mais como prática do que característica, razão pela qual foram

² “Enxutez” ou “Leveza” (*Leanness*)

excluídas. Cada *cluster* ficou com 2 documentos, sendo a formação de cada um deles conforme se segue: *cluster* 1 [Abrahamsson et al, 2003 e Abrahamsson et al, 2002]; *cluster* 2 [Boehm e Turner, 2004a e Lindvall et al, 2002]; *cluster* 3 [Miller, 2001 e Miller, 2006].

Observando-se a distribuição das 19 características encontradas, de acordo com a incidência de presença em 9 artigos após a exclusão das similaridades e consideração dos *clusters*, constata-se que as características mais freqüentes foram Adaptabilidade, Incrementalidade e Iteratividade, com 55,56 % de incidência nos artigos para as duas primeiras e 44,44 % para a Iteratividade. Dentre estas, a adaptabilidade parece estar mais diretamente alinhada com um dos valores do Manifesto Ágil publicado em 2001: “Valorizar mais a resposta a mudanças do que seguir um plano”.

As características com uma freqüência média de presença nos artigos foram Colaboratividade, Cooperação, Orientação à Pessoas, Parcimônia e Restrição de Prazo. Dentre estas, parecem estar mais alinhadas com valores especificados no Manifesto Ágil: cooperação – associada com “valorizar mais a colaboração do cliente do que negociação de contratos”; orientação à pessoas – associada com “valorizar mais indivíduos e interações do que processos e ferramentas”. Parcimônia – associada com “valorizar mais software funcionando do que documentação abrangente”.

As características menos freqüentes nos artigos foram Incorporação de Feedback, Modularidade, Reflexão/Introspecção, Auto-organização, Convergência, Cortesia, Emergência, Equipes Locais, Equipes Pequenas, Testes Constantes e Transparência.

Dentre as 19 características identificadas, uma foi contemplada por 3 *clusters* (incrementalidade), duas foram contempladas por 2 *clusters* (adaptabilidade e iteratividade), dez foram contempladas por apenas 1 *cluster* (cooperação, transparência, auto-organização, emergência, modularidade, restrição de prazo, parcimônia, convergência, orientação a pessoas e colaboratividade) e 6 não têm participação em qualquer *cluster* (reflexão e introspecção, *feedback* rápido, equipes pequenas, equipes locais e cortesia).

Observa-se que as características de Incrementalidade, Adaptabilidade e Iteratividade permanecem com maior destaque também quando se considera a quantidade de *clusters* associados às características identificadas nos artigos. Já as características de Auto-organização, Convergência, Emergência e Transparência que estavam na faixa de baixa incidência nos artigos, tiveram um *cluster* associado a cada uma delas.

A distribuição das características abordadas, por faixa de ano de publicação, permite constatar que dentre aquelas com maior amplitude de tempo e maior densidade de publicações no período estão a adaptabilidade, a incrementalidade, a iteratividade e a cooperação. Observa-se que algumas características foram preocupações apenas pontuais no período. Com exceção do artigo publicado por Aoyama (1998) todos os demais foram publicados a partir da declaração do Manifesto Ágil.

5- Proposta de Caracterização Básica para Métodos Ágeis

Para se alcançar a agilidade de métodos de desenvolvimento de software, algumas características podem ser consideradas fundamentais. Dentro do contexto apresentado, a

característica fundamental e mais aderente aos valores do Manifesto Ágil parece ser a adaptabilidade. Em seguida, e considerando um aspecto mais pragmático, seguem as características de incrementalidade e iteratividade que devem caminhar juntas. A essas características devem se somar outras três que estão diretamente associadas a uma idéia central dos métodos ágeis que é o foco nas pessoas que conduzem as atividades dos processos. São elas: colaboratividade, cooperação e orientação à pessoas. As características de colaboratividade e cooperação às vezes são confundidas uma com a outra na literatura, dependendo da interpretação que se faça para o seu significado. Contudo, nesta *quasi-revisão* sistemática não foi feita a mescla destas duas características, tentando respeitar as idéias dos autores, além de não se ter encontrado indicativos que permitissem fazer tal mescla de forma segura. A diferença entre as duas é sutil e não está destacada nos textos dos artigos recuperados. A idéia da cooperação parece ter ligação mais direta com a interação do cliente com o grupo desenvolvedor e está associada ao papel do *feedback* constante [Abrahamsson et al, 2003; Meso e Radhika, 2006; Hansson et al, 2006]. Já a idéia de colaboratividade parece estar associada com a interação entre os membros do grupo desenvolvedor e relacionada com a integração contínua de novos incrementos de software, com funcionalidades modificadas ou novas [Holmstrom et al, 2006; Miller, 2001; Coram e Bohner, 2005; Miller, 2003]. Deve-se acrescentar também, duas características que além de necessárias para o sucesso de um método ágil, são fundamentais em qualquer processo: são elas: parcimônia (*leanness*) e restrição de prazo. Convém observar que as entregas devem ser feitas com rapidez, mas têm que agregar valor para o cliente, através de resultados práticos ou efetivos.

As demais características elencadas nesta revisão também são importantes para o sucesso de um método ágil. Contudo, pode-se considerá-las como decorrentes ou subsidiárias daquelas citadas como fundamentais. Desta forma, baseado nas informações obtidas através deste estudo secundário, propõe-se a seguinte caracterização:

“Um método para ser caracterizado ágil, deve apresentar, em um grau adequado ao contexto de desenvolvimento de software em que se insere, as características de adaptabilidade, incrementalidade, iteratividade, colaboratividade, cooperação, orientação à pessoas, parcimônia (leanness) e restrição de prazo.”

6- Conclusões

A abordagem ágil, ao invés de resistir à mudança, busca acomodá-la o mais fácil e eficientemente possível, mantendo consciência de possíveis conseqüências. Dentre as características propostas, cada uma tem o seu papel e todas concorrem para se atingir os objetivos de satisfação do cliente, sendo que a adaptabilidade ocupa um lugar de destaque dentre o que é desejável para um método ágil de desenvolvimento de software.

Espera-se que a caracterização acima, resultado de estudo secundário (*quasi-revisão* sistemática) cuja síntese foi apresentada neste artigo, possa permitir algum direcionamento nos trabalhos e pesquisas relacionadas a obtenção de agilidade em métodos de desenvolvimento de software. Em particular, o interesse dos autores está relacionado a obtenção de agilidade em processos e métodos associados a testes de software, principalmente quando aplicados no contexto da engenharia de aplicações *web*. Como foi possível perceber, as características relacionadas com testes de software

não tiveram muita incidência nas abordagens apresentadas nos diversos artigos. Contudo, conforme Lindvall et al (2002) enfatizam, os testes são uma questão fundamental em métodos ágeis, fortalecendo a idéia da necessidade de pesquisas nesta direção.

Referências Bibliográficas

- Abrahamsson, P. Salo, O. Ronkainen, J. Warsta, J. (2002) “Agile Software Development Methods. Review and Analysis”, Espoo. VTT Publications 478.
- Abrahamsson, P.; Warsta, J.; Siponen, M.T. & Ronkainen, J. (2003) “New directions on agile methods: a comparative analysis”, IEEE Computer Society, 244-254
- Aoyama, Mikio. (1998) “Agile Software Process and Its Experience”, Proceedings of the 1998 International Conference on Software Engineering, p. 3-12.
- Biolchini et al. (2005) “Systematic Review in Software Engineering”, COPPE / UFRJ, Relatório Técnico, ES-679/05.
- Boehm, B. & Turner, R. (2004a) “Balancing agility and discipline: A Guide for the Perplexed”, Pearson Education Inc, Boston, MA.
- Cockburn, A. (2002) “Agile Software Development Joins the ‘Would be’ Crowd”, Cutter IT Journal, v.15, n.2.
- Coram, Michael. Bohner, Shawn. (2005) “The Impact of Agile Methods on Software Project Management”, Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS’05).
- Hansson, C. Dittrich, Y. Gustafsson, B. Zarnak, S. (2006) “How agile are industrial software development practices?”, The Journal of Systems and Software 79, 1295–1311.
- Holmstrom, Helena. Fitzgerald, Brian. et al. (2006) “Contemporary practices in systems development. Agile Practices Reduce Distance in Global Software Development”, Information Systems Management, summer.
- Lindvall, M. Basili, V. et al. (2002) “Empirical Findings in Agile Methods”, In: Proceedings of Extreme Programming and Agile Methods – SP/Agile Universe, pp. 197-207.
- Meso, Peter. Jain, Radhika. (2006) “Contemporary practices in systems development. Agile Software Development: adaptive systems principles and best practices”, Information Systems Management, summer.
- Miller, Granville G. (2001) “The Characteristics of Agile Software Processes”, Proceedings of the 39th Int’l Conf. and Exhibition on Technology of Object-Oriented Languages and Systems (TOOLS’01).
- Miller, R. (2003) “The Dynamics of Agile Software Processes, Part I: Characteristics”, Disponível em <http://bdn.borland.com/article/29726/> acessado em 13/12/2006.
- Pai, M. McCulloch, M. Gorman, J.D. et al. (2004) “Systematic Reviews and meta-analyses: An illustrated, step-by-step guide”, The National Medical Journal of India, vol. 17, n.2.