

Usando PBL na Qualificação de Profissionais em Engenharia de Software

Simone C. dos Santos, Maria da Conceição Moraes Batista, Ana Paula Cavalcanti, Jones O. Albuquerque, Silvio Meira

CESAR.EDU – Centro de Estudos e Sistemas Avançados do Recife
(C.E.S.A.R)

simone.santos@cesar.org.br, ceca.moraes@cesar.org.br, ana.paula@cesar.org.br,
jones@cesar.edu.br, silvio@cesar.org.br

Abstract. This article presents an educational methodology based-on PBL (Problem Based Learning) to improve the effectiveness of learning in software engineering, promoting the ability of students to solve real problems into software factories environments. This methodology has been applied in a professional master course in Software Engineering on three classes of this program, since 2007. The use of PBL showed a growth of the software factories performance and a reduction of the difference among the student's grades of 3,58/10 for 0,83/10 to the end of the first semester. This course is run by C.E.S.A.R (<http://www.cesar.org.br>), a research institute with experience in development of innovative software.

Keywords: PBL, Software Engineering, Software Factories, Master Course

Resumo. Este artigo apresenta uma metodologia de ensino baseada em PBL (aprendizagem baseada em problemas) com o objetivo de aprimorar a efetividade do aprendizado em Engenharia de Software, promovendo a habilidade de estudantes para resolver problemas reais dentro de ambientes de fábricas de software. Esta metodologia tem sido aplicada no Mestrado Profissional em Engenharia de Software (MPES) em três turmas, desde 2007. O uso de PBL mostrou um crescimento no desempenho das fábricas e uma redução da diferença entre os rendimentos dos estudantes de 3,58/10 para 0,83/10 ao final do primeiro semestre. Este curso é executado pelo C.E.S.A.R, instituto de pesquisa com experiência em desenvolvimento de software com inovação.

Palavras-chave: PBL, Engenharia de Software, Fábricas de software, Mestrado Profissional

1 Introdução

O crescimento do mercado de Tecnologia da Informação (TI) e seu papel cada vez mais importante na criação de vantagem competitiva das organizações têm gerado uma forte demanda por novas e diferentes maneiras de se produzir software com alta qualidade, flexibilidade e ganhos de produtividade. Em particular, esta demanda se expressa através de requisitos como tempo de resposta ao mercado (time-to-market) de soluções de TI altamente reduzido e necessidade de eficiência e eficácia de soluções, exigindo inovação na aplicação de TI para melhores e mais rápidos retornos. Claramente estes requisitos não podem ser alcançados sem profissionais com competências (conhecimentos, habilidades e aptidões individuais) diferenciadas, capazes de ter uma visão ampla de problemas sob aspectos gerenciais, interpessoais e de negócios, além dos aspectos tecnológicos.

Estudos sobre os objetivos da educação superior em TI têm cada vez mais relacionado o programa de qualificação e sua metodologia de ensino às futuras carreiras profissionais dos estudantes. Em [TYNÄLÄ, 1999], o autor discute que a educação superior deveria ter como objetivos principais o treinamento de profissionais, o aprendizado contínuo e a preparação para a prática profissional. Esta abordagem pode ser aplicada integrando teoria e prática dentro de um currículo que promova a aquisição de conhecimento geral e específico para resolução de problemas reais de mercado.

Neste contexto, a proposta deste artigo é apresentar e discutir os resultados da aplicação do processo instrucional de aprendizagem baseado em problemas reais (PBL - Problem Based Learning) [NING, 1995] em um curso de Mestrado Profissional em Engenharia de Software (MPES) como uma forma eficaz de aprendizagem.

Para a implementação do método PBL, o curso provê um ambiente baseado em Fábricas de Software [MORAES, MEIRA e ALBUQUERQUE, 2003], no qual os estudantes estão imersos em projetos práticos de desenvolvimento de software de forma sistemática e controlada por processos de qualidade. A avaliação do aprendizado é conduzida de forma que o trabalho coletivo e as habilidades individuais sejam complementares e reflitam diretamente no desempenho do estudante. Resultados da aplicação desta metodologia na primeira turma do mestrado são apresentados ao final do artigo.

Este artigo está organizado em cinco seções. A Seção 2 discute sobre a motivação do uso do método PBL no ensino de Engenharia de Software. A Seção 3 descreve a metodologia proposta para o ensino de Engenharia de Software. Alguns resultados da aplicação da metodologia na primeira turma do MPES são apresentados na Seção 4. Finalmente, a Seção 5 apresenta as conclusões e considerações finais.

2 Aprendizagem baseada em Problemas Reais

Desenvolvido na educação médica na década de 70, o método PBL [NING, 1995] tem sido adaptado em um número crescente de áreas de atuação, incluindo a Engenharia, e em diferentes níveis educacionais [RIBEIRO e MISUKAMI, 2005], [GÜZELIS, 2006].

Sob uma ótica bem simples, PBL pode ser definido como um método instrucional que usa um problema para iniciar, direcionar e motivar o aprendizado. Como um método instrucional, PBL é consistente com os princípios da abordagem construtivista, que defende que *“o que as pessoas entendem é uma função do conteúdo, contexto, atividades e objetivos do aprendiz”* [SAVERY and DUFFY, 1995]. Entretanto, o

desenvolvimento de um processo efetivo para resolução de problemas é só um dos objetivos do PBL. Em [RIBEIRO e MISUKAMI, 2005], os autores enfatizam:

“Este método também pretende apoiar os estudantes na aquisição de uma base de conhecimento estruturada em torno de problemas da vida real e no desenvolvimento de competências e atitudes, incluindo trabalho em equipe e habilidades de auto-aprendizado, cooperação, ética e respeito aos pontos de vista de outras pessoas.”

Em [PETERSON, 1997], o autor ressalta três importantes critérios que promovem um aprendizado mais eficaz com o uso de PBL:

1. O aprendizado acontece em um ambiente onde os estudantes estão imersos na prática, em atividades em que recebem feedback de seus colegas estudantes e professores;
2. Os estudantes recebem guias e suporte de seus pares, de maneira a promover um ensino multi-direcional envolvendo outros estudantes, professores e monitores, diferentemente do ensino convencional, normalmente unidirecional (professor para estudante);
3. O aprendizado é funcional, a partir de problemas reais.

A adoção de PBL não é uma tarefa fácil, principalmente porque propõe a quebra do paradigma do ensino convencional já tão enraizado em nossa cultura. Exige mudança de postura tanto do professor como do estudante, trabalhando uma forma de aprendizagem altamente investigativa e questionadora, além de uma certa maturidade profissional. Em [WATERS and MCCRACKEN, 1996], os autores descrevem uma experiência do uso de PBL em Engenharia de Software, em um curso de graduação em Sistemas de Gerenciamento da Informação. A classe foi dividida em 5 grupos de 6 pessoas e um projeto de sistema de leilão interativo foi eleito como estudo de caso. Nesta experiência, uma das principais dificuldades já se apresentou na etapa de levantamento de requisitos, quando os estudantes se preocupavam em rapidamente desenvolver o software a partir de requisitos mal levantados e pobremente documentados, ao invés de entenderem bem os problemas a serem resolvidos e suas implicações nos requisitos de software.

Com base nos princípios de Peterson e nas experiências de Waters, este artigo propõe uma metodologia para o ensino de profissionais da indústria de TI, em particular, para os profissionais envolvidos no ensino de pós-graduação, uma vez que estudantes neste nível possuem alguma experiência anterior com desenvolvimento de software e, portanto, com um nível de maturidade diferente dos estudantes que ainda se encontram no estágio de graduação, sedimentando conceitos e fundamentos que serão pré-requisitos em um curso de pós-graduação.

3 Implementação de PBL com Fábricas de Software

A metodologia tem como principal fundamento o uso de fábricas de software para criar o ambiente onde os estudantes estarão imersos na prática e em contínua interatividade com professores e colegas estudantes, como destaca o Princípio 1 de Peterson.

Com o objetivo de sistematizar a criação das fábricas de software, alguns elementos-chave foram identificados com foco nos outros dois princípios de Peterson, como ilustra a Figura 1: (1) Capital Humano e (2) Conteúdo, no que diz respeito ao ensino multidirecional e guias e suporte ao capital humano envolvido e; (3) Projetos Reais, com foco na necessidade de um aprendizado funcional. A partir destes

elementos, um processo de construção de fábricas foi definido tendo como principal referência o trabalho de [MORAES, MEIRA e ALBUQUERQUE, 2003]. As seções subsequentes discutem brevemente estes aspectos, considerando o contexto do Mestrado Profissional em Engenharia de Software (MPES).

3.1 Elementos-chave na Criação e Execução das Fábricas de Software

A metodologia identifica quatro grupos de capital humano envolvidos no processo e criação das fábricas de software: o *time* que compõe uma fábrica, responsável pela definição do negócio da fábrica e desenvolvimento de software; os *tutores/professores*, responsáveis por ministrar as aulas, executando o papel de consultores especialistas em áreas de sua disciplina e facilitadores no processo de aprendizado; os *monitores de fábrica*, responsáveis pelo acompanhamento de todas as fábricas quanto ao atendimento de objetivos aos quais elas se comprometeram, sob os pontos de vista de processos e produtos; os *clientes*, responsáveis pela demanda do software, validação de requisitos e dos resultados acordados com as fábricas de software.

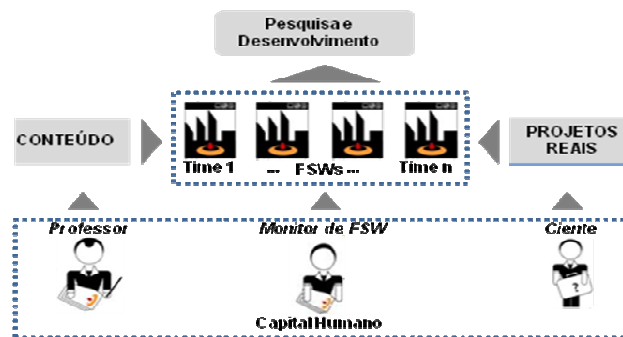


Figura 1: Elementos-chave na criação e execução das fábricas de software.

Quanto ao conteúdo, as disciplinas constituem a base conceitual da metodologia, subsidiando o time, professores, monitores e clientes em conhecimentos para a resolução dos problemas. No contexto de Engenharia de Software, as disciplinas e suas ementas usam como referência o guia SWEBOK (Software Engineering Body of Knowledge - <http://www.swebok.org/>), uma iniciativa do IEEE Computer Society. O SWEBOK está subdividido em dez áreas de conhecimento em Engenharia de Software: Gerência de Configuração de Software; Gerência de Projetos de Software; Processo de Engenharia de Software; Ferramentas e Métodos; Qualidade de Software; Requisitos de Software; Design de Software; Construção de Software; Teste de Software; Manutenção de Software. Tendo como base as disciplinas do SWEBOK, o MPES trabalha com sete disciplinas durante a etapa da vivência das fábricas de software, associadas de acordo com a Tabela 1.

O programa foi projetado para executar as disciplinas em módulos semanais de 27 horas/aula cada, sendo 15 horas de aulas conceituais, 12 horas de práticas guiadas e uma avaliação individual ao final da semana. As horas de práticas são utilizadas em atividades para ajudar os alunos a aplicar a teoria vista em sala de aula nos projetos das fábricas de software, e essas tarefas são apoiadas por docentes que atuam como consultores especialistas nas atividades. Um pool de projetos é criado para cada turma que se inicia.

Um projeto, para participar do pool, deve cumprir alguns requisitos: deve ter características de inovação; relevância para a indústria; modelo de negócios aplicado e; ter um processo de desenvolvimento de software que permeie todo o ciclo de

desenvolvimento. Vale salientar que, os projetos do pool de projetos são realmente projetos reais, portanto, possuem características diferentes um dos outros, mesmo que concentrados em uma mesma área de conhecimento.

Tabela 1. Relação entre disciplinas do MPES e SWEBOK

MPES	SWEBOK
1. Fábrica de Software	Processos de ES; Ferramentas e Métodos
2. Requisitos e de Interface	Requisitos de Software
3. Gestão de Projetos	Gerência de Projetos de Software
4. Arquitetura	Design de Software
5. Engenharia de Reuso	Design de Software; Construção de Software
6. Interoperabilidade	Design de Software; Construção de Software
7. Verificação e Validação	Testes; Manutenção; Qualidade de Software

Os projetos precisam ainda ter um cliente real envolvido. Os clientes motivam os estudantes a se envolverem no projeto para a solução dos problemas. É importante ressaltar que, se o estudante se engaja em uma autêntica resolução de problemas, então ele é o “dono” do problema. Os aprendizes percebem o problema como real quando existe relevância pessoal de cada indivíduo envolvido [SAVERY and DUFFY, 1995].

3.2 Processo de Construção das Fábricas de Software

O processo de construção das fábricas de software se dá como atividades de avaliação da disciplina Fábrica de Software, na qual os estudantes serão orientados a compor seus grupos e assim estruturar suas equipes para definirem e formalizarem suas respectivas fábricas de software. Este processo é exercitado desde 2003 [MORAES, MEIRA e ALBUQUERQUE, 2003] e tem-se mostrado eficiente como um mecanismo de organização de equipes, definição de um processo de desenvolvimento de software e gestão de projetos associados a estas equipes. O processo de construção é composto por sete macro-atividades:

1. *Divisão dos Estudantes em Equipes:* A metodologia recomenda equipes de 5 a 7 integrantes, cada uma representando uma fábrica de software. A divisão de equipes considera afinidades, competências e habilidades de cada estudante para que se evite concentração e/ou escassez de competências/habilidades em uma mesma equipe. Para cada novo projeto, é necessário alocar os papéis entre os membros da fábrica, tais como: gerente de projeto, arquiteto de software, engenheiro de software, analista de requisitos, engenheiro de testes e de processos, entre outros. Considerando o tamanho da equipe, normalmente cada integrante precisará desempenhar mais de um papel neste processo. É importante ainda salientar que, apesar do seu papel, cada estudante se envolve em todas as etapas do processo de desenvolvimento de software conduzidas pelas disciplinas.
2. *Estudo e Apropriação dos Modelos de Fábricas de Software Disponíveis:* Este estudo se dá em modelos práticos apresentados pelos professores no decorrer do curso. Uma vez entendido os processos de desenvolvimento de software destas fábricas, estas precisarão definir quais características dos processos estudados serão incorporadas a sua realidade.

3. *Definição do Ciclo de Vida do Software na Fábrica:* Esta etapa inclui a definição do processo de desenvolvimento de software, fluxo de atividades, papéis, artefatos e templates. Uma fábrica de software precisa de uma plataforma de processo de desenvolvimento de software configurável baseada em conceitos sólidos assim como as melhores práticas encontradas no mercado.
4. *Organização das Fábricas em sua Infra-estrutura:* Este passo inclui se apropriar da infra-estrutura necessária para o funcionamento da fábrica: site da fábrica, repositório de templates de artefatos, contratos de nível de serviço, métricas de desenvolvimento e contratação, ambientes de desenvolvimento e de produção da fábrica, garantia da qualidade dos produtos e processos da fábrica.
5. *Definição do Modelo de Negócio da Fábrica:* Aqui são definidos os tipos de licença de software utilizados pela fábrica em seus contratos, a política de propriedade intelectual dos produtos gerados pela fábrica, o modelo de comunicação com os clientes, o modelo de contratação e os templates de contrato comercial e técnico da fábrica.
6. *Definição da Política de Avaliação Interna da Fábrica:* Esta etapa contempla a avaliação dos processos e do capital humano como executor deste processo, bem como definição da política de contratação e renovação/capacitação do capital humano da fábrica.
7. *Política de avaliação do Cliente:* Nesta etapa cada fábrica precisará definir a política de avaliação de satisfação do cliente, definindo critérios de aceitação de produtos e processos, além de estratégias de melhoria para os itens avaliados.

Observa-se, do processo de construção de fábricas de software, que suas atividades apenas guiam o estudante para que ele próprio efetive suas definições e organização de equipes necessárias às suas respectivas fábricas. Assim, tem-se como componente pedagógico, além do aspecto prático-experimental, a inovação, uma vez que o processo não limita nem impõe restrições à criatividade das equipes em suas definições.

Uma vez construídas as fábricas de software, com seus times, projetos e clientes reais, planejamento de disciplinas e professores alocados, as fábricas seguem a execução de seus projetos, acompanhadas pelos professores das disciplinas e pelos monitores das fábricas. Estes últimos avaliam o processo conduzido e produtos gerados, por meio de reuniões periódicas de acompanhamento (status report), análise de artefatos gerados e avaliações individuais de desempenho. O número de reuniões de acompanhamento é definido com base no número de iterações dos projetos de software desenvolvidos nas fábricas, planejados antes do início das aulas pelo monitor de fábrica. Mais detalhes sobre o processo de avaliação de disciplinas e fábricas podem ser encontrados em [CAVALCANTI, A. P. et al., 2008].

A etapa de vivência das fábricas dura seis meses e é concluída com uma apresentação dos projetos para uma banca, simulando uma situação de apresentação para investidores onde as equipes terão que apresentar os resultados de seus projetos sob quatro aspectos: vendas, tecnologia, inovação e capacidade de gerar resultados positivos. Concluída esta etapa, os estudantes passam para etapa de Pesquisa, para elaboração da sua dissertação.

4 Estudo de Caso: Resultados da Aplicação da Metodologia no MPES

O MPES foi implantando em 2007, iniciando sua primeira turma em agosto do mesmo ano. A primeira turma foi composta por 18 alunos com faixa etária média de 26,84 anos

e perfis profissionais focados nas áreas de engenharia de testes e engenharia de qualidade. Embora duas novas turmas estejam em andamento, no relato de experiência apresentado nesta seção, só serão consideradas as notas da primeira turma, visto que as fábricas de softwares montadas na segunda turma ainda não concluíram seu processo de avaliação e a terceira turma iniciou suas atividades no mestrado há pouco tempo.

Com o objetivo de validar a aplicação de PBL através da implantação de fábricas de software, algumas métricas foram definidas para verificar a efetividade da metodologia no contexto de educação de disciplinas relacionadas à Engenharia de Software. Além disso, verificou-se a necessidade de identificar as possíveis ações de melhoria no processo de educação definido, de forma que o mesmo possa ser evoluído de maneira incremental, aplicando os princípios de melhoria contínua. De uma forma mais abrangente, busca-se validar o como a aplicação de PBL está contribuindo para a formação dos alunos em uma pós-graduação.

Nesse contexto, foi necessário definir um método para validar os resultados obtidos, de forma que os valores alcançados possam ser analisados para geração de insumos e oportunidades de melhoria para o programa do MPES. Dessa forma, a avaliação da efetividade do processo e resultados obtidos pelos alunos são realizados com base nos resultados coletivos das fábricas, como mostra a Tabela 3, representando as notas aplicadas às fábricas de software no decorrer do semestre, com base no processo de avaliação descrito em [CAVALCANTI, A. P. et al., 2008].

A análise dos resultados coletados é feita considerando a escala descrita na Tabela 2, onde x representa o percentual de aumento da primeira à última nota.

Tabela 2: Escala de resultados.

VARIAÇÃO	RENDIMENTO
$x \geq 70\%$	Excelente
$40\% \leq x < 70\%$	Satisfatório
$x < 40\%$	Ruim

A Tabela 3 representa a evolução das notas coletivas, em uma linha de tempo cronológica, através de 3 formas de avaliação coletiva, realizada tomando como base 4 fábricas de software, onde foram possíveis executar:

- 3 status report, uma avaliação baseada no acompanhamento do processo das fábricas e cumprimento de marcos;
- 1 avaliação independente, focada em avaliação de artefatos produzidos;
- 1 avaliação de alinhamento, foco em desempenho e sintonia entre os membros de um time.

Tabela 3: Análise do rendimento com base em percentual.

FSW	Status Report	Avaliação Independente	Status Report	Avaliação Alinhamento	Status Report	Média	Desempenho 1 ^a .e última
1	8,58	8,37	9,33	9,28	9,17	8,95	7%
2	6,17	7,90	8,92	8,20	9,00	8,04	46%
3	5,00	6,90	7,00	7,90	8,67	7,09	73%
4	7,08	7,90	8,50	8,24	9,50	8,24	34%

Os dados obtidos a partir das avaliações deram origem ao gráfico da Figura 2, onde é possível observar que 3 das 4 fábricas obtiveram um constante crescimento ao longo de suas avaliações. A única fábrica que apresentou a nota do último status report inferior a um status report anterior teve um desempenho melhor do que as outras fábricas durante a maior parte do processo de avaliação. Isto mostra que as fábricas evoluem ao longo das avaliações realizadas sobre elas.

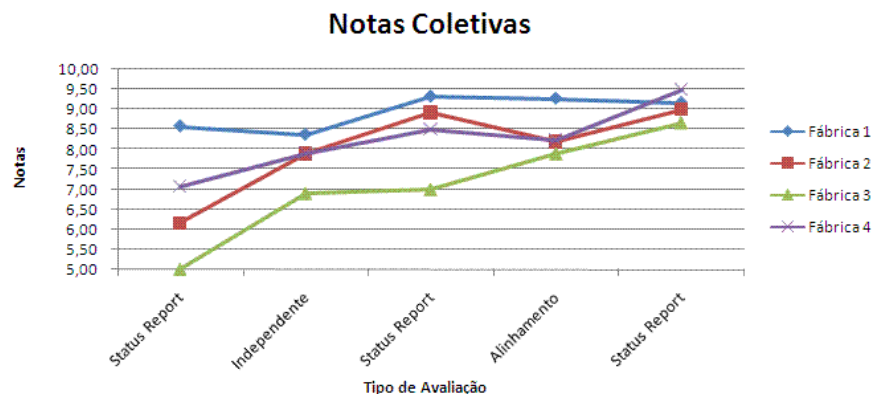


Figura 2: Gráfico de desempenho das fábricas de software da 1ª. turma do MPES.

Pode-se observar também que, no primeiro status report, havia uma diferença relevante entre os desempenhos das fábricas, chegando a 3,58 pontos. No último status report, a maior diferença entre os desempenhos das fábricas ficou em apenas 0,83 pontos (a maior nota foi 9,50 e a menor foi 8,63), o que mostra uma convergência das notas para um padrão de qualidade das fábricas observado ao final das disciplinas do mestrado. Outro dado importante foi o maior progresso observado em uma fábrica, que evoluiu em 3,67 pontos da primeira à última nota, representando um progresso de 73%, um desempenho considerado excelente de acordo com os critérios definidos na Tabela 2. Desta forma, os resultados da primeira turma mostraram que a metodologia está no caminho certo.

5 Conclusões e Considerações Finais

As demandas de mercado por profissionais com qualificação voltadas para a prática profissional e desenvolvimento de competências multidisciplinares têm estimulado o uso de metodologias de ensino diferenciadas. Embora, o método PBL tenha construído sua reputação na área médica, seus princípios mostram-se completamente alinhados às necessidades do ensino de Engenharia de Software, em particular, quando trata-se do desenvolvimento de software, uma vez que esta é uma atividade que exige trabalho em equipe, alta interatividade entre todos envolvidos e vivência prática de projetos em condições reais de mercado. Neste contexto, este artigo apresentou os resultados da metodologia adotada no MPES do C.E.S.A.R, tendo como base a criação de fábricas de software para ensinar a desenvolver software. Neste cenário, os estudantes aprendem fazendo, princípio fundamental da metodologia PBL. A metodologia é inovadora quando define a estratégia de criação e acompanhamento destas fábricas, estabelecendo processos, ferramentas e papéis específicos, permitindo que a prática aconteça de forma sistematizada e estruturada. Já na primeira turma, resultados positivos foram alcançados, como os apresentados na Seção 4.

Atualmente duas turmas estão em andamento. Algumas melhorias foram implantadas a partir da execução da metodologia na primeira turma, em particular quanto à preparação de professores, à participação mais efetiva do cliente nos projetos

e ao processo de acompanhamento das fábricas. Neste último, as etapas de avaliações foram ampliadas, novos status reports foram incluídos, tornando o processo mais controlado e interativo.

Finalmente, por não ter encontrado métricas publicadas sobre a aplicação de PBL no contexto deste artigo, não foi possível a comparação com propostas externas. Para se obter uma análise comparativa, os resultados de cada turma serão comparados entre si, a partir de métricas coletadas durante o processo. Com estas informações poderemos identificar pontos de melhoria na metodologia, que serão objetos de futuras publicações.

Referências

- CAVALCANTI, A. P.; Santos S. C.; M. C. Moraes; Albuquerque J. O.; and Meira S. R. L. An Evaluation Approach Based on the Problem-Based Learning in a Software Engineering Master Course. *Journal of Technology Management & Innovation, Special Issue on Entrepreneurship Education*, 2008.
- GÜZELIS, C. An Experience on Problem Based Learning in an Engineering Faculty. *Turk J Elec Engin, Vol.14, No., p. 67-76*, 2006.
- MORAES, A. K.O.; and MEIRA, S. R. L. and ALBUQUERQUE, J. O. Open Source Software Factory - Step by Step: A Case Report. In: *FIRST INTERNATIONAL CONFERENCE ON OPEN SOURCE COLLABORATIVE DEVELOPMENT PLATFORMS (Héphaïstos)*, 2006.
- NING, C. Undergraduate academic programme: planning, development, implementation and evaluation. *J. Engng. Educ., v.11, n.3, p.175-84*, 1995.
- PETERSON, M. Skills to Enhance Problem-based Learning. *Med Educ Online [serial online] 2,3*, 1997.
- RIBEIRO, L. R. C. and MIZUKAMI, M.G. An experiment with PBL in higher education as appraised by the teacher and students. *Interface - Comunic., Saúde, Educ., v.9, n.17, p.357-68*, 2005.
- SAVERY, JR and DUFFY, TM. Problem based learning: An instructional model and its constructivist framework. *Educ Technology,;35(5):31-7*, 1995.
- TYNÄLÄ, P. Towards expert knowledge? A comparison between a constructivist and a traditional learning environment in the university. *J. Educ. Res., v.31, p.357-442*, 1999.
- WATERS, R. and MCCRACKEN, M. Problem-Based Learning in Computer Science. In: *5TH ANNUAL CONFERENCE ON PROBLEM-BASED LEARNING*, 1996.