UMA ABORDAGEM PARA SENSIBILIZAÇÃO EM MODELOS DE PROCESSO COM FOCO NO PROCESSO UNIFICADO

Francisco S. Marcondes[1], João L. Bernardes[2], Elisangela Botelho[3], Marta Nagata[4], Jefferson Zanutto[5]

Abstract *— This paper tackles an educational approach used to improve the learning about software development process applied to undergraduate students of a information systems course. This approach depict a preliminary phase of the educational strategy that intends to increase student awareness of the importance of the development process, its phases, roles, artifacts, etc. During this phase the students are presented, using PBL and progressive formalization techniques, to some of the Booch's best practices and the MPS.BR (a CMMI like certification) expected results. It resulted on a effective learning that prepares the students to be easily absorbed by a full development process.*

Index Terms — education, software engineering, software process, unified process.

INTRODUÇÃO

Ao lecionar em cursos de graduação em diversas instituições, os autores perceberam que em geral há um baixo interesse e consequente baixo entendimento sobre modelos de processo de desenvolvimento de software pelos alunos. Acredita-se que um dos motivos disto seja a sua pouca experiência em projeto, bem como a pouca necessidade profissional enquanto na graduação.

Um indicativo deste fato, é que, durante as fases avançadas do curso, alguns alunos já apresentam interesse por estes temas, embora para a maioria tais conceitos sejam muito abstratos levando ao baixo grau de interesse. Este fato leva a algumas distorções conceituais que prejudicam a formação do aluno fazendo com que este não substitua o conhecimento principiante com relação a este tema levando-o a organizar as atividades que levam à construção do software de maneira pouco efetiva e pouco escalável.

Vem se observando durante as aulas que as principais distorções atuais são a de que o *RUP* (*Rational Unified Process*) não passa duma sucessão de modelos do tipo *step-wise* e a de que o *XP (eXtreme Programming)* é melhor do que o *RUP* pelo motivo deste não utilizar documentação e que por este mesmo motivo o *RUP* é burocrático, sem qualquer outro tipo de análise.

Buscando uma solução para a origem destas distorções, diversas tentativas pedagógicas foram apresentadas pela comunidade científica ao longo dos anos [4, 1, 2, 11], no entanto acredita-se que nenhuma dessas iniciativas tiveram grande sucesso, pois ainda nenhuma foi amplamente divulgada nem mostrou resultados práticos que atestassem seu efetivo sucesso.

As principais abordagens encontradas visam simular o ambiente de desenvolvimento por meio de experiências e jogos aumentando o teor prático da disciplina na efetiva aplicação do conceito [4, 1, 2, 11]. Ou seja, trata-se de uma realidade simulada de software *house* ou fábrica de software conforme o nível de processo desejado pelo professor.

Este estudo faz parte de uma pesquisa maior que segue por esta mesma linha, buscando métodos, técnicas e ferramentas que apóiem uma estratégia de ensino de realidade simulada. Este trabalho apresenta o resultado de uma fase de sensibilização anterior à efetiva aplicação do processo, ela visa a introduzir paulatinamente os conceitos de forma a sensibilizar os alunos para que a incorporação do processo seja mais suave.

A dificuldade de implementação de um processo já é conhecida, pois implica na mudança da cultura organizacional, orientando a definição e melhora dos processo de desenvolvimento de software [10]. Logo, o impacto em estudantes acostumados a desenvolver software *ad hoc* tende a ser tão grande ou até maior do que nas empresas de desenvolvimento de software, mostrando a necessidade de uma sensibilização anterior.

Utilizando as estratégias de ensino por projetos formatada para incorporar a união das técnicas de PBL (Problem-Based Learning) e formalização progressiva para apresentar alguns conceitos de Boas Práticas de Booch e do nível G do MPS.BR (Melhoria de Processo de Software Brasileiro), cria-se um ambiente apropriado para despertar e instigar o interesse e entendimento de alunos de graduação, e sensibilizá-los para a necessidade e importância dos processos de desenvolvimento de software (dimensão de processo da Engenharia de Software).

[1] Francisco Supino Marcondes, docente do Centro Universitário Senac, Av. Engenheiro Eusébio Steuvax, 823 – Santo Amaro, São Paulo, SP, Brasil francisco.smarcondes@sp.senac.br
[2] João Luiz Bernardes, docente do Centro Universitário Senac, Av. Engenheiro Eusébio Steuvax, 823 – Santo Amaro, São Paulo, SP, Brasil joao.lbjunior@sp.senac.br
[3] Elisangela Botelho, docente do Centro Universitário Senac, Av. Engenheiro Eusébio Steuvax, 823 – Santo Amaro, São Paulo, SP, Brasil elisangela.botelho@sp.senac.br
[4] Marta Nagata, aluna do Centro Universitário Senac, Av. Engenheiro Eusébio Steuvax, 823 – Santo Amaro, São Paulo, SP, Brasil, mayallan@ig.com.br
[5] Jefferson Zanutto, coordenador de SI do Centro Universitário Senac, Av. Engenheiro Eusébio Steuvax, 823 – Santo Amaro, São Paulo, SP, Brasil elisangela.botelho@sp.senac.br

A hipótese acima foi aplicada como experimento a cerca de 30 alunos de 4º semestre de um curso de bacharelado em Sistemas de Informação, durante uma disciplina de projeto interdisciplinar com duração de 18 semanas.

DESCRIÇÃO DO EXPERIMENTO

A técnica de ensino por projetos, em geral, é a base de disciplinas de projetos interdisciplinares com o principal objetivo de ajudar o aluno a relacionar a teoria de várias disciplinas com a prática [9]. A técnica de PBL vem sendo utilizada com bastante sucesso em cursos de computação exercitando características importantes como solução de problemas e iniciativa nos alunos [5]. Num experimento anterior aplicado a grupo semelhante de alunos não contou com uma fase de sensibilização, envolvendo apenas a união destas duas técnicas pedagógicas.

Com a mudança de intenção que originou a realização da fase de sensibilização, foi incorporado o conceito de formalização progressiva onde cada conceito é incorporado conforme o tempo e necessidade conduzindo o estudante na transição entre o conhecimento principiante, refutando um a um cada estágio, até se chegar no conhecimento especializado que se deseja incorporar. Percebeu-se que esta técnica funciona muito bem quando aliada à PBL pois permite que a solução do problema seja apresentada gradualmente, permitindo um maior tempo de reflexão.

Uma vez que a técnica pedagógica havia sido definida, foi necessário estabelecer os objetivos de aprendizagem que deveriam ser atingidos. Optou-se por, neste experimento, incorporar o máximo possível das boas práticas de Booch [4] e do nível G do MPS.BR [10] utilizando para isso um subconjunto de papéis, atividades e artefatos do OpenUP [3].

Das boas práticas de Booch, o projeto contemplou: desenvolver software interativamente; usar arquitetura baseada em componentes; modelar o software visualmente; e verificar continuamente a qualidade do software (a partir da segunda de quatro interações).

A prática desenvolver software interativamente foi aplicada de forma que os alunos tinham que apresentar a cada 4 semanas uma evolução de seu projeto, ou seja, houveram 4 interações em 18 semanas.

A prática de usar arquitetura baseada em componentes, foi implementada conforme apresentado na tabela 1. Individualmente, os alunos eram chamados a desenvolver uma USC (Unidade de Software de Computador), a qual deveria ser integrada num CSC (Componente de Software de Computador), formando um grupo de alunos responsáveis pelas USCs e por fim novamente integrados num ICSC (Item de Configuração de Software de Computador). Cada item – USC, CSC e ICSC – deveria possuir os atributos do conceito componente de software (encapsulamento, interfaces, etc) para possibilitar a integração.

A prática de modelar o software visualmente foi implementada solicitando que os alunos entregassem modelos do sistema em UML. Tais diagramas deram origem aos artefatos de modelo do sistema gerado.

A verificação constante de qualidade foi implementada pela realização de testes para verificar continuamente se o sistema atingia aos requisitos. Os mesmos testes deveriam ser realizados nos 3 níveis de integração propostos.

O projeto tratava-se de um SSC para processamento de multi-documentos bancários e deveria envolver além da disciplina de projeto, a disciplina de Banco de Dados e de interface Humano-Computador.

TABELA 1
Divisão de componentes por aluno

USC	CSC	ICSC	SSC
Captura de Documentos	Digitalização de Documentos	Captura e Reconhecimento	Sistema de Processamento de Multi-Documentos
Captura de Documentos após gravação de Malote			
Criação de Malotes			
Gravação de Malotes			
Reconhecimento por OCR	Reconhecimento de Documentos		
Reconhecimento por ICR			
Digitação de Documentos	Digitação	Digitação	
Redigitar Documento gravado			
Gravar Documento digitado	Manutenção de Digitação		
Rejeição de Documentos			
Alterar valor de Documentos	Configuração de Transação	Batimento	
Remover Documento de uma Transação			
Batimento automático de Transação			
Incluir Envelope de Depósito em uma Transação			
Incluir Documento em uma Transação			
Gravar Transação	Gerenciamento de Transação		
Editar Transação gravada			
Criar uma nova Transação			
Visualizar quantidade de Documentos capturados/minuto	Supervisão de Captura	Supervisão	
Imprimir relatório de Documentos capturados/minuto			
Visualizar quantidade de Documentos digitados/minuto	Supervisão de Digitação		
Visualizar quantidade de Documentos digitados com erro			
Imprimir relatório de Documentos capturados/minuto			
Visualizar quantidade de Transações batidas/minuto	Supervisão de Batimento		
Visualizar quantidade de Transações batidas com erro			
Imprimir relatório de Transações batidas/minuto			
Visualizar relatório de Entrada e Saída de Documentos no Sistema	Supervisão Geral		
Imprimir relatório de Entrada e Saída de Documentos no Sistema			

Das 6 práticas, apenas 2 não puderam ser implementadas na fase de sensibilização: gerenciar

requisitos e controlar mudanças no software, pois envolvem aspectos dinâmicos que ocorrem num projeto como requisitos mutantes e mudanças causadas por diversos elementos da equipe.

Uma outra preocupação no projeto foi com relação ao MPS.BR, onde buscou-se conformidade com o nível G – Parcialmente Gerenciado. A tabela 2 apresenta uma análise de conformidade do projeto apresentado neste estudo. Pode-se dizer que foi obtida uma conformidade de 70% com o processo de Gerência de Projetos e 40% com o processo de Gerência de Requisitos (análise adaptada ao contexto acadêmico) [8]. Percebe-se a dificuldade de implementar de uma forma direta os aspectos relativos a requisitos em ambiente acadêmico, conforme apresentado em [7, 8].

TABELA 2
Análise de Adequação do Projeto ao Nível G do MPS.BR

Gerência de Projetos	Gerência de Requisitos
GPR1 – Totalmente atendido GPR2 – Não Atendido GPR3 – Totalmente atendido GPR4 – Não Atendido GPR5 – Totalmente atendido[6] GPR6 – Totalmente atendido GPR7 – n/a[7] GPR8 – Totalmente atendido GPR9 – Não Atendido GPR10 – Parcialmente Atendido GPR11 – Não Atendido GPR12 – Totalmente atendido GPR13 – Largamente Atendido GPR14 – Parcialmente Atendido GPR15 – Totalmente atendido GPR16 – Não Atendido GPR17 – Não Atendido	GRE1 – Totalmente atendido GRE2 – Não Atendido GRE3 – Não Atendido GRE4 – Largamente Atendido GRE5 – Não Atendido

1ª Iteração

Na 1ª iteração, os alunos deveriam desenvolver uma USC de forma individual e não foi definido como eles deveriam realizar a tarefa. Muitos utilizaram a técnica de iniciar o desenvolvimento pela programação e eventualmente eram apresentados alguns poucos modelos.

Foi atribuído a cada aluno uma USC e as informações que cada um possuía eram baseadas no plano de aquisição do MPS.BR apresentado em aula. A primeira dificuldade que pôde ser observada foi a de cada aluno separar a funcionalidade do componente que lhe foi atribuído, como um componente único e com funcionalidades próprias, sem alinhá-lo com outros componentes. Cada aluno deveria apresentar um protótipo e um relatório denotando a experiência, as dificuldades e as pesquisas envolvidas para a conclusão de seu protótipo. Era necessário identificar os requisitos do componente e utilizar conceitos básicos de outras matérias que compunham a base deste projeto.

Esta fase tem como objetivo fazer com que os alunos tenham as dificuldades inerentes à falta de planejamento e de um projeto do software, criando o espaço necessário para refutar este conhecimento na segunda iteração. Eles deveriam efetuar o desenvolvimento de seu componente auxiliado pelo tutor da disciplina de projeto, fazer com o que o aluno trouxesse não apenas dúvidas, mas soluções baseadas em idéias, de abstração do problema e em pesquisas. As dúvidas iniciais envolveram cada aluno com seu componente resultando ainda como uma abstração do problema, sem concretização de aplicabilidade e funcionalidade.

Em geral, os alunos apresentaram algum resultado devido ao *deadline* da atividade, mas não estavam seguros quanto ao resultado pois não tinham com o que comparar. Devido a natureza do problema apresentado ser um software de médio porte, o qual deveria passar por integrações sucessivas e por ter sido apresentada apenas a necessidade do usuário e não os requisitos, surgiram naturalmente uma série de problemas e dificuldades.

Estes problemas abriram espaço para a discussão sobre o motivo da falha de iteração inicial e os alunos foram induzidos a entender que eles ainda não haviam obtido um perfeito conhecimento do que fazer, nem de como fazer.

2ª Iteração

Foi apresentado, na primeira semana da 2ª iteração, um artefato de escopo e um artefato de caso de uso, ambos modelos do OpenUP. Na segunda semana, foi apresentado um artefato de plano de iteração e de avaliação de *status*, propiciando um modelo de como deveria ser realizado o planejamento

Na terceira semana, foram apresentados aos alunos os artefatos de arquitetura e de casos de teste mostrando como organizar o sistema. Tal procedimento resultou numa entrega bastante apropriada na quarta semana da 2ª iteração da USC de cada aluno.

O protótipo anterior deveria ser aprimorado, porém baseado em alguns documentos vistos na disciplina, como padrões de projeto. O funcionamento de cada componente ainda era abstrato frente à complexidade do sistema como um todo.

Mas a documentação solicitada como parte integrante desta fase teve como por objetivo formalizar alguns conceitos que foram aplicados na 1ª iteração, utilizado como um guia para o aprimoramento do protótipo. Inicia-se nesta fase uma maior contextualização dos padrões de desenvolvimento, ainda vistos como um pequeno componente, o desenvolvimento deste protótipo deve começar a seguir estes padrões.

O aluno, em quatro semanas, deveria entregar o protótipo II, relatório individual – contendo as dificuldades, pesquisas, conceitos aplicados, auto-avaliação, conclusão e identificação de melhorias de desempenho pessoal –

[6]Por se tratar de um projeto acadêmico não existe espaço para orçamento.
[7]Planejamento de Recursos Humanos

preenchimento dos artefatos: Caso de Uso, Plano de Iteração, Documento de Arquitetura e Documento de Teste.

3ª Iteração

Na 3ª iteração, foi solicitado que os alunos integrassem as suas USC em CSC formado por alunos do mesmo sub-grupo. Houve também a solicitação de que integrassem os artefatos para formar os documentos do CSC. Nesta etapa, foi necessária uma maior participação do aluno no desenvolvimento do protótipo, utilizando os artefatos da fase anterior – Caso de Uso, Plano de Iteração, Documento de Arquitetura e Documento de Teste. O aluno deve identificar requisitos do sistema e iniciar a análise de arquitetura do protótipo.

Ainda nesta etapa, o aluno possuía uma breve experiência com os artefatos da fase anterior, e isto facilitou ao aluno a concepção e utilização dos padrões de desenvolvimento do software. O grupo pertinente a cada integração teve um novo desafio, utilizar seu componente individual para agregar funções a outro componente.

Para este fim, os documentos de apoio foram utilizados como um guia para este desenvolvimento. Utilizou-os para complementar a formalização de conceitos que já haviam sido aplicados, e baseado na técnica PBL, o aluno pôde discutir em seu grupo qual o comportamento de cada componente para que a aplicação final possuísse todas as funcionalidades deste grupo; pôde, também, trazer ao grupo as mesmas dificuldades obtidas no início, no que se refere ao desenvolvimento e seus objetivos de funcionalidade. Com base nestas discussões e informações que cada aluno traz ao grupo, os problemas puderam ser observados e extraídos diretamente da prática vivenciada em cada componente e na própria integração entre componentes, com base na necessidade de incorporação de conceitos, noções e princípios.

Logo na primeira semana da 3ª iteração, foram entregues os artefatos de plano de projeto, requisitos suplementares, modelos e teste do desenvolvedor e log de teste. Na segunda e terceira semana, houve apenas o acompanhamento da realização das atividades e as aulas foram usadas para discussões pontuais e para tirar dúvidas.

Esta fase, gerou para os alunos uma série de problemas que tiveram que ser tratados, um deles foi repensar o software de forma que ele pudesse ser integrado com os outros componentes. Outro foi como montar o componente da melhor forma para que ele fosse passível de integração para formar o ICSC (Item de Configuração de Software de Computador). Os alunos tiveram ainda que exercitar a negociação e a divisão de tarefas para poder integrar tanto o código como os artefatos.

O problema de entender o software como um todo para repensar como cada componente seria integrado teve sua solução facilitada por uma tarefa requisitada no início dessa iteração pela disciplina de Usabilidade e Interface Humano-Computador: a criação de um protótipo simples (em papel) da interface do sistema completo e a realização de testes sobre esse protótipo. A resposta dos alunos foi bastante positiva e os testes de fato resultaram em mudanças na interface, mostrando que o processo de ensino descrito aqui é útil para sensibilizar os alunos não só da importância do processo de desenvolvimento utilizado, mas também de outras metodologias, como o desenvolvimento da interface baseado em protótipos e centrado no usuário.

O prazo para conclusão desta fase foi de 4 semanas, e além de relatório individual, a cada grupo foi solicitado o preenchimento dos artefatos: Caso de Uso, Plano de Iteração, Documento de Arquitetura, Documento de Teste, Plano de Projeto, Lista de Risco, Documento de Visão, Requisitos Suplementares e Caso de Teste.

4ª Iteração

Na 4ª iteração, o objetivo era fazer a integração em nível de ICSC, de forma que cada grupo agora possuía cerca de 7 a 9 pessoas. Foi formalmente solicitado a divisão de papéis onde cada aluno, ao assumir o papel, deveria ficar responsável pelas atividades e pelos artefatos correspondentes à sua função. Nesta etapa foi passado um *script* de alto nível de como as atividades deveriam ser conduzidas, pois percebeu-se que mesmo apresentando o motivo dos artefatos, os alunos não foram capazes de deduzir a melhor ordem de utilização. Por outro lado, percebeu-se que, por meio de seus relatórios, eles haviam identificado apropriadamente o uso dos artefatos.

Quando o *script,* com a ordem em que as atividades deveriam ser desempenhadas, foi entregue e realizada uma discussão sobre ele, aparentemente os conceitos foram sedimentados e os alunos compreenderam a idéia do processo, a função dos artefatos, atividades e papéis. Vale ressaltar que mesmo sem uma definição formal, os alunos já haviam realizado a divisão por papéis.

Nesta última etapa, ocorreu a integração dos componentes de um mesmo grupo, já integrados na etapa 3. Os alunos deveriam preencher todos os artefatos solicitados em etapas anteriores, apenas acrescentando o PSP0 *Process Script*, e efetuar uma apresentação de como foi a integração dos componentes e como todo o sistema funciona, envolvendo sua comunicação com Banco de Dados e aplicação de conceitos de Interface Humano-Computador.

Na apresentação de cada grupo, foi necessário levantar junto a cada integrante do grupo as maiores dificuldades, as dúvidas encontradas e as soluções implementadas. Nesta etapa, obtém-se maior resultado quanto à formalização progressiva, fazendo com que os grupos discutam entre si como cada artefato o auxiliou na concepção do protótipo final, a utilização dos conceitos e a aplicação de recursos utilizados com base em pesquisas. Cada grupo trouxe idéias e soluções diferentes para resolver a funcionalidade do protótipo, e, ao mesmo tempo, obteve informações de como cada protótipo funciona e qual a responsabilidade no contexto geral. Esta fase foi a de maior importância para

toda a turma de BSI (Bacharelado em Sistemas de Informação), pois proporcionou não apenas a formalização dos conceitos, mas os objetivos de funcionalidade já possuíam maior definição.

Comentários Finais

Para ilustrar o resultado segue o comentário de um aluno extraído de seu relatório: *Nos deparamos com uma estrutura de trabalho diferente de todas que nos já desenvolvemos anteriormente(...). Achei bem interessante esse modelo de trabalho, cada aluno responsável por uma USC e documentando os passos que vão sendo alcançados. Esse projeto irá desenvolver tanto a nossa capacidade de desenvolvimento de projeto com a nossa capacidade de trabalho em equipe, sendo essas as grandes características procuradas por empresas de nome no mercado. Pude observar que conforme vamos desenvolvendo o projeto as nossas idéias vão ficando mais concretas e mais fáceis de realizar conforme vamos escrevendo a documentação desenvolvendo os pontos estabelecidos.*

Pode-se notar claramente que os itens que tiveram por objetivo serem trabalhados foram bastante significativos, mostrando que o projeto atingiu o seu resultado esperado. Pode-se dizer que o aspecto de maior dificuldade foi o de formular corretamente o problema e elicitar os requisitos, o que já era esperado, pois trata-se de uma dificuldade já identificada [6, 3], para o início do desenvolvimento do sistema. A vivência da utilização dos artefatos, após a identificação da sua necessidade, concretiza a abstração de um documento. A dúvida, os resultados das pesquisas e a descoberta de soluções para resolver o problema, mostram ao aluno as reais necessidades de aprendizado, envolvendo o aluno e lhe permitindo a descoberta.

Uma vez que todos os artefatos foram disponibilizados, os papéis e as atividades e a divisão por iterações seguiram o padrão do OpenUp. Espera-se que os alunos estejam preparados para serem apresentados ao processo e que esta transição seja muito pouco traumática. Acredita-se que isso será possível, pois na 4a fase do projeto, os alunos já dispunham de uma gama grande de itens do processo, precisando apenas haver uma pequena adaptação.

Conclusão

O aluno inicialmente recebeu informações básicas para concepção do protótipo, isto gerou um pouco de frustração individual e um elevado grau de dificuldade para iniciar a primeira fase, porém analisando como as informações chegaram ao aluno ou o envolvimento pessoal do aluno para sanar parte das dúvidas, verifica-se que os alunos não perceberam no início do projeto as técnicas utilizadas para o estudo da engenharia de software.

Por se tratar de um estudo complexo do ponto de vista do aluno, a aplicação das técnicas de aprendizado trouxe a eles as informações como uma solução, como uma concretização de idéias iniciais, sem conexão com o final do projeto. A parte mais difícil foi formular o problema e reconhecer requisitos do sistema para iniciar o protótipo, porém o objetivo seria o de concretização dos conceitos e não o desenvolvimento do protótipo em si.

A vivência da utilização dos artefatos, após a identificação da necessidade do artefato, concretiza a abstração de um documento. A dúvida, os resultados das pesquisas e a descoberta de soluções para resolver o problema, mostram ao aluno as reais necessidades de aprendizado, envolvendo o aluno e lhe permitindo a descoberta.

Referências Bibliográficas

[1] Baker, A., Navarro, E. O. Hoek, A. V. D. "An Experimental Card Game for Teaching Software Engineering Process", *Journal of Systems and Software,* vol 75, 2005, pp. 3.

[2] Beniti, F. B. V., Molléri, J. F., "Utilização de um RPG no Ensino de Gerenciamento e Processo de Desenvolvimento de Software", *XXVIII Congresso da SBC*, 2008, pp 258.

[3] Eclipse, "OpenUP", IBM, 2006.

[4] Kruchten P., "Introdução ao RUP", *Ciência Moderna Ed.,* 2004.

[5] Marcondes, F. S., Cunha, A. M., Dias, L. A. V. "Uma Metodologia de Ensino das Técnicas de Qualidade, Confiabilidade, e Segurança (Safety) de Software Aplicadas num Protótipo de Sistema de Software de Computador Embarcado de Tempo Real no Instituto Tecnológico de Aeronáutica – ITA", *VI Simpósio Brasileiro de Qualidade de Software - SBQS*, 2007, pp. 405.

[6] Marcondes, F. S.*, Sonoda, E. H., Brumatto, H. J., Barbosa Jr, L. C., Zanutto, J. "Problemas no Ensino de Engenharia de Software: Elicitação de Requisitos", VI International Engineering and Computer Education – ICECE, 2009, pp. 64*.

[7] Marcondes, F. S., Sonoda, E. H., Brumatto, H. J., Barbosa Jr, L. C., Zanutto, J. "Uso do OpenUP no Ensino do Processo Iterativo e Incremental de Desenvolvimento de Software", *IV Workshop de Pós-Graduação e Pesquisa do Centro Paula Souza*, 2009.

[8] Marcondes, F. S., Sonoda, E. H., Brumatto, H. J., Barbosa Jr, L. C., Zanutto, J. *"Problems on Software Engineering Learning: Domain Engineering"*, VI International Conference on Information Technology, 2009, pp 1636

[9] Masetto, M. T. "Competência Padegógica do Professor Universitário", *Summus Ed.*, 2003

[10] MPS.BR, "Guia de Implementação – Parte 1: Nível G", *Softex*, vol. 1 , 2009

[11] Ohlsson, L. e Johansson, C. "A Practice Driven Approach to Software Engineering Education." *IEEE Transactions on Education*, vol. 38, no. 3, 1995