

ESCollab: Uma metodologia colaborativa voltada para o ensino de Engenharia de Software

Severino Paiva¹, Álvaro Medeiros²

¹*Coordenação de Informática, Faculdades ASPER, Brasil*
CEP 58033-450 João Pessoa/PB, Brasil

²*PPGI/UFPB, Universidade Federal da Paraíba, Brasil*
CEP 58059-900 João Pessoa/PB, Brasil

{paiva@asper.edu.br, alvaro@di.ufpb.br}

Abstract. *Deficiencies in training of software engineers and the advances of resources for distance learning, especially with Web 2.0, are motivators for the development of tools focused on improving education in the area of Software Engineering. In this paper, we present the ESCollab, a methodology focused on teaching software engineering using web 2.0 resources and philosophy of collaborative work in order to make teaching more attractive and more effective learning.*

Resumo. As deficiências na formação dos engenheiros de software e os avanços dos recursos para educação à distância, especialmente com a Web 2.0, são motivadores para o desenvolvimento de ferramentas voltadas para a melhoria da educação em Engenharia de Software. Neste artigo, apresentamos a ESCollab, uma metodologia de ensino para Engenharia de Software usando recursos da Web 2.0 e da filosofia de trabalho colaborativo no sentido de tornar o ensino mais atraente e a aprendizagem mais efetiva.

1. Introdução

As aplicações de informática fazem parte de todas as atividades humanas na atualidade, criando uma verdadeira dependência do uso de softwares na sociedade contemporânea. Essa importância do software no mundo atual, fez surgir uma preocupação mundial com a qualidade do processo de desenvolvimento de software, fazendo com que mais atenção seja dada à forma como os discentes aprendem os conceitos de programação ao longo de um processo de ensino-aprendizagem que é objeto de estudo da Engenharia de Software - ES.

Essa relevância também é traduzida em cifras. No âmbito nacional, a indústria do *software* tem evoluído de forma substancial nas últimas décadas, mas ainda se tem uma balança comercial deficitária nesse segmento, atingindo aproximadamente US\$ 5 bilhões de dólares de déficit anuais. A receita de exportação do Brasil (US\$ 1,0 bilhão) pode ser considerada baixa em comparação com a da Índia (maior exportador mundial, com US\$ 50 bilhões) (IBGE, 2010).

Outrossim, a literatura técnica mundial registra várias discussões sobre problemas no processo ensino aprendizagem na área de Engenharia de Software, podendo claramente ser observada a necessidade de uma maior integração e cooperação, no ensino-aprendizagem de disciplinas da área de Engenharia de Software, conforme Paiva (2011a).

Propor novas abordagens metodológicas e desenvolver ferramentas tecnológicas que possam propiciar a evolução da qualidade da formação dos desenvolvedores de *software* e, conseqüentemente, a competitividade do *software* nacional, poderá ser uma

valiosa contribuição para o desenvolvimento econômico, educacional e social do nosso país.

Com esse intuito, nas próximas seções, apresenta-se a caracterização do ensino em ES, os seus problemas e a Metodologia ESCollab de ensino-aprendizagem com uma abordagem colaborativa, visando aperfeiçoar o ensino das disciplinas ligadas à grande área da Engenharia de Software.

2. Caracterização do Ensino das Disciplinas da área de Engenharia de Software

As Instituições de Ensino Superior (IES) contam com um corpo docente onde professores $P = \{ p_1, p_2, \dots, p_n \}$ são responsáveis por ministrar disciplinas $D = \{ d_1, d_2, \dots, d_n \}$ que fazem parte de uma Grade Curricular $G = \{ \langle e_1, c_1 \rangle, \langle e_2, c_2 \rangle, \dots, \langle e_n, c_n \rangle \}$ tal que cada disciplina de D tem uma relação $\langle e_i, c_i \rangle$ entre uma ementa e_i e o respectivo conteúdo c_i que deve ser ministrado. A forma de se aferir se o conteúdo c_i foi realmente ministrado por p_i é realizado por uma avaliação $A = \{ \langle t_1, t_2, \dots, t_n \rangle, \langle s \rangle \}$ que é composta por diversas ferramentas avaliativas, destacando-se dentre elas, em alguns casos, um projeto obrigatório, onde um software S é elaborado.

Para caracterizar o estado da arte do ensino de Engenharia de *Software*, foram formuladas três etapas de levantamentos: pesquisa de opinião nacional sobre a avaliação dos alunos e ex-alunos da área de computação sobre o ensino de ES; revisão sistemática das pesquisas realizadas sobre a melhoria no ensino de Engenharia de Software; pesquisa sobre as iniciativas das entidades formuladoras de parâmetros e padrões para o ensino na área de computação.

2.1 Pesquisa de Opinião sobre o ensino de Engenharia de Software

Foi realizada uma pesquisa de opinião sobre a avaliação dos alunos e egressos das disciplinas da área de Engenharia de Software através de entrevistas realizadas com visitas in loco às instituições de ensino superior na cidade de João Pessoa, como também via formulários disponibilizados através da Internet, e entrevistas feitas no Congresso da Sociedade Brasileira de Computação 2011, em Natal-RN, conforme mostra Paiva (2011b). As respostas aos questionários atingiram 142 alunos e ex-alunos de mais de 30 instituições do Brasil, totalizando 1,34% do total das IES brasileiras. Dentre os formulários respondidos, foram 30 questionários de alunos ou ex-alunos de instituições da Paraíba e 112 de outros estados. Da pesquisa, pode-se apresentar algumas considerações:

- Os respondentes são oriundos de 30 instituições de ensino superior, representando as cinco regiões do país, predominando a UFPI (12%), ASPER (9%), UFCG (6%), UFF (6%), UFPA (6%), IFRN (6%), UFPB (4%) e UNIPÊ (4%).
- Os cursos predominantes entre os participantes da pesquisa são Ciência da Computação (50%), Sistemas de Informação (24%) e cursos tecnológicos (15%).
- Cerca de 85% dos respondentes não possuíam nenhuma experiência em desenvolvimento de software antes de cursar o seu curso superior na área de computação.
- As disciplinas da área de Engenharia de Software cursadas pelos respondentes têm carga horária de até 60 horas (49%) e de até 75 horas (23%), sendo afirmado que na grade dos seus cursos têm, no máximo, 2 disciplinas (35%) ou 1 disciplina (26%). A maioria dos respondentes considera a carga horária das disciplinas insuficiente (52%) para um bom aprendizado.
- As principais ferramentas didáticas utilizadas pelos docentes para ministrarem os seus cursos são aulas expositivas (82%), aulas práticas de laboratório (61%), elaboração de projetos (56%), apresentação de seminários (45%), estudo de casos

(40%) e ensino baseado em problemas (27%). Com relação aos projetos desenvolvidos durante as disciplinas, os respondentes afirmam que em apenas 53% das disciplinas cursadas houve o desenvolvimento de projetos; os respondentes avaliam ainda que, em sua grande maioria (55%), não houve a implementação de projetos ou os projetos não tiveram nenhuma aplicação prática, sendo utilizados apenas para atribuir notas aos discentes nas disciplinas.

- Os principais mecanismos de avaliação utilizados nas disciplinas cursadas foram as provas subjetivas (72%), implementação de projetos (59%), seminários (51%) e provas objetivas (44%).
- Com relação ao conhecimento prévio do programa e do planejamento pedagógico das disciplinas no início das disciplinas, os respondentes afirmam que tomaram conhecimento (70%) e que os mesmos teriam sido entregues pelos docentes em sala (45%).
- Após cursar as disciplinas da área de ES, os respondentes julgam que suas habilidades na área de desenvolvimento de software são apenas regulares (30%), pouca (16%) ou nenhuma (4%), avaliando que os seus conhecimentos na área estão na média das suas turmas (61%), embora a maioria (59%) considere-se apta a entrar no mercado de trabalho.
- Ao avaliarem a qualidade do atendimento docente extra-sala de aula, os respondentes julgaram ser ótimo (25%) ou bom (35%), afirmando que foram utilizados como mecanismos de comunicação adicional as listas de discussão (45%), ferramentas do Google Docs (37%) e MSN (11%).
- Perguntados sobre a ocorrência de uma efetiva cooperação entre os docentes da área na elaboração de conteúdos, na continuidade dos assuntos e nos projetos das disciplinas, os respondentes avaliam que perceberam um trabalho em equipe (44%), que cada docente fazia o seu trabalho independente (27%), que todas as disciplinas foram ensinadas por apenas um docente (21%) e que não houve nenhuma cooperação, havendo até críticas ao trabalho dos outros docentes (7%).

2.2 Revisão Sistemática das Pesquisas Realizadas sobre a melhoria no ensino de Engenharia de Software

Segundo Bordenave(2007), a pesquisa bibliográfica é um procedimento formal para a aquisição de conhecimento sobre a realidade estudada. Nesse sentido, investigouse na literatura técnica da área, o material relevante sobre iniciativas que visassem melhorar o processo de ensino-aprendizagem na área de Engenharia de Software. Para qualificar os resultados obtidos, gerando mais valor científico, optou-se por realizar uma revisão sistemática nas seguintes bases de dados:

- 1) IEEE (<http://ieeexplore.org>);
- 2) ACM (<http://portal.acm.org/dl.cfm>);
- 3) Science Direct (<http://www.sciencedirect.com>);
- 4) SBC (anais).

Os trabalhos deveriam estar disponíveis na Web sem custo, tendo sido publicados nos últimos três anos, abordando iniciativas voltadas para o ensino de Engenharia de Software. Um extrato de alguns desses trabalhos, suas propostas para a melhoria do ensino de ES e uma avaliação das mesmas podem ser vistos no Quadro 1:

Quadro 1 – Algumas Pesquisas voltadas para a melhoria do ensino de ES

AUTORES	ABORDAGEM UTILIZADA	AVALIAÇÃO
(PRIKLADNICKI, Rafael et al.,	Metodologia baseada em dinâmicas de grupo, EAD,	Utiliza algumas ferramentas didáticas interessantes, mas

2009)	experimentos, projetos, atividades lúdicas, jogos	ignora outras possibilidades como colaboração, produção de artigos científicos e seminários, dando pouca ênfase ao planejamento prévio da disciplina.
(Yadav, S. S. & Xiahou, J.,2010)	Metodologia orientada por projetos	Concentra-se unicamente no uso de projetos integrados ao processo educacional.
(Qiu, M. & Chen, L.,2010)	Metodologia baseada em problemas e EAD	Enfatiza PBL e uso de recursos EAD, deixando de lado outras ferramentas didáticas.
(Mottok, J. & Gardeia, A.,2011)	Cooperação entre o ensino básico e o universitário usando projetos de robôs LEGO	Baseia-se na integração com o ensino médio e no desenvolvimento de projetos na área de robótica. Boa estratégia, mas de difícil implementação.
(Mirian-Hosseiniabadi, S.-H. et al, 2010)	Metodologia baseada em experimentos	A proposta foca-se nos experimentos, deixando de lado outras estratégias possíveis de serem utilizadas, como por exemplo, os estudos de caso e projetos.
(Jianguo, L.,2009)	Interação entre pesquisa e ensino usando projetos	Baseia-se na idéia de que a aprendizagem pode ser potencializada na integração com projetos de pesquisa.
(Gotel, O., 2009)	Colaboração à distância à base de projetos	Enfatiza apenas duas ferramentas básicas: pedagogia de projetos e recursos EAD.
(Giraldo, F. D.,2009)	Colaboração multi-institucional e EAD	Utiliza a abordagem tradicional, focando na colaboração e no uso de recursos EAD.
(Garousi, V.,2010)	Pedagogia de projetos e revisão em pares	A proposta foca apenas na pedagogia de projetos e revisão em pares.
(Garg, K. & Varma, V.,2009)	Avaliação baseada em estudo de casos	Enfatiza a atividade de ensino-aprendizagem no estudo de caso como único elemento central.
(Coccoli, M. et al,2011)	Pedagogia de projetos e colaboração com comunidades de desenvolvedores	Foca apenas em projetos e colaboração com comunidades de desenvolvedores.
(Anisetty, Priatham; Young, Paul, 2011)	Metodologia baseada em problemas, colaboração e projetos em grupo	Restringe-se apenas a três ferramentas didáticas: PBL, colaboração e projetos.

De uma maneira geral, conforme pode ser visto em Paiva(2011a), observa-se uma série de problemas recorrentes no processo ensino-aprendizagem das disciplinas ligadas à grande área de Engenharia de *Software*:

Os principais problemas apresentados por estes pesquisadores no processo de ensino-aprendizagem em ES são:

- 1) A existência de uma grande distância entre o que se ensina e a realidade existente no mercado de trabalho.
- 2) Os discentes têm pouco interesse pelas aulas maciçamente teóricas.
- 3) Em geral, os cursos de ES têm significativas taxas de evasão e reprovação.
- 4) Índices de aprendizagem efetiva bastante questionáveis.
- 5) É dada pouca ênfase ao trabalho em grupo.
- 6) Pouca prática de interdisciplinaridade.
- 7) Em geral, o que se ensina e aprende em um semestre ou disciplina, não é apropriado pela comunidade como um patrimônio intelectual para uso futuro.
- 8) há pouca ou nenhuma interação entre professores e alunos fora da sala de aula.

3. A Metodologia ESCollab

Nos últimos anos, Entidades como a ACM¹, SBC² e o MEC³ têm tentado estabelecer padrões e parâmetros norteadores para o ensino de Computação e Engenharia de *Software*.

Em 2001, a ACM publicou o “Computing Curricula 2001 Computer Science”, tentando estabelecer uma orientação para as instituições de ensino na área de computação (ACM, 2011).

Com a mesma intenção, em 2005, A SBC publicou o “Currículo de referência para os cursos de graduação em bacharelado em Ciência da Computação e Engenharia da computação” (SBC,2005).

Também em 2005, o MEC publicou as “Diretrizes curriculares de cursos da área de computação e informática”, que têm servido de padrões para o trabalho das comissões de especialistas do ministério que avaliam novas autorizações e/ou reconhecimento de cursos na área de computação (MEC, 2005).

O trabalho desenvolvido por estas instituições foi utilizado como parâmetro para formatar a metodologia norteadora do presente trabalho. Do ponto de vista da fundamentação em termos de teorias da aprendizagem, a ESCollab ancora-se em aspectos do Construtivismo de Bruner (BRUNER, 1996), da Teoria Sócio-Cultural de Vygotsky (DANIELS, 2003) e na Teoria da Cognição Distribuída de Hutchins (HUTCHINS, 1995).

Propõe-se a Metodologia ESCollab para nortear os processos de elaboração, manutenção e oferta de cursos voltados para ministrar os conteúdos programáticos da área de engenharia de *software*. Essa metodologia tem o seu foco na colaboração, embasando-se na abordagem 3C proposta por Ellis et al (1991), ancorada nos princípios da cooperação, comunicação e controle.

A abordagem proposta tem a preocupação de introduzir práticas didáticas inovadoras com estratégias multifacetadas para tornar o ensino desses conteúdos da engenharia de *software* mais atraente e interativo de tal forma que os alunos possam, envolvendo-se desde o início do curso, aprender mais e melhor. A ESCollab apropria-se dos mecanismos e recursos da própria Web 2.0 (redes sociais, blogs, wikis, documentos compartilhados, ferramentas de comunicação etc.).

¹ – ACM - Association for Computing Machinery

² - SBC – Sociedade Brasileira de Computação

³ - MEC – Ministério da Educação

A metodologia em questão especificamente proposta para o ensino de Engenharia de Software, baseia-se em 6 princípios básicos:

- **comunicação** – a interação permanente entre os diversos agentes envolvidos nas disciplinas integrantes da área de Engenharia de Software, tendo como suportes básicos as redes sociais e um ambiente *Web* voltado para tanto. Os agentes participantes do processo são: professores de ES, alunos atuais, ex-alunos, professores de outras instituições e profissionais da área.
- **cooperação** – aproveitamento do conhecimento produzido ao longo do processo ensino-aprendizagem em cada disciplina atualmente sendo ministrada e o seu repasse, dentro da IES e nas demais instituições participantes do processo, materializando o conceito de inteligência múltipla e cooperativa; máxima: tudo é público, nada é privado.
- **coordenação** - A atividade de coordenação inclui a identificação de objetivos, o mapeamento destes em tarefas e sua distribuição entre os participantes, sendo essas tarefas realizadas na ordem e tempo previstos e em conformidade com os objetivos e restrições determinados;
- **transparência** – os cursos deixam de ser mundos fechados restritos apenas ao binômio professor-aluno e passam a estar abertos à supervisão, ao acompanhamento e à participação da comunidade interna e externa da IES;
- **reutilização** – os artefatos produzidos por um docente poderão ser utilizados pelos demais docentes da área dentro da própria IES ou por professores de outras instituições, permitindo o compartilhamento de materiais didáticos e conseqüentemente a redução do esforço na montagem dos cursos;
- **responsabilidade social** – o processo ensino-aprendizagem deve estar conectado com as demandas sociais, voltando a sua ação e seus subprodutos para atender às necessidades da sociedade. Expor os agentes às áreas sociais menos abastadas como domínio para desenvolver projetos de fim de disciplina em cooperação com ONGs⁴, mostrou-se uma excelente forma de trabalhar o lado social em uma área eminentemente técnica.

3.1 Visão Conceitual da ESCollab

A ESCollab está pautada em dois pilares: o primeiro é a necessidade crescente de cooperação entre agentes p_i e p_j que ministram disciplinas $\langle e_i, c_i \rangle$ e $\langle e_j, c_j \rangle$ que tem relação de pré-requisitos entre $\langle e_i, c_i \rangle$ e $\langle e_j, c_j \rangle$; o outro é o apelo tecnológico para reaproveitar artefatos no processo de forma que crie uma cultura de reaproveitamento, e conseqüente, cooperação entre os agentes.

A metodologia ESCollab é subdividida em 3 fases distintas nas quais são definidas as entradas, saídas e participantes de cada etapa:

1) Planejamento: é a fase inicial da metodologia correspondendo ao planejamento da(s) disciplina(s) a serem ministradas. O planejamento deve acontecer entre os docentes pertencentes à subárea de ES.

Entradas: PPC – Projeto Pedagógico de Curso; Recomendações ACM e SBC; Diretrizes Curriculares do MEC; calendário escolar; levantamento bibliográfico na área/disciplina.

Saídas: Ementa(s), programa(s) da(s) disciplina(s), Bibliografia básica/complementar, planejamento de curso, formulário de auto-avaliação discente e avaliações diagnósticas inicial (a ser aplicada no início das aulas) e final (a ser aplicada no final das aulas). No planejamento de curso devem ser explicitados os recursos didáticos e estratégias a

⁴ Organizações Não Governamentais.

serem utilizadas ao longo da disciplina, devendo ser respeitadas as determinações quanto à estratégia e ferramental obrigatório da metodologia.

Participantes: docente titular da disciplina, docente(s) voluntário(s), coordenação de curso, supervisão pedagógica, membro(s) da sociedade civil.

2) Execução: é a fase na qual as aulas e atividades didáticas ocorrem.

Entradas: Ementa, programa da disciplina, plano de curso, Bibliografia básica/complementar, avaliação diagnóstica.

Saídas: Resultado da avaliação Diagnóstica Inicial, Plano de curso ajustado, registro das atividades didáticas, controle de frequência, resultado das avaliações, artigos, projetos concluídos, seminários apresentados, problemas resolvidos, Estudos de casos, listas de exercícios resolvidas, avaliações aplicadas, correções das avaliações aplicadas.

Participantes: docente titular da disciplina, docente(s) voluntário(s), membro(s) da sociedade civil e aluno(a)s.

3) Conclusão: corresponde à fase final da disciplina.

Entradas: Resultado da avaliação Diagnóstica Inicial, Plano de curso ajustado, registro das atividades didáticas, controle de frequência, resultado das avaliações, artigos, projetos concluídos, seminários apresentados, problemas resolvidos, Estudos de casos, listas de exercícios resolvidas, avaliações aplicadas, correções da avaliações aplicadas.

Saídas: Avaliação diagnóstica final, análise comparativa da evolução da aprendizagem da turma, avaliação da disciplina/docente, relatório da auto-avaliação dos discentes. Relatório final da disciplina

Participantes: docente titular da disciplina, docente(s) voluntário(s), membro(s) da sociedade civil, aluno(a)s, coordenação de curso, supervisão pedagógica.

A ESCollab elenca um conjunto de recomendações para que os agentes possam planejar suas atividades de forma cooperativa, utilizando os recursos da Web 2.0, proporcionando um repositório de objetos de aprendizagem voltado para o ensino da Engenharia de Software.

O fundamental é que p_i e p_j sintam a necessidade de fazerem um planejamento em conjunto ao ministrarem as disciplinas $\langle e_i, c_i \rangle$ e $\langle e_j, c_j \rangle$. Neste sentido, esses agentes devem especificar um mesmo conteúdo semântico para alguns artefatos:

- Toda Aula **AL** é composta por um conjunto de Recursos Didáticos - **RD**; que deve ser de domínio público e deve estar na Web.
- Um recurso pode ser um vídeo, um áudio, um texto, uma apresentação, um artigo científico, uma aplicação, ou um objeto de aprendizagem disponível na Web.
- O conjunto de alunos matriculados em uma disciplina $\langle e_i, c_i \rangle$ compõe o corpo discente, onde: cada membro do corpo discente deve possuir endereço eletrônico (e-mail) de domínio público; ser membro de uma rede social comum aos demais discentes; ter permissão de publicação na página da disciplina.
- Cada disciplina $\langle e_i, c_i \rangle$ deve possuir no mínimo, um colaborador voluntário da sociedade civil organizada tanto como especialista que vá trazer as histórias do usuário para determinação do problema como pode atuar na comunidade de desenvolvedores. Um projeto deve exigir a elaboração de, pelo menos, um projeto ao longo da disciplina; A disciplina deve adotar uma abordagem didático-pedagógica semelhante às melhores propostas encontradas na pesquisa feita por Paiva(2011a):
- empregar a técnica PBL (aprendizagem baseada em problemas); b) realizar Estudos de Caso – EC sobre os conteúdos da disciplina; c) a produção de Artigo(s) Científico(s) – ARTC como mecanismo de ensino-aprendizagem;

- os recursos associados às disciplinas de G, tais como $\langle e_i, c_i \rangle$, devem ser públicos, possuir uma página Web (contendo a e_i -*Ementa*; e o c_i – *Conteúdo Programático*, Plano de aulas a ser executado ao longo do período letivo; Links ou referências para os recursos associados; Links para os mecanismos de avaliação; Informações sobre Avaliações (Contínua e Somativa) e controle de frequência e/ou participação; Informações sobre os projetos, Problemas e Estudos de casos associados à disciplina;
- os mecanismos de avaliação devem fazer parte do planejamento e podem ser: mini-testes, avaliações objetivas e/ou avaliações subjetivas, estudo dirigido, projeto, estudos de caso, artigo científico e problemas, conforme Bordenave(2007);
- um projeto deve estar associado a cada disciplina $\langle e_i, c_i \rangle$ com o objetivo de materializar conceitos abordados ao longo do curso. O projeto deve ter um usuário real, oriundo de entidades públicas (creches, hospitais, orfanatos, escolas, sindicatos, cooperativas etc); ter o seu código fonte disponível para receber a participação/contribuição das comunidades de desenvolvedores de software;
- É fundamental na ESCollab que os agentes p_i e p_j planejem conjuntamente os seus cursos e as formas de avaliar o conteúdo trabalhado com os alunos das disciplinas $\langle e_i, c_i \rangle$ e $\langle e_j, c_j \rangle$, que têm relação de pré-requisitos entre $\langle e_i, c_i \rangle$ e $\langle e_j, c_j \rangle$.

4. Estudo de Caso

A metodologia ESCollab foi utilizada parcialmente pela primeira vez no semestre 2010.1 na turma do 4º período do curso de Ciência da Computação da UFPB, na disciplina Análise e Projeto de Sistemas I (60h). No semestre 2010.2, a mesma turma cursou a disciplina Análise e Projeto de Sistema II (60h) também sob a ótica da ESCollab, tendo sido registrado um ganho de cerca de 15% da carga horária em termos da não necessidade de revisão dos conteúdos da disciplina anterior, diminuição do índice histórico de evasão e aumento do interesse da turma pelas aulas.

No semestre 2011.1, foi introduzida outra IES no experimento, a Faculdade Asper, que possui o mesmo conceito no ENADE (4) que a UFPB. A turma pioneira da UFPB, então no 6º período, cursou a disciplina Especificação de *Software* I (60 h). Na Faculdade Asper, propositadamente, não foi aplicada a metodologia na disciplina Engenharia de Software (60 h) no semestre 2011.1. Para melhor aplicação do estudo de caso, acordou-se, antes do início das aulas, uma interseção de conteúdos programáticos superior a 90% entre as ementas das disciplinas.

Na UFPB, no semestre 2011.1, houve o planejamento conjunto com os professores que ensinam ES, os alunos foram divididos em 2 grupos rotativos, em cada atividade, compostos de 4 alunos e trabalharam ao longo do semestre no desenvolvimento do Sistema de Controle de Aluguel por Temporada. Os objetos de aprendizagem (ementa, apresentações, textos, listas de exercícios, estudos de casos, problemas etc) foram colocados à disposição da turma na página Web do professor, conforme preconiza a metodologia ESCollab. As ferramentas de interação foram o Google Groups e o Twitter, sendo usada como principal ferramenta de compartilhamento o Google Docs. A cada 15 dias foram aplicados mini-testes para aferir a aprendizagem dos conteúdos teóricos lecionados em sala. O curso foi acompanhado por um agente externo ao curso de graduação, que foi um aluno do mestrado em Informática, participando como um consultor externo.

Na Faculdade Asper, utilizou-se a abordagem tradicional. O docente fez o planejamento isoladamente, os alunos não desenvolveram projetos, a principal ferramenta didática foi a aula expositiva, foram avaliados apenas em duas provas bimestrais durante o curso e os objetos de aprendizagem foram sendo disponibilizados ao longo do semestre. A turma era composta por 16 alunos e não foram compostos grupos de trabalho. A comunicação foi feita através de uma lista de discussão da turma.

Uma prova objetiva sobre os assuntos comuns às duas disciplinas foi elaborada conjuntamente pelos docentes das duas turmas e aplicada no final do semestre aos alunos de ambas as turmas, mostrando, em média, uma diferença de aprendizagem da ordem de 40% a mais em favor da turma da UFPB, que utilizou a metodologia ESCollab.

Com os resultados promissores dos semestres anteriores, no semestre 2011.2, a metodologia ESCollab está sendo aplicada na íntegra na turma do 7º período, na disciplina Métodos Computacionais da UFPB e na Faculdade Asper, a aplicação efetiva está sendo feita em 3 disciplinas (Qualidade de Software, Sistemas de Informação Inteligentes e Desenvolvimento de Aplicações Distribuídas), no 8º período do curso de Ciência de computação, podendo ser percebida uma boa receptividade por parte da comunidade da IES.

5. Considerações Finais

A ESCollab é uma metodologia colaborativa que visa construir conhecimento e práticas de ensino voltadas para a área de Engenharia de Software com o propósito central de contribuir com a melhoria da formação dos profissionais envolvidos no desenvolvimento de software.

Os resultados preliminares foram encorajadores e confirmaram nossa hipótese de que algo precisava ser feito para criar uma **cultura de cooperação** entre os agentes do processo de ensino-aprendizagem das disciplinas de Engenharia de Software, como também contornar a carência de um maior rigor metodológico no ensino-aprendizagem na área de Computação.

O que pudemos comprovar na aplicação da metodologia ESCollab é que um pequeno esforço gerencial pode ter um impacto grande na qualidade final da formação de nossos alunos de graduação. A sistematização e a visão canônica de conceitos e do próprio processo utilizado no ensino-aprendizagem no experimento mostrou o quanto pode ser efetivo um pouco mais de formalização na gerência dos recursos didático-pedagógicos.

A título de trabalhos futuros, o grupo de pesquisa trabalha na implementação de um ambiente Web para facilitar a integração das diversas ferramentas utilizadas com a metodologia, assim como interagir com outros trabalhos de pesquisa centrados no desenvolvimento de repositórios de recursos compartilhados.

6. Bibliografia

- ACM/IEEE. (2001) Association for Computing Machinery. Computing Curricula 2001 Computer Science. Disponível em: <<http://www.acm.org>>. Acesso em: 29 mar.2010.
- ANISSETTY, Priatham; YOUNG, Paul. (2011) Collaboration problems in conduction a group project in a software engineering course. Journal of Computing Sciences in Colleges, Volume 26 Issue 5, Mai. 2011.

- BORDENAVE, Juan D.; PEREIRA, Adair M. (2007) Estratégias de Ensino-Aprendizagem. 28. ed. Petrópolis: Vozes.
- BRASIL. Ministério da Educação e Cultura. (2005) Diretrizes curriculares de cursos da área de computação e informática. Disponível em: <<http://www.portalme.gov.br>>. Acesso em: 30 jan. 2010.
- BRUNER, J. (1996) *The Culture of Education*. Cambridge, MA: Harvard University Press.
- COCCOLI, M.; STANGANELLI, L. & MARESCA, P. (2011) Computer Supported Collaborative Learning in software engineering. *Proc. IEEE Global Engineering Education Conf. (EDUCON)*, 2011, 990-995
- DANIELS, H. (2003) *Vygotsky e a Pedagogia*. São Paulo: Edições Loyola.
- ELLIS, C.A., GIBBS, S.J., REIN, G.L. (1991) Groupware - Some Issues and Experiences. *Communications of The ACM*, vol. 34, no. 1, pp. 38-58.
- GARG, K. & VARMA, V. (2009) Case Studies as Assessment Tools in Software Engineering Classrooms. *Proc. 22nd Conf. Software Engineering Education and Training CSEET '09*, 2009, 8-11
- GAROUSI, V. (2010) Applying Peer Reviews in Software Engineering Education: An Experiment and Lessons Learned. *#IEEE_J_EDU#*, 2010, 53, 182-193
- GIRALDO, F. D.; COLLAZOS, C. A.; OCHOA, S. F.; ZAPATA, S. & DE CLUNIE, G. T. (2010) Teaching Software Engineering from a Collaborative Perspective: Some Latin-American Experiences. *Proc. Workshop Database and Expert Systems Applications (DEXA)*, 2010, 97-101
- GOTEL, O.; KULKARNI, V.; SAY, M.; SCHARFF, C. & SUNETNANTA, T. (2009) A Global and Competition-Based Model for Fostering Technical and Soft Skills in Software Engineering Education. *Proc. 22nd Conf. Software Engineering Education and Training CSEET '09*, 2009, 271-278
- HUTCHINS, Edwin. (1995) *Cognition in the Wild*. Cambridge, MA: MIT Press.
- IBGE. (2010) Pesquisa de Serviços de Tecnologia da Informação 2009. Disponível: <<http://www.ibge.gov.br>>. Acesso em 20. mar. 2011.
- JIANGUO, L. (2009) Combination of Research and Teaching in Software Engineering Education. *Proc. WASE Int. Conf. Information Engineering ICIE '09*, 2009, 2, 437-440
- MIRIAN-HOSSEINABADI, S.-H.; AGHAKASIRI, Z.; SADEGHI, A.; DELFANI, P. & GHANDEHARI, M. (2010) Emphasizing experiences in teaching software engineering courses. *Proc. 2nd Int Education Technology and Computer (ICETC) Conf*, 2010, 2
- MOTTOK, J. & GARDEIA, A. (2011) The Regensburg Concept of P-Seminars - How to organize the interface between secondary school and university education to create a didactic cooperation between teaching and learning of Software Engineering with Lego Mindstorms NXT Embedded Robot Systems. *Proc. IEEE Global Engineering Education Conf. (EDUCON)*, 2011, 917-920
- PAIVA, Severino R. (2011a) Uma Revisão Sistemática das Pesquisas Realizadas sobre a melhoria no ensino de Engenharia de Software. João Pessoa, UFPB. Relatório Técnico – UFPB/CCEN.
- PAIVA, Severino R. (2011b) Pesquisa nacional sobre ensino de Engenharia de Software. João Pessoa, UFPB. Relatório Técnico – UFPB/CCEN.
- PRIKLADNICKI, Rafael et al. (2009) Ensino de Engenharia de Software: desafios, estratégias de ensino e lições aprendidas. In: XII SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE. Fortaleza.
- QIU, M. & CHEN, L. (2010) A Problem-Based Learning Approach to Teaching an Advanced Software Engineering Course. *Proc. Second Int Education Technology and Computer Science (ETCS) Workshop*, 2010, 3, 252-255
- SBC. Sociedade Brasileira de Computação. (2005) Currículo de referência para os cursos de graduação em bacharelado em Ciência da Computação e Engenharia da Computação.
- YADAV, S. S. & XIAHOU, J. (2010) Integrated project based learning in software engineering education. *Proc. Int Educational and Network Technology (ICENT) Conf*, 2010, 34-36