

BUSCANDO UMA INTEGRAÇÃO DOS CAMPOS DE CONHECIMENTO DA ENGENHARIA DE SOFTWARE ATRAVÉS DE PROJETOS

David M. B. Santos – davidmbs@gmail.com

Universidade Estadual de Feira de Santana, Departamento de Exatas

Av. Universitária, s/n - Km 03 da BR 116, Campus Universitário

44.031-460 - Feira de Santana - BA – Brasil

José A. M. Santos – zeamancio@ecomp.uefs.br

João B. R.-Junior – joao@ecomp.uefs.br

Gabriela R. P. R. Pinto – gabrielarprp@gmail.com

***Resumo:** Buscando minimizar os desafios observados no processo de ensino-aprendizagem de componentes curriculares tradicionais de Engenharia de Software, o curso de Engenharia de Computação da Universidade Estadual de Feira de Santana vem oferecendo Projetos (Anuais) Temáticos que integram vários campos do conhecimento, desenvolvidos por uma pequena equipe heterogênea de estudantes atuando em diferentes funções (desenvolvimento, engenharia e gerência). Este artigo descreve justamente esse componente curricular, Projeto Anual, e faz uma avaliação crítica desta abordagem baseada na experiência obtida de três docentes após oferecer este componente oito vezes, seguindo as orientações de um método de pesquisa denominado “experiência de vida e formação”.*

***Palavras-chave:** Projeto, Engenharia de Software, Interdisciplinaridade*

1 INTRODUÇÃO

O ensino de Engenharia de Software deve se aproximar ao máximo da realidade que o educando encontra fora do ambiente acadêmico, seja em uma esfera mais restrita, a profissional, ou uma mais ampla, a sociedade. Portanto, isso abrange muitos desafios. Primeiramente, o volume de informações na Engenharia de Software é muito extenso, envolve uma grande quantidade de tópicos: gerência de projetos, gerência de requisitos, conceitos e tecnologias orientadas a objetos, arquitetura de software, projeto e padrões, teste, inspeção e revisão de software, garantia da qualidade de software, métricas, processo de software, entre tantos outros (VON WANGENHEIM & SILVA 2009, SOMMERVILLE, 2003). Não é por acaso que, em meados de 2010, no Brasil, já tínhamos, pelo menos, cinco cursos de graduação em Engenharia de Software. A integração destes conteúdos de forma direta e articulados com a prática tem sido uma solução pleiteada por alguns cursos de graduação (SANTOS et al, 2007, ALVES E BENITTI, 2006, COSTA et al, 2008).

Entretanto, esse tratamento em componentes curriculares tradicionais tem se mostrado ainda insuficiente. Geralmente, as iniciativas observadas abrangem apenas os conceitos/práticas básicos, sem contemplar os conceitos/práticas avançados. Ademais, para Bittencourt e Figueiredo (2003), os cortes disciplinares nem sempre são eficazes, além de os componentes curriculares oferecidos frequentemente serem de curta duração e não oferecerem

uma experiência de trabalho em grupo e gerência de projetos próxima ao ambiente profissional. O efeito disso é o foco em uma formação específica, demasiadamente técnica, não abrangendo competências mais amplas necessárias à formação do engenheiro para tratar problemas mais complexos e interdisciplinares.

Uma alternativa para ampliar as possibilidades de integração de conhecimentos no processo de formação dos estudantes, encontrada pela comissão curricular do curso de Engenharia de Computação (EComp) da Universidade Estadual de Feira de Santana (UEFS) foi a elaboração e oferta de um componente curricular denominado Projeto Anual (PA) (BITTENCOURT & FIGUEIREDO, 2003). Em suma, o PA consiste em um projeto com tópicos relacionados ao curso, orientado por um professor e desenvolvido por uma equipe de seis discentes, em média, em diferentes níveis de atuação (e.g desenvolvedor, engenheiro e gerente).

Assim, o objetivo deste artigo é relatar a experiência de três de seus autores — docentes do referido curso —, a partir do acompanhamento de oito PA, com temas relativos à Engenharia de Software, avaliando seus desempenhos e apontando as possibilidades e os desafios observados. Segue, para tanto, as orientações de um método de pesquisa denominado “experiência de vida e formação”. Conforme Josso (2004), a formação torna-se um conceito gerador em torno do qual vêm agrupar-se, progressivamente, conceitos descritivos: processos, temporalidades, experiência, aprendizagem, conhecimento e saber-fazer, temática, tensão dialética, consciência, subjetividade, identidade etc. Para uma melhor compreensão do leitor, ele foi estruturado como descrito a seguir. Na Seção 2, apresentam-se mais detalhes do componente curricular e seu contexto no curso de EComp na UEFS. Em seguida, na Seção 3, descrevem-se os projetos anuais ofertados e as equipes envolvidas. A avaliação destes projetos está na Seção 4. Finalmente, as considerações finais e os trabalhos futuros estão traçados na Seção 5.

2 ESTRUTURA CURRICULAR DO PROJETO ANUAL

O curso de EComp da UEFS, criado em 2003, enfatiza a flexibilização curricular e a interdisciplinaridade, visando acompanhar as constantes transformações ocorridas na sociedade, especialmente na área de Computação. Um dos elementos de destaque na formação do currículo, a partir da proposta pedagógica do curso é o componente curricular PA, que ocorre ao longo de 2 semestres — e possui carga horária total de 150 horas divididas em 75 horas/semestre. O seu objetivo central é possibilitar ao estudante experimentar e vivenciar um ambiente profissional para o desenvolvimento de um produto tecnológico relacionado à sua formação acadêmica. O PA engloba três outros (sub)componentes: 1) Projeto Anual I; 2) Projeto Anual II; e 3) Projeto Anual III — conforme pode ser visto na Figura 1.

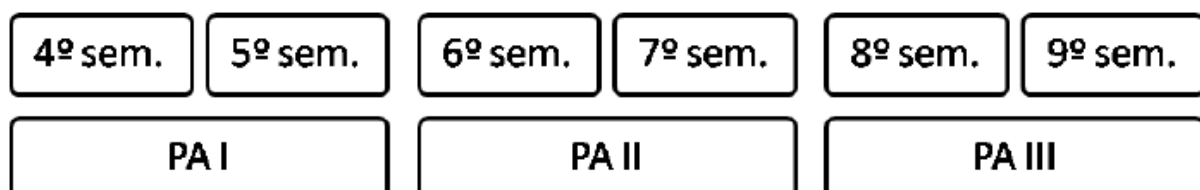


Figura 1: Estrutura do currículo para o Projeto Anual.

Dessa forma, cada estudante deve cursar o PA três vezes durante sua formação. Na primeira, o subcomponente é denominado Projeto Anual I e o pré-requisito exigido é que o estudante tenha concluído com aproveitamento, pelo menos, 20% da carga horária total do

curso de EComp. Neste subcomponente, o estudante realizará atividades introdutórias de desenvolvimento de software – basicamente, programação –, compatíveis com a sua formação prévia. Na segunda vez, o subcomponente é denominado Projeto Anual II e os pré-requisitos são que o estudante tenha concluído com aproveitamento, pelo menos, 40% da carga horária total do curso, e que tenha sido aprovado no Projeto Anual I. Aqui o estudante realizará algumas atividades de engenharia (eg. modelar dados e software, gerenciar a qualidade do software, modelar testes de unidade, etc). Na terceira e última vez, o subcomponente é referido como Projeto Anual III, tendo como pré-requisitos a conclusão de, pelo menos, 60% da carga horária total do curso, e a aprovação no Projeto Anual II. Nesta etapa, o estudante ajudará nas atividades de Engenharia de Software, além de desempenhar atividades de gerência, compatíveis com a experiência nos projetos anteriores e com sua formação prévia. A especificação precisa de cada função será ajustada de acordo cada projeto.

Cada PA possui seis vagas, duas para cada subcomponente. Em função dessa organização curricular, uma típica turma de PA é formada por seis estudantes, por exemplo, dois do quarto semestre, dois do sexto e dois do oitavo.

As propostas de projeto são submetidas pelos professores do curso para então serem avaliadas e selecionadas pelo Colegiado do curso. O critério para seleção dos PA que serão ofertados baseia-se na preferência dos estudantes. Antes de cada período de matrícula, em um período denominado de pré-matrícula, o Colegiado divulga para eles as propostas apresentadas pelos professores. Os estudantes preenchem uma tabela indicando uma ordem de preferência dos projetos que desejam participar. Com base nessas preferências, os projetos são pontuados e a quantidade necessária para atender àqueles pré-matriculados é estipulada de acordo com a pontuação obtida. Na matrícula, os estudantes são alocados ao PA de sua preferência, tendo o seu score no curso como critério de desempate. Assim, o que apresenta maior score é matriculado no projeto que definiu como sendo de maior prioridade; enquanto que os demais estudantes são alocados ao projeto de maior prioridade, desde que não tenha sido preenchido número máximo de 6 vagas por projeto. Caso haja mais vagas que solicitações, o Colegiado do curso decide como realizar a alocação. É comum, por exemplo, para evitar que PA sejam oferecidos com menos de 6 membros por turma, que alguns deles possuam uma quantidade maior de estudantes.

Para uma melhor compreensão do funcionamento do PA faremos a seguinte ilustração hipotética. Um estudante ao chegar no quarto semestre e que desejar se matricular no Projeto I tem três opções de projeto: A (do Professor A), B (do Professor B) e C (do Professor C). Sua ordem de preferência para ser alocado no PA foi A, C, B. Pelo seu score, que não era tão alto, ele acabou sendo lotado no projeto B. Tendo em vista os pré-requisitos, ele se matriculou no subcomponente Projeto Anual I, desempenhando assim, a função de desenvolvedor. Como o componente é anual, nada é alterado quando ele passa para o quinto semestre (vide Figura 1). Porém, ao chegar no sexto, há uma nova oferta de projetos: C (do Professor C), D (do Professor A) e E (do Professor H). Novamente este estudante fará uma nova ordem de prioridade sendo alocado em outro projeto, diferente daquele do PA I uma vez que o projeto B não foi ofertado novamente — e mesmo que fosse, não existiria a garantia de que o estudante participasse dele mais uma vez; primeiro, porque ele poderia atribuir baixa prioridade para o projeto; e, segundo, porque a alocação, como já dissemos, depende do score. Sendo assim, ele será matriculado no PA II, desempenhando desta vez funções de engenheiro. Note que a cada ano um projeto pode ser ofertado novamente, mas isso não é comum, geralmente, a lista pouco se repete. Em outras palavras, os professores acabam ofertando novos projetos. É importante destacar que é muito comum que os estudantes estejam em projetos distintos durante o curso dos subcomponente (PA I, PA II e PA III).

O processo de condução do PA é flexível, dependendo principalmente do professor e da própria natureza do Projeto. Em função da sua carga horária semestral de 75 horas, o

estudante deve cumprir uma carga horária de 5 horas semanais. Também devem ser realizadas reuniões semanais com duração de uma hora, com todos os integrantes do PA, em horário definido pelo Colegiado, que não coincida com nenhuma outra atividade dos envolvidos (estudantes e professor). Além disso, o professor deve se reunir individualmente com cada um deles durante 30 minutos, fora do horário da reunião do grupo, para orientar e acompanhar a execução das atividades. Obedecendo a estes requisitos, o professor tem flexibilidade para organizar o Projeto da forma como considerar mais conveniente para alcançar os objetivos de aprendizagem.

Os critérios estabelecidos para a avaliação da aprendizagem do estudante participante do PA também apresentam certa flexibilidade. A sua nota final é formada a partir da análise de três instrumentos (1. Desempenho. 2. Relatório final, 3. Comunicação oral). O primeiro deles é a nota de desempenho, com peso 7, a qual é a média aritmética entre três notas dada pelo professor do PA, sendo que duas delas devem ser dadas no primeiro semestre do PA e uma no segundo semestre. A flexibilidade neste caso ocorre porque cada professor define e apresenta ao estudante os critérios que serão utilizados para a geração destas notas de desempenho, que são apresentados através do plano de ensino disponibilizado logo no início das atividades. Há casos, por exemplo, de professores que geram uma nota de desempenho semanal para chegar às três notas obrigatórias. Além da nota de desempenho, os estudantes devem gerar um relatório final (peso 2) e uma apresentação (peso 1). Para ambos os instrumentos, uma única nota final é dada por dois professores: o professor do PA e outro convidado, geralmente do próprio curso de EComp. Caso o estudante não obtenha nota superior a 7,0, o professor do PA aplica uma outra avaliação, considerada "prova final", de acordo como considerar apropriado para o projeto.

3 PROJETOS ANUAIS AVALIADOS

Nesta seção apresentam-se em linhas gerais as propostas ofertadas como PA que são tema de estudo deste artigo. A Tabela 1 contém um resumo dos PA ofertados, contendo seu nome, objetivo geral e, na coluna configuração, o número de estudantes alocados em cada função (desenvolvedor, engenheiro e gerente).

Tabela 1: Lista resumida dos Projetos Anuais avaliados (D=Desenvolvedor, E=Engenheiro, G=Gerente).

Nome	Objetivo	Configuração
NPeer	aplicativo de backup de dados utilizando arquitetura peer-to-peer	6 D
SiGA	portal corporativo para gerenciar as atividades da área de informática do Departamento de Exatas da UEFS	3 D e 3 E
Smelly	plugin do Eclipse para detecção de maus cheiros em códigos JAVA desenvolvidos nesta IDE	2 D
MuSE	sistema de busca na Web com foco na recuperação de arquivos multimídia	2 D, 2 E, e 2 G
Odontosis-1	software para oferecer suporte ao atendimento nas clínicas odontológicas da UEFS, junto à população	2 D, 2 E, e 2 G
Odontosis-2	software para oferecer suporte ao atendimento nas clínicas odontológicas da UEFS, junto à população	3 D, 2 E, e 2 G
Jogoteca	aplicação para a TV Digital Interativa contendo um acervo de jogos eletrônicos para crianças	4 D, 1 E, e 1 G
Problem Database Manager	software WEB para oferecer suporte à criação e compartilhamento de problemas usados no método de ensino-aprendizagem do curso	2 D, 2 E, e 2 G

O projeto NPeer teve seis desenvolvedores porque foi oferecido para a primeira turma do curso, não dispondo, portanto, de engenheiros e gerentes. O mesmo vale para o SiGA que não teve gerentes. No caso do Smelly aconteceu uma situação atípica. Inicialmente, a equipe era composta por três desenvolvedores e três engenheiros de software — não foram alocados gerentes de projeto por conta dos poucos estudantes da primeira turma do curso, os únicos que podiam exercer tal função até então. Esta equipe rapidamente ficou reduzida porque quatro deles desistiram do projeto devido à falta de afinidade com sua temática. Os demais projetos seguem uma composição típica de um PA, com algumas pequenas variações. O OdontoSis aparece duas vezes porque foi ofertado por dois anos consecutivos.

Por restrições de espaço, descrevemos com um pouco mais detalhes apenas dois projetos: o Multimedia Search Engine (MuSE) (ROCHA-JÚNIOR et al. 2008) e o Odontosis (Santos et al. 2009). O primeiro tinha como objetivo desenvolver um sistema de busca na Web para arquivos multimídia. O projeto foi composto por 6 estudantes com a configuração típica de um PA, 2 estudantes em cada função. Um dos problemas enfrentados durante este projeto foi a gerência do grupo, pois, muitas vezes, os gerentes não tinham conhecimento suficiente para delegar tarefas precisas e que pudessem ser executadas nas 4 horas semanais de dedicação ao projeto. Isso gerou diversos problemas como desmotivação nos desenvolvedores, irritação nos engenheiros e difícil acompanhamento por parte do professor. Posteriormente, optou-se por definir um horário no qual todos os membros estivessem presentes para realizar o desenvolvimento em conjunto. Esta iniciativa mostrou-se bastante eficaz pois criou um ambiente similar ao vivenciado em uma empresa de software, o que motivou bastante os estudantes. Outro fator motivador foi a disponibilização do MuSE para que outros estudantes do curso pudessem opinar sobre os resultados de busca.

Já o Odontosis tinha como finalidade desenvolver software para oferecer suporte ao atendimento nas clínicas odontológicas da UEFS, sendo ofertado duas vezes. Na primeira vez, havia uma configuração típica de um PA e uma nota de desempenho semanal enquanto que, na segunda vez, eram 7 estudantes alocados, sendo 3 desenvolvedores e a nota era quinzenal. Em ambas as ofertas havia uma reunião semanal de três horas, onde todos trabalhavam em conjunto. Toda semana, o professor participava da reunião para acompanhar, ajudar a delegar tarefas e realizar avaliações, tanto do grupo como individuais. Os gerentes ficaram encarregados de realizar o levantamento de requisitos, junto com alguns professores do curso de Odontologia, além da definição e acompanhamento das atividades dos demais integrantes. Engenheiros e desenvolvedores ficavam com as tarefas de programação. A maior dificuldade do PA foi o fato de não haver disponibilidade e mesmo consenso por parte dos clientes quanto ao que deveria ser informatizado. Essa, que já é uma tarefa árdua, ficou prejudicada sobremaneira pela pouca experiência dos gerentes. Na segunda versão do PA, um outro problema com o levantamento de requisitos ocorreu: o cliente que oferecia maior suporte para o desenvolvimento afastou-se, e foi necessário um novo levantamento de requisitos para atender à nova visão de outros odontólogos. Assim, em função disso e de problemas técnicos, os estudantes resolveram reiniciar o desenvolvimento da ferramenta. Como principais resultados, podem ser apontados a participação e compreensão dos estudantes das dificuldades de um processo de software e conhecimento das tecnologias inerentes ao desenvolvimento para WEB.

A Tabela 2 aponta os principais tópicos de Engenharia de Software trabalhados em cada uma das propostas ofertadas. Em sua última linha indicamos os principais tipos de ferramentas CASEs usadas ao longo do projeto, tendo a seguinte correspondência: 1 - Controle de versão, 2 - Modelagem UML, 3 - Desenvolvimento (IDE), 4 - Automação de Testes, 5 - Prototipagem, 6 - Gerência de Projeto.

4 AVALIAÇÃO E DISCUSSÃO

Esta seção faz uma avaliação, baseada na experiência dos autores, dos sete PA apresentados na seção anterior. O principal objetivo é listar as principais possibilidades e desafios encontrados.

Tabela 2: Tópicos de Engenharia de Software abordados nos Projetos Anuais.

	NPeer	Smelly	SiGA	MuSE	OdontoSis	Jogoteca	PDM
Orientação a Objetos	✓	✓	✓	✓	✓	✓	✓
Arquitetura de Software		✓	✓	✓	✓	✓	✓
Testes de Unidade					✓		
Métricas de Software		✓					
Processo de Software				✓	✓	✓	✓
Gerência de Projetos				✓	✓	✓	✓
Usabilidade					✓	✓	✓
Levantamento de Requisitos				✓	✓		
Refatoração de código-fonte	✓	✓	✓	✓			
Framework					✓	✓	✓
Ferramentas CASE	1, 2, 3, 5	1, 2, 3	1, 2, 3, 5	1, 2, 3, 5, 6	1, 2, 3, 4, 6	1, 2, 3, 6	1, 2, 3, 6

Uma primeira possibilidade do PA é, indubitavelmente, uma formação do engenheiro que potencializa a sua capacidade de trabalhar em equipe. Os estudantes interagem intensamente entre si buscando cumprir o objetivo geral do projeto. Neste caminho, eles elencam e discutem idéias, falam, silenciam, escutam, retrucam. Produzem conhecimento coletivamente. Ao mesmo tempo em que entram em conflito, também vão aprendendo a enfrentá-los e solucioná-los. Para ilustrar, frequentemente há reclamações de membros que trabalham menos que outros ou que são menos interessados, gerando insatisfação — obviamente, gerente(s) e orientador devem ficar atentos a estas situações para que sejam resolvidas de imediato.

Os estudantes são incitados a definir e usar recursos de comunicação que ampliem a eficiência da troca de informações e a socialização de experiências entre eles. Isto fica evidente sobretudo nas reuniões individuais com o orientador e naquelas com toda a equipe onde eles se expressam oralmente. Esta última é uma das principais, pois é o momento que todos ficam a par do que cada um está realizando e dos conceitos, práticas e tecnologias envolvidos. Apesar destes benefícios, este cenário também apresenta dificuldades, principalmente porque, quando estão fora da reunião geral, trabalham muito isolados uma vez que não tem um horário e ambiente físico de trabalho compartilhados onde possam se encontrar. Alguns docentes contornaram isso alocando um laboratório em um determinado horário em algum dia da semana, onde assim, todos puderam dedicar suas horas (ou algumas delas) ao projeto, trabalhando em conjunto. Esta solução se mostrou bastante eficaz para a comunicação entre os membros e para o desenrolar do projeto. Os estudantes produziram muito mais quando trabalharam desta forma, juntos, do que quando realizaram atividades individuais. Essa experiência foi vivenciada nos projetos Smelly, Jogoteca, MuSE, Odontosis e Problem Database Manager. No entanto, nem sempre é possível reuni-los em um horário específico, devido ao choque de horários dos próprios estudantes, e dificuldade em encontrar um espaço apropriado para a realização das atividades. Este é um ajuste informal, não há horário alocado oficialmente para isto, apenas para a reunião geral, como descrito anteriormente (Seção 2).

Mesmo assim, é consenso entre os docentes que o tempo semanal determinado para cada estudante é curto para executar determinadas tarefas. Caso um estudante precise fazer um pequeno pedaço de software, com os constantes *bugs* e visitas à documentação, as quatro horas passam rapidamente e não se consegue cumprir a tarefa. Por conta disso, quebrar atividades em mini-tarefas de quatro horas não é fácil, frequentemente subestima-se o tamanho (complexidade) delas. Este fator nos leva a um dilema na avaliação: o estudante que dedicou as quatro horas para o trabalho, mas não cumpriu a meta, cumpriu a tarefa, ou melhor, recebe nota máxima? Como avaliá-lo de forma justa? É uma questão muito difícil. O que normalmente acontece é que o orientador faz uma arguição nos encontros individuais para saber se realmente ele buscou com afinco realizar a tarefa, verificando se o estudante dedicou o tempo estipulado para solucioná-la.

Nota-se que em muitos projetos, como Smelly, Jogoteca e Problem Database Manager, os estudantes acabaram por usar mais horas do que aquelas previstas oficialmente no currículo, o que pode gerar uma sobrecarga. Além disso, eles não têm muita experiência nos temas propostos, que, como vimos na seção anterior, são diversificados. Isso implica em uma fase de capacitação mais prolongada no início do projeto. Os desenvolvedores, por exemplo, até o Projeto Anual I, passaram apenas por dois componentes curriculares de programação, um deles que trata Orientação a Objetos, e, assim, não têm muita experiência. No projeto SiGA, aconteceu de algumas atividades programadas para serem realizadas pelos desenvolvedores em duas horas, demorarem mais de dois meses (32 horas) para serem concluídas. Se o mentor e os gerentes do projeto não tiverem cuidado com o escopo do projeto e a complexidade do desafio proposto, isso pode fazer com que o estudante tenha que tratar assuntos densos, esforçando-se por demais, quando já se tem uma carga horária curricular onerosa. Essa difícil relação com o tempo acaba fazendo com que os produtos gerados ainda sejam incompletos e/ou instáveis – os testes, por exemplo, comumente acabam ficando em segundo plano. Entretanto, se bem orientados, os desafios advindos do limite de tempo podem favorecer a aprendizagem e o desenvolvimento do estudante. Ao perceber que uma tarefa de duas horas se prolongou excessivamente, por exemplo, ele começa a notar que a estimativa que faz está equivocada, e que é preciso repensar a forma de como fazê-la. Assim, os educandos vão percebendo os obstáculos no dimensionamento de tarefas de software e outras mais no desenvolvimento de software, comuns a muitos profissionais da área. Claro que deve haver precaução para não acontecer de a experiência se tornar frustrante. Neste sentido, conclui-se que o escopo do projeto deve ser bem planejado e dimensionado para que se consiga alcançar a finalidade da proposta.

Noções de gerência de pessoal e projeto também são bastante exploradas. A equipe é composta de acordo com seu papel obrigatório no projeto (desenvolvedor, engenheiro ou gerente) e tarefas são divididas entre os membros, sendo as formas individual e em dupla as mais comuns. Um aspecto interessante aqui é que enquanto alguns estudantes não gostam de receber tarefas dos colegas, outros, quando gerentes, sentem-se desconfortáveis em exigí-las. Outro desafio dos estudantes é que, por falta de experiência, demonstram dificuldades em gerenciar projetos, por exemplo, inicialmente, não sabem por onde começar ou quebrar o projeto em macro-etapas e estas em micro-tarefas. Muitas vezes a visão de coordenador de projeto que falta, só vai aflorar no final; porém, aqui cabe ao orientador agir de tal forma a manter o objetivo do projeto em foco.

Outros desafios do PA dizem respeito a sua própria estrutura curricular. Como relatado na Seção 2, os estudantes são alocados nos projetos de acordo com suas ordens de preferência, sendo que aqueles com maiores escores têm prioridade. Em outras palavras, os estudantes com maiores notas no currículo conseguem ficar em projetos que têm afinidade, enquanto que para os demais isso nem sempre é possível. Ademais, o número de projetos ofertados, historicamente, em média, é o mesmo da demanda, o que, portanto, não gera muitas

opções. Isso acaba por desanimar os estudantes. Um caso que frequentemente ocorre é o de estudantes que têm afinidade pelos projetos envolvendo hardware às vezes serem alocados para um projeto de software (ou vice-versa), o que foi relatado no projeto Smelly. Para uma equipe que possui poucos membros, o desinteresse de um ou dois pode prejudicar o projeto, ainda mais quando estes são desenvolvedores ou engenheiros. Não há – e infelizmente, muito provavelmente, não haverá – como casar perfeitamente interesse dos estudantes e proposta dos professores.

Outro aspecto da estrutura do PA é a formação da equipe com dois estudantes em cada função. Dois gerentes para uma equipe de 6 é muito (ex. MuSE) ou 3 engenheiros e 3 desenvolvedores (SiGA). Uma configuração como a do Jogoteca funciona melhor com 4 desenvolvedores, 1 engenheiro e 1 gerente. Mas como fazer isso dando oportunidade igual para todos os estudantes de passarem por todas as funções? É uma questão de difícil solução, que os docentes frequentemente discutem.

Um aspecto interessante notado nestes projetos é que muitos deles tinham envolvimento com pesquisa científica. Inclusive, quatro deles (Smelly, OdontoSis, NPeer, MuSe) foram publicados em conferências diversas, desde regionais até internacionais. Porém, também se pode questionar se isso não interfere no projeto. Sendo mais específico, uma vez que o objetivo é vivenciar um ambiente profissional para o desenvolvimento de um produto tecnológico relacionado à sua formação acadêmica, será que atividades relacionadas à pesquisa, como revisão bibliográfica, não atrapalhariam o caminhar do projeto? Pois este já tem diversas outras atividades que tomam tempo, como o período de capacitação que citamos. Não se questiona aqui a qualidade que a pesquisa pode agregar ao projeto, mas focando no cerne do ensino-aprendizado do estudante. Neste sentido, a pesquisa pode fazer com que projetos sejam mais interessantes. O que se percebe é que, em caso de pesquisa, deve haver um bom senso, e não pender demais para um projeto de iniciação científica.

Vale destacar que, no momento da submissão deste trabalho, o curso de EComp está passando por uma reforma curricular que irá modificar a grade do curso. O PA tem sido, de forma bastante destacada, um dos pontos de maior discussão da reformulação. Uma evidência de sua importância no curso é o fato de que os professores pretendem mantê-lo, alterando apenas o seu formato, a fim de melhor operacionalizá-lo. Além dos desafios listados até aqui, também tem constado em pauta, nas reuniões da comissão curricular, a viabilidade do PA com relação à recursos físicos e humanos disponíveis em EComp, já que demanda um número excessivo de professores para a sua realização. Uma vez que a média é de 6 estudantes por turma, normalmente são necessários em torno de 14 projetos por ano. Este número tem contribuído para tornar a oferta do PA pouco viável.

Em função dos desafios supracitados, há uma tendência de manutenção do componente curricular PA em um formato diferente, que vise a minimizar os problemas identificados. A principal proposta é ampliar o número de estudantes de 6 para 15 ou 20 e garantir dois turnos completos (8 horas) em um laboratório para os professores que estivessem ofertando o PA. Isso garantiria um horário e local fixo de trabalho para professores e estudantes. Além disso, devido ao aumento de carga horária durante o ano, e com objetivo de diminuir a carga horária total do curso, o PA deve ser ofertado somente uma vez. Ou seja, cada estudante passaria a cursá-lo apenas uma vez durante o curso.

Há expectativa de que muita discussão ocorra até que se chegue a uma proposta viável, mas uma forte evidência dos benefícios trazidos pelo PA é que há um interesse pela maioria das pessoas envolvidas em mantê-lo, melhorando o seu formato em função da experiência adquirida. Está claro para nós que o PA proporciona uma visão mais ampla de assuntos que são abordados de forma pontual. Buscando atingir o objetivo geral do projeto, os estudantes interagem entre si, integrando conhecimentos, recorrendo tanto a novos conteúdos e práticas quanto aqueles abordados em componentes curriculares anteriores. Aliás, esse pragmatismo é

outro ponto forte do PA, quando os estudantes executam de forma práticas diversas tarefas do desenvolvimento de software e de tópicos relacionados à Engenharia de Software. Nota-se ainda uma maior animação por parte dos estudantes quando está prevista a liberação de pequenas versões do software que está sendo desenvolvido, ou seja, os resultados práticos parciais é um fator que contribui para a motivação do grupo.

5 CONSIDERAÇÕES FINAIS

Este artigo relata a experiência de professores que ministraram Projetos Anuais (PA) relacionados à Engenharia de Software no curso de Engenharia de Computação da UEFS. Após fazermos uma avaliação de tal experiência, concluímos os principais pontos positivos desta abordagem são: i) possibilidades de realizar um trabalho em equipe para desenvolver um problema mais complexo; ii) interação entre os estudantes para troca de experiências; iii) experiência de gerenciar uma equipe; iv) amadurecimento dos estudantes em função da vivência em um ambiente com pessoas com diferentes níveis de experiência; v) aprendizado de tecnologias que normalmente não são enfatizadas nas disciplinas teóricas, mas são bastante úteis no início da vida profissional dos estudantes. Por outro lado, como principais desafios ainda encontram-se: i) a pouca carga horária semanal alocada para o desenvolvimento das atividades; ii) a dificuldade em simular um ambiente de trabalho, devido à falta de infraestrutura e tempo disponível pelos estudantes que compõem o projeto; iii) a definição de um projeto factível de ser realizado pelos estudantes no tempo disponível; iv) a alocação dos estudantes para atuarem nos projetos de suas preferências; v) o número excessivo de docentes envolvidos nos projetos, que são demandados para a sua realização; vi) a composição desequilibrada de equipes.

Como trabalhos futuros, pretende-se aprofundar as discussões aqui levantadas com o intuito de auxiliar na futura reestruturação curricular do curso da UEFS, a fim de que seja possível minimizar os desafios observados pelos professores que ministraram este componente. Ademais, faz-se mister acrescentar a visão dos estudantes através de recursos como questionários ou entrevistas, por exemplo. Uma possível forma de contornar esses problemas pode ser através de um pequeno manual descrevendo as melhores práticas, contendo diretrizes de como os projetos devem ser elaborados, de como as equipes devem ser divididas e de como os professores devem gerenciar o componente para que um número maior de experiências sejam vivenciadas pelos estudantes.

Agradecimentos

Agradecemos à UEFS por nos fornecer os recursos necessários para o desenvolvimento do trabalho e aos professores, estudantes e funcionários do curso de EComp pelas colaborações diretas e indiretas neste trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

ALVES, A. G. e BENITTI, F. B. V. Processo de desenvolvimento integrando disciplinas de engenharia de software. Anais: XIV Workshop Sobre Educação Em Computação (WEI) – CSBC. Campo Grande, 2006. p. 206–215.

BITTENCOURT, R. A. e FIGUEIREDO, O. A. O currículo do curso de engenharia de computação da uefs: Flexibilização e integração curricular. Anais: XI Workshop Sobre Educação Em Computação (WEI) – CSBC. São Paulo, 2003. p. 171–182.

COSTA, H., RESENDE, A., e SILVEIRA, F. Relato de experiência de ensino de modelagem e implementação de software em um curso de graduação em ciência da computação. Anais: I Fórum De Educação Em Engenharia De Software (FEES) – SBES. Campinas, 2008.

JOSSO, Marie-Christine. Experiências de Vida e formação. São Paulo: Cortez, 2004.

ROCHA-JÚNIOR, J. B., CAVALCANTE, I., e MATOS-JÚNIOR, O. C. Muse: Multimedia search engine. Anais: XIV Simpósio Brasileiro De Sistemas Multimídias E Web. Vila Velha, 2008. p. 113–116.

SANTOS, D. M. B. et al. Integrando as disciplinas de engenharia de software, análise e projeto de sistemas e banco de dados utilizando PBL. Anais: XV Workshop Sobre Educação Em Computação (WEI) – CSBC. Rio de Janeiro, 2007. p. 66–75.

SANTOS, J. A. M., SANTOS, C. A. B., e ARAÚJO, R. Aplicação de aprendizagem cooperativa no desenvolvimento do software odontosis. Anais: VI Internal Conference On Engineering Computer Education - ICECE, Buenos Aires, 2009.

SOMMERVILLE, I. Engenharia de Software. São Paulo: Pearson, 2003.

VON WANGENHEIM, C. G. & SILVA, D. A. Qual conhecimento de engenharia de software é importante para um profissional de software? Anais: II Fórum De Educação Em Engenharia De Software (FEES) – SBES. Fortaleza, 2009.

IN THE SEARCH FOR AN INTEGRATION OF THE SOFTWARE ENGINEERING KNOWLEDGE AREAS THROUGH PROJECTS

Abstract: *In order to minimize limitations on teaching and learning of Software Engineering through traditional methods, the undergraduate course in Computing Engineering of UEFS has employed thematic projects that integrate several knowledge areas, developed by a small team of students working at different levels (development, engineering and management). This paper describes the Annual Project course and makes a critical evaluation based on the experience acquired by the teachers after offering this course eight times, following the guidelines of a research method called "life experience".*

Key-words: *Project, Software Engineering, Interdisciplinarity.*