Conduzindo Projetos Ágeis em um Contexto Multidisciplinar: Um Relato de Experiência

Aline Jaqueira, Roberta Coelho, Márcia Lucena

Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte (UFRN)
Natal –RN

aline_o@hotmail.com,{ roberta, marciaj }@dimap.ufrn.br

Abstract. *Collaborative learning relates theory and practical experience in solving problems in order prepare students for the enterprise environment. The possibility of linking the theoretical and practical knowledge during graduate courses gives students a practical experience while in the academic context. Not only, technical experience, but also, human experience regarding the development of relationship skills. Accordingly, we applied a practical project in a multidisciplinary context (Software Engineering and Database courses) in which the students applied Scrum agile methodology and a set of software development tools in order develop a software project. Such projects bring the student as close as possible to a real environment of software development. This article presents our experience, difficulties and lessons learned along a semester. Results have showed that applying a multidisciplinary project is a valuable initiative. We also present the perspective of students and teachers on the project.*

Resumo. *O aprendizado de forma colaborativa relaciona a teoria e a experiência prática na solução de problemas visando formar profissionais bem preparados para o mercado de trabalho. A possibilidade de associar o conhecimento teórico e prático durante a graduação proporciona aos alunos uma vivência prática de sua futura profissão. Não somente no sentido técnico, mas também nos relacionamentos interpessoais. Nesse sentido, aplicou-se um projeto prático multidisciplinar envolvendo as disciplinas Engenharia de Software e Banco de Dados, utilizando a metodologia ágil Scrum e um conjunto de ferramentas de desenvolvimento de software de forma a levar o aluno a estar mais próximo possível a um ambiente real de desenvolvimento de software. Este artigo apresenta nossa experiência, dificuldades e lições aprendidas durante a aplicação de um projeto multidisciplinar. Também é apresentada uma análise dos resultados, além das percepções dos alunos e professores com relação ao projeto.*

1. Introdução

Podemos perceber que cada vez mais busca-se um novo modelo de ensino onde os alunos não desempenhem somente o papel de simples receptores de informações, mas passem a ser construtores de seu próprio conhecimento [Lê 2002]. Tal modelo visa atender às exigências do mercado de trabalho, que demanda profissionais com

iniciativa, que saibam trabalhar em equipe para resolver problemas complexos [Laffey *et al*. 2003]. O emprego da multidisciplinaridade pode ser uma via para estabelecer esse perfil. A multidisciplinaridade é considerada importante para acabar com um ensino extremamente especializado, concentrado em uma única disciplina e ocorre quando "a solução de um problema torna necessário obter informação de duas ou mais ciências ou setores do conhecimento" [Morin 2002].

Também de acordo com Almeida (2003), aprender envolve planejar, desenvolver ações, gerenciar informações, estabelecer conexões, refletir sobre o processo em desenvolvimento de forma conjunta com os pares, desenvolver a interaprendizagem, a competência de resolver problemas em grupo e a autonomia em relação à busca, ao fazer e ao compreender. Esse é o modelo de aprendizagem colaborativa, que vem da experiência prática baseada na solução de problemas, no pensamento crítico e na interatividade entre alunos e professores [Litto 2004].

Em cursos de tecnologia da informação, utilizar um processo de design de software (PDS) adequado ao contexto acadêmico contribui para o sucesso do aprendizado. Entende-se por academicamente adequado um PDS que considere [Lima *et al*. 2009]: (i) projetos com escopo pequeno ou médio a serem desenvolvidos dentro do período de um semestre letivo; (ii) a presença de clientes reais; (iii) o uso de processos e tecnologias considerados *estado da arte*; (iv) o aprendizado colaborativo; (v) agilidade, como conseqüência direta do tempo limitado, do escopo restrito e do aprendizado colaborativo; (vi) avaliação contínua e iterativa do progresso do projeto para que o aprendizado dos alunos possa ser avaliado ao longo da disciplina. Assim, definimos e aplicamos um projeto prático multidisciplinar envolvendo em PDS com as disciplinas Engenharia de Software e Banco de Dados, do curso de Ciência da Computação da Universidade Federal do Rio Grande do Norte. A possibilidade de associar o conhecimento teórico e prático na graduação vai proporcionar aos alunos uma vivência da prática de sua futura profissão. Não somente no sentido técnico, mas também social no que diz respeito ao trabalho em equipe, liderança e comprometimento [Pinto *et al*. 2010].

Neste trabalho são descritas nossa experiência, dificuldades e lições aprendidas ao longo da aplicação do projeto multidisciplinar. Também é apresentada uma análise dos resultados, além das percepções dos alunos e professores com relação ao projeto.

Este artigo está estruturado da seguinte forma. Na Seção 2 o projeto prático multidisciplinar é descrito. Na Seção 3 são apresentadas as tecnologias adotadas no projeto. Na Seção 4 as lições aprendidas são apresentadas. Por fim, na Seção 5 são apresentados as conclusões e trabalhos futuros.

2. O Projeto Multidisciplinar

O projeto multidisciplinar foi realizado com os alunos de graduação matriculados nas disciplinas de Engenharia de Software e Banco de Dados do curso de Ciência da Computação da Universidade Federal do Rio Grande do Norte em um semestre letivo, como citamos anteriormente. Um total de 65 alunos participou do projeto. 18 equipes foram formadas e cada uma escolheu seu domínio para desenvolver as atividades.

Neste projeto os alunos foram orientados a utilizar a metodologia ágil Scrum [Schwaber 1997]. As tecnologias adotadas no projeto foram: o framework Play [PlayFramework 2012] para desenvolvimento de sistemas Web; Google Code como repositório de código [GoogleCode 2012]; SVN para gerenciar versões, PostgreSQL [PostgreSQL 2012] para gerenciamento dos dados e o Acunote [Acunote 2012] para gerenciar as sprints Scrum. A tabela 1 apresenta detalhes do projeto.

Tabela 1 – Estrutura do Projeto

Duração do Projeto	3 meses
Tamanhos dos grupos	3 a 5 alunos
Número de Sprints	3 sprints de 1 mês
Objetivo Sprint 1	Definir os requisitos iniciais do sistema. Definir o modelo de dados inicial (ER) – apenas para alunos BD. Incluir o Planejamento no Acunote. Priorizar requisitos para a próxima Sprint.
Objetivo Sprint 2	Implementar requisitos no framework PLAY Utilizar Ferramenta de Controle de Versão Atualizar o Acunote
Objetivo Sprint 3	Entrega final do projeto com o software funcionando.

Como apresentado na tabela 1, no contexto da disciplina foram realizadas 3 s*prints*. Ao final de cada s*print* os grupos das duas disciplinas realizavam uma apresentação formal em um horário compartilhado. Para acompanhar o andamento dos projetos semanalmente, foi definido o papel de *Tutor* desempenhado por uma aluna de pós-graduação realizando estágio docência em uma das disciplinas. O *Tutor* era responsável por: (i) orientar sobre o uso correto da metodologia de desenvolvimento e das tecnologias adotadas e (ii) acompanhar o andamento dos projetos avaliando as informações inseridas no *Acunote* de cada grupo, orientando no desenvolvimento de tarefas e resolvendo conflitos de modo a contribuir para o bom o andamento dos projetos.

Com o objetivo de avaliar a experiência do projeto multidisciplinar para posteriormente relatar seus pontos positivos e desafios, foram analisados os procedimentos dos alunos e professores em suas práticas no decorrer do projeto, tais como (i) a participação das equipes, (ii) o rendimento dos alunos e (iii) as dificuldades enfrentadas por ambos. Foram avaliados pela *tutora* o andamento e progresso dos projetos e a frequência dos alunos na monitoria. De acordo com o andamento mensal dos projetos era atribuída pela *tutora* uma nota de 0 a 1 às equipes, de maneira a incentivá-las na continuidade e melhoria dos projetos. Os professores das disciplinas avaliavam se os requisitos do software solicitados para as apresentações mensais tinham sido atendidos. Na Seção 4 é apresentado o resultado dessa avaliação.

3. Tecnologias Adotadas

Nesta seção apresentaremos as tecnologias adotadas para o desenvolvimento dos projetos.

Scrum. O framework Scrum [Schwaber 1997] foi escolhido para o projeto por ser bastante utilizado para gerenciar e controlar o desenvolvimento ágil de software. Utiliza práticas iterativas e incrementais, assumindo que o desenvolvimento de software é imprevisível e só pode ser completamente descrito durante sua própria evolução. Dessa maneira, aumenta a flexibilidade e procura produzir um software que responda aos requisitos iniciais e adicionais descobertos durante o curso do desenvolvimento. O Scrum foi apresentado aos alunos com seus papéis determinados (dono do produto, Scrum Master e equipe), suas principais características (planejamento, reuniões diárias, de revisão e retrospectiva) e seus artefatos (product backlog, sprint backlog e burndown charts). A cada *sprint* os alunos deveriam selecionar um dos membros do grupo para desempenhar o papel de *Scrum Master* o qual seria o principal responsável pelo sucesso ou fracasso da *sprint*. A orientação foi que durante a execução dos projetos o grupo se revezasse no papel de *Scrum Master*, de modo que todos tivessem essa experiência. O papel do *dono do produto* era desempenhado pelo cliente real escolhido livremente pelas equipes.

Framework Play. Play [PlayFramework 2012] é um framework de alta produtividade que integra os componentes e APIs[1] necessários para o desenvolvimento de aplicações Web modernas. É baseado em uma arquitetura leve, seguindo o padrão MVC (model view controller) e apresenta características previsíveis e mínimo consumo de recursos. Uma palestra para apresentar a ideia geral do framework foi dada aos alunos do projeto. Além disso, foram apresentadas também suas possíveis fontes de consulta. A escolha deste framework foi pela praticidade de criar sistemas Web sem precisar de ter grande conhecimento de desenvolvimento Web e assim os alunos estariam mais focados no domínio do problema e não em entender a fundo as tecnologias Web.

Repositório de Código e Gerenciador de Versões. Utilizamos o repositório de código do Google Code [GoogleCode 2012] combinado com o cliente SVN (SubVersioN) do Eclipse. Assim os alunos puderam controlar as alterações no sistema, através do compartilhamento dos projetos em repositórios que guardam inclusive a versão anterior à alteração.

PostgreSQL. Para desenvolvimento da base de dados foi utilizado o gerenciador de banco de dados postgresSQL [PostgreSQL 2012] considerado um dos mais robustos e completos, além de ser um software de código aberto. A medida que os conteúdos de banco de dados eram apresentados aos alunos eles usavam na prática os conceitos vistos em sala de aula tanto para entender e melhor assimilar o conteúdo como para desenvolver o projeto.

[1] *Application Programming Interface*

Acunote. O Acunote [Acunote 2012] é uma ferramenta para gerenciamento de projetos baseado em *Scrum*. Escolhemos ess a ferramenta por ser de fácil utilização e possuir uma versão *freeware* disponível. O Acunote permite aos alunos atribuir estimativas de tempo, estimar as tarefas e alterar o progresso das mesmas até que sejam concluídas. Além disso, gera gráficos *Burndown* do trabalho restante, muito úteis em um processo Scrum. O Acunote permite que cada membro da equipe possa rastrear seu próprio progresso e o progresso da equipe ao visualizar a quantidade de trabalho estimado, concluído e restante para cada um.

4. Lições aprendidas

Organizamos nesta seção as lições aprendidas neste trabalho, do ponto de vista dos alunos (Seção 4.1) e do ponto de vista dos professores e do tutor (Seção 4.2) de acordo com as variáveis analisadas conforme apresentadas na Seção 2.

4.1 Sob o Ponto de Vista dos Alunos

No final do semestre, como tarefa, foi solicitado aos alunos que respondessem a um questionário de avaliação da execução dos seus projetos. O questionário consistia em três tópicos principais: (i) os dados das equipes, (ii) os requisitos do software, (iii) o uso do framework Play e mais um espaço para comentários espontâneos.

Com relação ao tópico referente aos requisitos do software desenvolvido, destacou-se o fato de a maioria das respostas afirmarem que o processo de elicitação foi bom, simples e sem problemas. É fato que os requisitos mudam de acordo com o aprendizado do cliente a cada validação ou interação com o software. Nuseibeh and Easterbrook (2000) afirmam que naturalmente existe um número de dificuldades inerentes ao processo de engenharia de requisitos. Portanto, a afirmação dos alunos sobre a facilidade na elicitação dos requisitos e o fato de os mesmos não necessitarem de modificação no decorrer do projeto, mostrou-se contraditória, principalmente para uma equipe de iniciantes.

A partir dessa contradição, levantamos os seguintes questionamentos: apesar de terem relatado que foram realizadas reuniões mensais com os clientes do software, como poderíamos garantir se as mesmas foram efetivamente realizadas de acordo com o que exige o *Scrum*? Como podemos garantir que as equipes buscaram realmente clientes reais conforme orientado ou elas mesmas fizeram esse papel? Através de entrevistas com os alunos, percebemos que apesar de a maioria utilizar clientes reais, os mesmos não se reuniram com frequência. As razões apresentadas são desde a falta de tempo do cliente e da equipe, que também foi citado como desafio em Pinto *et al.* (2010), até a compreensão correta de que os requisitos mudam com o tempo. Diante de nossa experiência, concluímos que em projetos reais cabe ao próprio professor entrar em contato com clientes reais que tenham disponibilidade de participar de reuniões mensais e que em contrapartida receberão o software gratuitamente. Outra alternativa seria o professor desempenhar o papel de cliente, porém isto traria outros problemas como: o professor pode ficar sobrecarregado com as tarefas de cliente e avaliador; o alunos não terão experiência de um cliente real.

A maior dificuldade relatada foi quanto à utilização do framework Play onde a maioria reportou no questionário que não conhecia e nunca tinha utilizado o framework. Encontramos alguns comentários relacionados à dificuldade em aprender o Play e encontrar fontes de consulta na literatura, estes comentários foram reportados por alunos que também responderam que tinham pouca experiência com programação para Web. Porém os alunos que já tinham experiência maior em programação ou já tinham cursado disciplinas de programação Web, demonstraram melhor aceitação ao framework Play. Ressaltaram como foi simples desenvolver um sistema Web em camadas. O funcionamento do Play também os ajudou no entendimento do modelo arquitetural MVC.

Com relação aos comentários espontâneos nos questionários, houve algumas sugestões de outros frameworks de desenvolvimento em substituição ao Play. Esta sugestão de substituição do framework se deve ao fato de que existem outros de maior utilização pelos profissionais, mais tempo no mercado e maior facilidade em encontrar consulta. Houve também manifestações a respeito da pouca orientação sobre o Play e Acunote.

4.2 Sob o Ponto de Vista dos Professores e do Tutor

Pode ser observado que as maiores dificuldades relatadas pelos alunos estavam relacionadas ao uso correto das ferramentas e metodologias nos projetos. Porém observamos que ao longo do semestre poucos alunos compareceram às seções de tutoria para tirar dúvidas ou enviaram emails sobre o Acunote. No entanto relataram pouca orientação, sendo que havia uma tutora disponível duas horas semanais. Do ponto de vista pedagógico, destacou-se o não comparecimento dos alunos ao laboratório para tirar dúvidas com a tutora e o interesse em desenvolver somente as atividades baseadas em notas. Foram enviados emails aos alunos cobrando a participação e *feedback* dos projetos à tutora, ainda assim a participação foi mínima. Percebemos que isto pode ser um indicativo da falta de interesse do aluno em se dedicar ao projeto em horário extra classe. Isto nos levou a propor que juntamente com as disciplinas Engenharia de Software e Banco de Dados haja uma disciplina co-requisito chamada de Projeto de Software onde os alunos terão a carga horária, dedicada ao projeto, e contabilizada nos créditos do curso.

Um ponto positivo que vale a pena ser ressaltado é que todas as equipes desenvolveram uma versão executável do projeto e atenderam de 40% a 80% dos requisitos solicitados. Podemos dizer que, o saldo da experiência multidisciplinar foi positivo, pois foram geradas versões funcionais do software. Escutando relatos de professores de outros períodos da disciplina Engenharia de Software, muitas vezes na última fase do projeto eram entregues apenas a especificação de requisitos e a modelagem, eram raros os grupos que entregavam uma versão funcional do sistema.

Percebemos também que o interesse dos alunos foi mais acentuado para aqueles que desenvolveram projetos inovadores e tinham inclusive vontade de colocar o mesmo em produção para os clientes. Então cabe aos professores elaborarem propostas de projetos que não sejam meros sistemas CRUD[2] para exercício das práticas de

[2] Create, Retrieve, Update e Delete.

engenharia de software e banco de dados. Este tipo de sistema desmotiva o aluno e não estimula o aprendizado. Como uma forma de motivar melhor o uso de novas tecnologias, pretendemos usar gerenciadores de banco de dados mais flexíveis e modernos no contexto de aplicações Web como os bancos de dados da categoria NoSQL [Lóscio *et al.* 2011].

Mais uma observação do ponto de vista dos professores é que as disciplinas tiveram seus conteúdos prejudicados em virtude das orientações e apresentações dos projetos. Se por um lado projetos práticos auxiliam na fixação do conteúdo das disciplinas por outro lado tais projetos consomem muito tempo para acompanhamento e execução. No caso deste projeto, esse tempo precisou ser retirado do espaço dedicado às disciplinas envolvidas. Esta observação reforça a necessidade de uma disciplina com créditos dedicados apenas ao projeto.

Do ponto de vista das práticas *Scrum*, a solicitação de revezamento do *scrum master* não funcionou. Durante as apresentações foi notório que somente um aluno fez esse papel assumindo a gerência do projeto. Os perfis de liderança e comprometimento muitas vezes não são comuns a todos e aqueles que tinham um perfil mais comprometido acabavam por gerenciar naturalmente o trabalho em equipe.

5. Conclusões e trabalhos futuros

Este artigo apresentou um relato de experiência da aplicação de um projeto prático multidisciplinar envolvendo as disciplinas Engenharia de Software e Banco de Dados com alunos da graduação em um semestre letivo. A aplicação do projeto nos permitiu elencar observações e levantar questionamentos.

Outras experiências semelhantes já foram realizadas como em Yamamoto *et al.* (2005), Alves e Benitti (2006), Cunha *et al.* (2008), Segura *et al.* (2009), Costa *et al.* (2010) O diferencial deste projeto é que foi realizado em apenas um semestre letivo, com domínios diferentes para as equipes, as apresentações e correções das *sprints* foram feitas em sala de aula para toda a turma, havia uma tutora disponível para orientar os alunos e foi realizada uma análise após sua aplicação.

A execução do projeto multidisciplinar exigiu bastante dos alunos, pois eles precisaram aprender os conceitos e praticar ao mesmo tempo o desenvolvimento do trabalho. Além disso, eram muitas novidades a serem assimiladas em paralelo ao conteúdo das disciplinas. A partir desta experiência sentimos a necessidade de uma disciplina exclusiva para a prática do projeto, onde os alunos poderiam se dedicar e ter um tempo maior para aulas destinadas ao uso das ferramentas e à realização do trabalho. Uma constatação secundária, porém relevante, foi que a prática autodidata era pouco desenvolvida nos alunos que participaram do projeto, ou seja, o perfil investigativo da curiosidade, ou até mesmo o interesse em procurar ajuda, era visto em poucos alunos, já que muitos relataram dificuldades mas em contrapartida não compareceram à monitoria.

Através da experiência da aplicação com o projeto multidisciplinar, das observações e questionamentos oriundos da prática, pode-se refletir sobre a prática multidisciplinar, conhecer os desafios enfrentados a fim de propor mudanças e melhorias. Como trabalhos futuros, propõe-se aplicar o projeto em uma disciplina exclusiva de modo a verificar se as dificuldades aqui apresentadas serão vencidas.

Referências

Acunote (2012). http://www.acunote.com/ Acesso em Maio 2012.

Almeida, M. E. B. (2003). Educação a distância na internet: abordagens e contribuições dos ambientes digitais de aprendizagem. *Educação e Pesquisa*, 29(2):327–340

Alves, A. G.; Benitti, F. B. V. (2006) Processo de Desenvolvimento Integrando Disciplinas de Engenharia de Software. Anais do XXVI Congresso da SBC. Campo Grande, MS.

B. Nuseibeh and S. Easterbrook. (2000). Requirements Engineering: A Roadmap , Proceedings of International Conference on Software Engineering (ICSE-2000), Limerick, Ireland, ACM Press.

Code.Google (2012) http://code.google.com/intl/pt-BR/ Acesso em Maio 2012.

Costa, C.; Rocha, R.; Fiqueirêdo, J.; Duarte, M.; Meira, S.; Prikladnicki, R. (2010) Ensino da Engenharia de Software por meio de Fábricas de Software no contexto Distribuído: Um Relato de Experiência. III Fórum de Educação em Engenharia de Software.

Cunha, A. M.; Braga e Silva, G.; Monte-Mor, J. A.; Domiciano, M. A.P.; Vieira, R. G. (2008) Estudo de Caso Abrangendo o Ensino Interdisciplinar de engenharia de Software. Fórum de Educação em Engenharia de Software. Disponível em: http://fees.inf.puc-rio.br/FEESArtigos/ Acesso em Junho 2012.

Laffey, J. M., Musser, D., Remidez, H., and Gottdenker, J. (2003). Networked systems for schools that learn. *Communications of the ACM*, 46(9):192–200.

Lê, T. (2002). Collaborate to learn and learn to collaborate. In Proceedings of the Seventh world conference on computers in education, pages 67–70, Copenhagen, Denmark. ACM.

Lima, G. A. N., Aguiar, Y. P. C., Lula, B. Lula Jr (2009). Impacto do Apoio Metodológico Ferramental a Aspectos de Usabilidade no Ensino Prático da Engenharia de Software. In XVII WEI – XXIX Congresso da Sociedade Brasileira de Computação, Bento Gonçalves.

Litto, F. M. (2004). A universidade e o futuro do planeta. In Siqueira, E., editor, *2015 – Como viveremos*, pages 211–214. Saraiva.

Lóscio, B., Oliveira, H., Pontes, J. (2011) NoSQL no desenvolvimento de aplicações Web. Mini-curso do VIII Simpósio Brasileiro de Sistemas Colaborativos, Parati, RJ.

Morin, E. A cabeça bem feita: repensar a reforma, reformar o pensamento. 6. ed. Rio de Janeiro : Bertrand Brasil, 2002.

Pinto, C. L. Q., Rocha, C. R. C., Vilarim, G. (2010). Desafios da Prática da Interdisciplinaridade em Cursos de Ciência da Computação: a Experiência do UNIFESO. In XVIII WEI – XXX Congresso da Sociedade Brasileira de Computação, Belo Horizonte.

Play framework. (2012) http://www.playframework.org/ Acesso em Maio 2012.

PostgreSQL. (2012) http://www.postgresql.org/ Acesso em Maio 2012.

Schwaber, K. Scrum Development Process. (1997). In OOPSLA Business Object Design and Implementation Workshop, J. Sutherland, D. Patel, C. Casanave, *J.* Miller, and G. Hollowell, Eds. London: Springer, 1997.

Segura, R. A.; Araújo Júnior, C. F.; Silveira, I. F. (2009) Aprendizagem Baseada em Problemas Apoiada por Ambientes Virtuais: um Estudo de Caso em Banco de Dados. Anais do XXIX Congresso da SBC, Bento Gonçalves, RS.

Yamamoto, F. S.; Silva, A. F.; Zanutto, J.; Zampirolli, F. A. (2005) Interdisciplinaridade no Ensino de Ciência da Computação. Anais do XXV Congresso da SBC, Unisinos, São Leopoldo, RS.