

Uso de Métodos e Técnicas para Desenvolvimento de Software Educacional em Universidades Brasileiras

Danilo L. Dalmon¹, Anarosa A. F. Brandão², Leônidas O. Brandão¹

¹Instituto de Matemática e Estatística – Universidade de São Paulo (USP)
São Paulo – SP – Brasil

²Escola Politécnica – Universidade de São Paulo
São Paulo – SP – Brasil

{ddalmon, leo}@ime.usp.br, anarosa.brandao@poli.usp.br

Abstract. *Educational software can provide many benefits to teachers and students. However, the experience of our research group shows that problems on software quality can disturb desired benefits and lead to difficulties in tasks such as software maintenance and evolution. These problems are usually caused by the adoption of ad-hoc development techniques, instead of those based on Software Engineering theories and practices. This paper presents the challenge faced by research groups in Technology Enhanced Learning in Brazil to adopt development techniques and methods, analyzing their contexts and possible solutions.*

Resumo. *Software educacionais possuem um grande potencial para promover benefícios a professores e alunos. Porém, a experiência de nosso grupo de pesquisa mostra que problemas na qualidade do software podem afetar os benefícios almejados aos usuários e ocasionar dificuldades nas tarefas de manutenção e atualização de software. Esses problemas são geralmente provocados pela adoção de técnicas ad-hoc de desenvolvimento, em vez daquelas baseadas em teorias e práticas da Engenharia de Software. Este artigo apresenta o desafio que grupos de pesquisa em Informática na Educação no Brasil têm para adotar essas técnicas e métodos sistemáticos de desenvolvimento, analisando seu contexto e possíveis soluções.*

1. Introdução

Software educacionais são projetados especificamente para desenvolver uma atividade favorável a alcançar os objetivos pedagógicos do contexto considerado (Tchounikine, 2011). Existem diversos desafios enfrentados pela educação que podem ser beneficiados por software educacionais, podendo agrupá-los em aprimoramento da aprendizagem ou do ensino. Do ponto de vista do aluno, alguns exemplos de como o software educacional pode contribuir são: melhorar o envolvimento do aluno em atividades que eram realizadas apenas na lousa e com papel (Marrades & Gutiérrez, 2001), criar experiências exclusivas com o uso do computador (Koerdinger & Alevén, 2007) e facilitar o acesso a informação e comunicação entre aprendizes e instrutores (Richards et al., 2002).

Do ponto de vista do professor, exemplos chave de melhoria que os software educacionais podem proporcionar são aumentar a agilidade ou automatizar

procedimentos repetitivos (Bridge & Appleyard, 2008) e a facilidade de acesso a boas atividades educacionais (van Dam & Simpson, 2005). Isso é possibilitado por sistemas gerenciadores de curso e repositórios integrados a eles, o que permite ao professor encontrar atividades adequadas ao seu contexto.

No Brasil, a criação de sistemas educacionais é realizada em muitas universidades por grupos de pesquisa em Informática na Educação (IE), além de empresas de desenvolvimento de software. O desenvolvimento de software educacional em universidades possibilita a experimentação de novos conceitos e paradigmas em protótipos que podem ser levados posteriormente ao mercado. Apesar desses sistemas promoverem diversos benefícios, nossa experiência mostra que sua qualidade computacional afeta as contribuições almejadas. Por qualidade computacional entendemos a qualidade técnica do código que constitui o sistema (Ben-Menachem & Marliss, 1997). Assim, mesmo que o software tenha uma qualidade pedagógica alta devida ao projeto instrucional bem elaborado, se sua qualidade computacional for baixa, haverá problemas para programadores realizarem sua manutenção e evolução, diminuindo a aplicabilidade de projeto instrucional e os benefícios que o mesmo poderia trazer ao processo de ensino e aprendizagem. Problemas relacionados à qualidade computacional são geralmente provocados pela falta de adoção de métodos e técnicas sistemáticas de desenvolvimento, que são o foco deste artigo. Isso caracteriza o desafio proposto: *a dificuldade de adoção de técnicas e métodos sistemáticos de desenvolvimento por grupos de pesquisa em IE no Brasil*.

Para levantar evidências desse desafio, foram usados três mecanismos: (i) leitura sistemática dos artigos publicados no Simpósio Brasileiro de Informática na Educação (SBIE); (ii) questionário entregue a desenvolvedores em um *workshop*; e (iii) entrevistas com programadores de diversos grupos de pesquisa. A partir desses dados, obtivemos informações para investigar possíveis causas e propormos sugestões para vencê-lo.

O artigo está organizado da seguinte maneira. A próxima seção fundamenta o desafio, com uma revisão da literatura de artigos de desenvolvimento de software educacional com relação ao método ou técnicas de desenvolvimento utilizados e relatados. Na seção 3, são apresentados dados do levantamento sobre o uso desses métodos e técnicas nas universidades do Brasil. Em seguida, a seção 4 apresenta algumas sugestões para caminhar em direção à solução do desafio, o que é acompanhado, na seção 5, pelas considerações finais.

2. Desenvolvimento em Contextos Acadêmicos

Software educacionais podem trazer diversos benefícios a professores e alunos. Porém, experiências do nosso grupo de pesquisa indicam que esses benefícios podem ser afetados pela qualidade do sistema, apesar de não encontrarmos literatura que afirme isso. Quando o sistema possui problemas em sua qualidade, a contribuição que proporciona aos usuários é comprometida. Alguns exemplos são um professor que relata um defeito no sistema que utiliza e os desenvolvedores levam muito tempo para consertá-lo, prejudicando o uso pelo professor, o que pode também ocorrer com pedidos de novas funcionalidades. Um defeito pode também atrapalhar a aprendizagem de um aluno, confundindo-o. Além disso, propriedades indesejadas, como erros de interface, de redirecionamento entre conteúdos e aplicações lentas desmotivam alunos.

Do ponto de vista técnico, a qualidade do software está relacionada com os métodos e técnicas utilizadas durante o seu processo de desenvolvimento (CMMI Product Team, 2002; Ben-Menachem & Marliss, 1997). No caso acadêmico isso é ainda mais importante pela volatilidade da equipe e pelo foco em pesquisa. Dessa forma, compete aos grupos de pesquisa que são desenvolvedores usarem bem esses métodos e técnicas com o objetivo de minimizar os efeitos indesejados durante o uso e manutenção dos sistemas. Além disso, os métodos e técnicas devem ser adaptados ao contexto particular, no caso o acadêmico. O desafio que queremos destacar aqui, portanto, é a *falta ou dificuldade de uso de métodos e técnicas de desenvolvimento de software para a criação de sistemas educacionais de qualidade em contexto acadêmico*.

Infelizmente, a literatura sobre desenvolvimento de software educacional trata alguns aspectos do processo, sem dar muita atenção à utilização de métodos e técnicas de desenvolvimento. A maior parte apresenta principalmente sistemas prontos, com sua arquitetura e funcionalidades principais (Choquet & Corbière, 2005; Kamiya & Brandão, 2009; e artigos consultados do SBIE descritos na seção 3.1). Outros trabalhos relatam a aplicação dos sistemas desenvolvidos com professores e alunos (Jones, 2010; Tanbellini & Brandão, 2010). Sobre métodos e técnicas de desenvolvimento, encontramos mais trabalhos propondo formas de realizar o projeto instrucional do software, onde é feito principalmente o trabalho interdisciplinar e são definidos, por exemplo, o uso, as funcionalidades, a interface e o conteúdo (Boyle, 2003; Mor & Winters, 2007).

Obviamente, o processo de desenvolvimento não é o foco da literatura de IE, como é o caso da Engenharia de Software (ES). Porém, tanto a literatura de IE quanto a de ES tem falta de trabalhos relacionados à avaliação de métodos e técnicas de desenvolvimento de software aplicados especificamente ao processo de criação de software educacionais em contexto acadêmico. Além disso, faltam trabalhos que analisem a relação entre a qualidade computacional dos sistemas e a experiência de uso por parte do usuário final, o que está relacionado à qualidade pedagógica. Este artigo tem como objetivo principal caracterizar o contexto acadêmico do desenvolvimento de software e relatar um panorama simplificado do uso de métodos e técnicas da área. Isso pode permitir a recomendação de um arcabouço de técnicas e processos para o desenvolvimento e manutenção de software educacional, visando facilitar a evolução desses sistemas no contexto acadêmico.

3. Ausência de Métodos e Técnicas de Desenvolvimento

Para analisarmos o uso de métodos e técnicas de desenvolvimento de software por grupos de pesquisa em universidades brasileiras e verificar se o desafio relacionado a esse uso realmente existe, conduzimos três estudos. Primeiro foi realizada uma leitura extensiva dos artigos do SBIE em busca de relatos desses usos. Em seguida, um questionário foi entregue a desenvolvedores com perguntas sobre os métodos e técnicas utilizadas em seus cotidianos. Por fim, entrevistamos desenvolvedores para analisar mais profundamente a situação em que o desenvolvimento de software ocorre. Todos os dados coletados estão disponíveis em planilhas e transcrições de entrevistas na página <http://line.ime.usp.br/artigos/desafie2012>.

3.1. Relatos em Artigos do SBIE

Esse estudo foi realizado com o objetivo de verificar a ocorrência de relatos de uso de técnicas e métodos de desenvolvimento em artigos publicados nas edições do SBIE entre 2009 e 2011 cuja contribuição fosse, de alguma maneira, caracterizada como software educacional. Essas informações podem ser usadas para a análise em conjunto com os resultados dos outros dois estudos, mas não podem ser consideradas isoladamente. O fato de um artigo relatar ou não o processo de desenvolvimento do software que apresenta nada diz sobre sua qualidade.

As informações relevantes relatadas nos artigos sobre o desenvolvimento dos software educacionais apresentados foram classificadas em quatro aspectos: (i) processo de implementação; (ii) processo de projeto instrucional; (iii) arquitetura; e (iv) tecnologia. Essa classificação é baseada na descrição de todas as etapas do desenvolvimento de um software educacional (Tchounikine, 2011; Mor & Winters, 2007). O processo de *implementação* envolve exatamente os métodos e técnicas de codificação e desenvolvimento de interesse do artigo, como métodos ágeis, desenvolvimento em espiral e testes de software. O processo de *projeto instrucional* contém informações sobre as etapas de definição das intervenções pedagógicas e o trabalho interdisciplinar. A *arquitetura* trata da estrutura e comunicação internas do software. E por *tecnologia* entende-se as linguagens de programação, *frameworks* e outras ferramentas utilizadas.

O método utilizado para a realização desse estudo consistiu nas seguintes etapas: (i) leitura do resumo de todos os artigos; (ii) busca no texto de cada artigo por uma seção ou parágrafo sobre um software educacional; (iii) seleção dos artigos que têm como contribuição um sistema; (iv) leitura da introdução e das seções relacionadas ao desenvolvimento e à apresentação da ferramenta dos artigos selecionados; (v) verificação de informações correspondentes aos aspectos de desenvolvimento escolhidos; e (vi) análise dos dados.

Foram analisados 268 artigos, dos quais 126 relatavam contribuição como software educacional. Na tabela 1 apresentamos a distribuição dos artigos analisados em relação aos quatro itens acima citados. Os resultados mostram que apenas 5,6% dos artigos que possuem uma contribuição na forma de software educacional apresentam informações sobre métodos e técnicas usadas no processo de implementação. Por outro lado, os aspectos de arquitetura e tecnologia são relatados com mais frequência, atingindo 42% e 60% respectivamente. O método de projeto educacional é apresentado em cerca de 28% dos artigos.

Tabela 1. Porcentagem dos artigos que apresentam informações relatando cada um dos aspectos do desenvolvimento

Aspectos	Implementação	Projeto Instrucional	Arquitetura	Tecnologia
%	5,6%	27,8%	42,1%	60,3%

Na tabela 2 apresentamos a distribuição de frequência quanto à quantidade de itens (dentre implementação, projeto instrucional, arquitetura e tecnologia) presentes nos artigos. Os resultados mostram que cerca de 25% dos artigos não relatam sequer um aspecto sobre o desenvolvimento do software apresentado, sendo que os demais relatam

ao menos um aspecto. 36% relatam dois aspectos e apenas 11% contêm informações sobre dois ou mais. Essas porcentagens são analisadas em conjunto com os dados obtidos pelos outros estudos para elaborar o desafio que propomos na seção 3.4.

Tabela 2. Proporção dos artigos que apresentam informações relatando quantidades de aspectos do desenvolvimento

No. de aspectos	0	1	2	>2	Total
%	25,4%	27,8%	35,7%	11,1%	100%

3.2. Questionário para desenvolvedores

Uma vez que a leitura dos artigos é abrangente mas carrega pouca informação sobre cada caso (cada processo de desenvolvimento), enquanto as entrevistas analisam com mais profundidade alguns poucos casos, o questionário se apresenta como um estudo intermediário. O objetivo do questionário entregue a desenvolvedores de diversos projetos de pesquisa foi levantar as características de suas tarefas no desenvolvimento dos sistemas, com foco no uso de métodos e técnicas para melhorar esse processo.

O método utilizado consistiu primeiramente na elaboração do questionário e em seguida na entrega dele aos desenvolvedores. Isso ocorreu no Workshop APPLETS, um evento paralelo ao SBIE 2011, ocorrido em Aracaju (SE). Esse *workshop* reuniu pesquisadores e desenvolvedores de software educacionais com o objetivo de discutir os desafios enfrentados por eles, visando uma maior colaboração entre os grupos.

O questionário foi dividido em cinco partes: (i) identificação do sistema; (ii) abordagem; (iii) projeto educacional; (iv) desenvolvimento; e (v) qualidade. A identificação e a abordagem do sistema caracterizam o grupo de pesquisa e os objetivos do software. As questões sobre projeto educacional tratam da etapa anterior ao desenvolvimento, da qual muitas vezes o programador também participa, enquanto que as questões sobre o desenvolvimento em si e sobre qualidade referem-se estritamente ao trabalho do programador.

Um total de 23 pessoas responderam o questionário, envolvendo 22 diferentes projetos de pesquisa. Dos respondentes, cerca de 70% eram coordenadores do projeto e 30% programadores. Dos projetos, 64% tinham menos de dois anos de execução, 22% tinham entre 2 e 4 anos e 14% tinham 5 ou mais anos.

Essas duas partes do questionário contêm três questões em comum, sobre a quantidade de pessoas para a execução da tarefa, qual é o tempo de experiência dessas pessoas e qual o método utilizado. As respostas das duas primeiras questões são numéricas, enquanto que a da última era discursiva. A figura 1 sintetiza as respostas sobre o projeto educacional e a figura 2 sobre o desenvolvimento de código.

A partir das respostas agrupamos os métodos de desenvolvimento em três níveis, 1 a 4, por ordem crescente de fundamentação teórica (em Projeto Instrucional ou ES), sendo o nível 1 aquele com menor fundamentação teórica. No nível 1 estão aqueles trabalhos que usaram técnica *ad-hoc* para desenvolvimento, sem empregar técnicas explícitas. No nível 4 estão os trabalhos que ancoraram fortemente o desenvolvimento do software a técnicas de ES.



Figura 1. Respostas ao questionário relacionadas ao projeto educacional dos software educacionais desenvolvidos

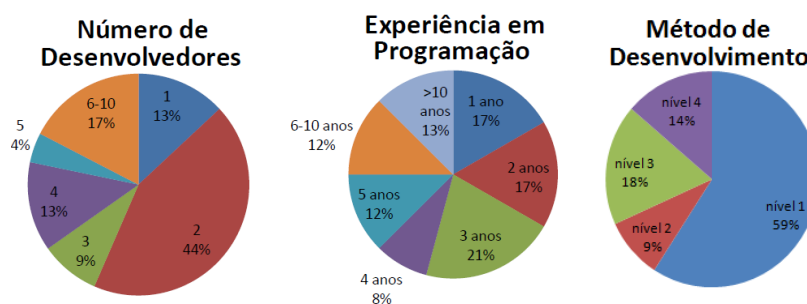


Figura 2. Respostas ao questionário relacionadas ao desenvolvimento de código dos software educacionais desenvolvidos

Uma análise preliminar das respostas indica que o número de projetistas educacionais varia mais que o de programadores, sendo que este último é menor, com 57% dos projetos tendo no máximo dois programadores. O número maior de projetistas pode ser justificado pelo fato dos programadores serem também projetistas, mas nem todos projetistas programam. Em geral a experiência dos programadores é maior do que a dos projetistas, mesmo assim 55% dos programadores tem três anos ou menos de experiência, o que equivale ao fim de um curso de bacharelado em Ciência da Computação. Por fim, notamos que 63% dos projetistas educacionais e 68% dos programadores usam método em nível 1 ou 2 e somente em 14% nos dois casos os grupos relatam usar um método de trabalho nível 4.

Além desses dados, o questionário abordou questões como tipo de licença de distribuição (sendo 77% dos sistemas são de código aberto), abordagem pedagógica, plataforma, áreas de especialização dos projetistas educacionais, linguagem de programação usada e problemas com relação a falta de recursos financeiros e de tempo.

3.3. Entrevistas com desenvolvedores

Foram realizadas entrevistas com cinco programadores de diferentes grupos de pesquisa que participaram do desenvolvimento de algum software educacional. As entrevistas objetivavam identificar o uso de métodos e técnicas de implementação e os problemas enfrentados pelos programadores nos processos de manutenção e evolução de software. Dessa forma, pudemos comparar os métodos e técnicas relatados com os respondidos no questionário para transferir possivelmente os problemas para outros casos.

O método para esse estudo foi o seguinte. Primeiro definimos as informações a serem coletadas, principalmente a definição do contexto de desenvolvimento, o método

usado e se havia dificuldades de programação provocadas por código de baixa qualidade. Também foi escolhido o perfil do entrevistado: programador em um projeto de pesquisa de desenvolvimento de software educacional em uma universidade. De preferência, o trabalho do programador deveria ser em um código criado por outra pessoa, para que os aspectos de manutenção fossem mais pronunciados. Para a escolha dos entrevistados, selecionamos grupos de pesquisa em IE de diferentes universidades, engajados em projetos que não participaram do Workshop APPLETS e, pela leitura de artigos publicados no SBIE, selecionamos os autores-programadores. Entramos em contato com dez pessoas e entrevistamos cinco.

As perguntas da entrevista continham grande parte daquelas presentes no questionário, mas apresentavam outras mais específicas, que solicitavam ao entrevistado que definisse melhor seu modo de trabalho e características do software que desenvolvia. Exemplos de questões desse tipo são: “Qual era o estado do projeto quando você entrou? Qual o número de usuários? Qual o tamanho do código?” e “Qual é a sua tarefa nesse projeto? Explique em detalhes.”. A seguir, na tabela 3, descrevemos cada caso:

Tabela 3. Descrição resumida dos casos relatados nas entrevistas

	Caso 1	Caso 2	Caso 3	Caso 4	Caso 5
Formação	Fez Ciência da Computação e agora está no mestrado em CC, mas em outra área	Aluno de Sistemas da Informação	Aluno de Engenharia de Computação	Técnico em Informática e agora faz Sistemas da Informação	Tecnólogo em Proc. de Dados, MBA em Gestão de Projetos, certificado Java
Experiência	IC	Bem pouca, disciplinas	Disciplinas durante 3 anos, fez IC antes em ES	Programou no ensino médio técnico	3 anos como programador em 3 empresas
Situação	TCC, sozinho	IC, 3º e 4º anos da graduação	TCC usando arcabouço criado numa disciplina	Bolsista com mais 1 sendo coordenado por pós-graduandos	Bolsista de projeto de pesquisa, para depois entrar no mestrado
Duração	9 meses	8 meses	1 ano e meio	6 meses	6 meses
Projeto	Expandir um jogo feito num outro TCC	Terminar um ambiente de testes, feito anteriormente por três pessoas, depois expandir	Criar um jogo educacional	Expandir um ambiente de simulação feito por dois pós-graduandos anteriores	Reestruturar um sistema feito por vários programadores
Dificuldades	Linguagem nova, modo de programar do outro programador, pouca documentação	Não acostumado com projetos grandes, código pouco legível, difícil integração com outras ferramentas, sem documentação	Projeto longo, trabalhou sozinho, pouca documentação, mas como fez do início teve pouco problema	O código tava tão ruim, duplicado e com documentação confusa, que refez totalmente em outra linguagem, inclusive	Código muito ruim, teve que preparar ambiente, muito trabalho para reestruturar o modelo de maneira mais desacoplada

Os resultados obtidos com as entrevistas complementam os estudos citados anteriormente. Todos os entrevistados tinham entre 22 e 25 anos e realizaram os projetos como programadores de software educacionais em uma de quatro categorias: em projetos de iniciação científica (IC), em trabalhos de conclusão de curso (TCC), em projetos de mestrado ou como bolsistas de projetos de pesquisa. Apenas no Caso 1 o programador possuía uma atividade profissional em paralelo.

A escolha dos entrevistados seguiu um critério em função da disponibilidade dos potenciais entrevistados, o que pode ter provocado um efeito de “amostra viciada”, pois os casos revelam resultados parecidos. A representatividade dos casos relatados nas entrevistas pode ser inferida a partir da comparação entre os métodos de implementação com os respondidos nos questionários, analisados a seguir.

3.4. Análise dos Dados

A partir da análise dos três estudos pode-se concluir que, no ambiente de pesquisa em IE, o uso de técnicas sistemáticas de desenvolvimento de software tem baixa adesão. Apenas 6% dos artigos publicados no SBIE dos últimos três apresentam informações sobre o processo de desenvolvimento, 68% dos respondentes do questionário afirmam que usam métodos de nível 1 ou 2 e as entrevistas realizadas mostram diversas dificuldades associadas a qualidade de software enfrentadas por programadores que entram em projetos em andamento. Como a maior parte dos métodos de implementação respondidos no questionário são similares aos relatados nas entrevistas, é possível que esses projetos também enfrentem, atualmente ou no futuro, problemas de manutenção e evolução de software.

O cruzamento dos dados obtidos nos três estudos permite inferir que existe uma *falta de uso de métodos e técnicas de desenvolvimento de software para a criação de sistemas educacionais de qualidade em contexto acadêmico*, sendo esse o desafio apresentado por este artigo. As respostas ao questionário refletem os resultados da análise dos artigos publicados no SBIE. Os artigos geralmente não apresentam informações sobre o método usado no desenvolvimento por esse aspecto não ter sido considerado explicitamente, entre outros motivos. As entrevistas relatam experiências de dificuldades de programadores que poderiam ser evitadas pelo uso de métodos e técnicas sistemáticas de desenvolvimento pelos programadores anteriores do projeto.

4. Uso de Métodos e Técnicas em Contextos Acadêmicos

As causas da dificuldade de uso ou da não adoção de métodos de desenvolvimento em projetos de IE não são claras, exigindo um estudo investigatório específico. Citamos algumas possibilidades. A volatilidade da equipe de desenvolvedores, formada por alunos de IC, TCC e pós-graduação que deixam o projeto ao fim do programa é um exemplo. Outra possível causa é o foco em resultados rápidos em desenvolvimento e aplicação, o que deixa poucos recursos para a adoção de técnicas sistemáticas. Porém, o uso de técnicas *ad-hoc* de desenvolvimento agrava os problemas provocados pela volatilidade e ocasiona dificuldades de manutenção e atualização dos sistemas pelos programadores, como mostram as entrevistas realizadas.

Apesar de o processo de desenvolvimento de software não ser o foco da IE, essa comunidade desenvolve software e por isso deveria usar resultados das pesquisas da área de ES adaptados ao seu contexto para melhorar seu trabalho. Existem diversos métodos e técnicas que poderiam ser aplicados na criação de software educacionais, mas geralmente as condições enfrentadas pelos pesquisadores da área no Brasil limita essa diversidade. Além disso, é necessário adaptar para o contexto acadêmico o que foi criado tendo em vista empresas, o que é comum na literatura em ES.

Escolher o método ou técnica e adaptá-los para a situação específica de cada projeto de pesquisa pode ser difícil. Isso inclui dificuldades relacionadas a falta de tempo e recursos, gerenciamento, formação e experiência dos programadores, tamanho da equipe, entre outras. Assim, a escolha e a adaptação devem considerar essas dificuldades a priori. Muitas vezes a Engenharia de Software traz ferramentas complexas que exigem uma grande equipe ou programadores experientes ou eventualmente pessoas que tratam especificamente de documentação ou burocracia, o que geralmente não é adaptado a contextos acadêmicos.

Para essas tarefas, porém, há pouca literatura. Dessa forma, outra proposta que apresentamos é o relato de iniciativas desse tipo em artigos de maneira a destacar as informações e decisões mais importantes em cada caso. Nesse caso, espera-se que relatos dessas iniciativas incluam avaliações das características das técnicas e métodos escolhidos, adaptados e utilizados no desenvolvimento de software educacionais e a explicitação de lições aprendidas em função das consequências de cada resultado.

5. Considerações Finais

O principal desafio aqui apresentado refere-se à introdução de métodos e técnicas de desenvolvimento de software visando a criação de sistemas educacionais de qualidade em contexto acadêmico. Uma consequência da falta do uso de técnicas significativas para assegurar a qualidade computacional dos software educacionais acarretam problemas de manutenção e continuidade dos projetos, que em última instância acaba limitando seu uso em larga escala, mesmo que sua qualidade pedagógica seja alta. Um estudo para avaliar a relação entre as qualidades computacional e pedagógica de um software educacional é sugerido como trabalho futuro.

Esse desafio, assim como as consequências para o trabalho dos desenvolvedores e pesquisadores, foi apresentado com base em resultados de uma análise dos artigos do SBIE, um questionário e entrevistas realizadas com vários grupos de IE no Brasil. Nossa proposta é que os grupos de pesquisa passem a considerar com mais frequência o uso de técnicas e métodos de ES para apoiar o desenvolvimento de software educacionais. Além disso, propomos a adaptação destas técnicas e métodos para o contexto acadêmico específico de cada projeto de pesquisa e o relato de avaliações desses trabalhos para constituir uma literatura do assunto.

Outra maneira de enfrentar esse desafio seria agrupar por interesse grupos de pesquisa, de modo a colaborarem compartilhando código, uma vez que para efetivar esse compartilhamento é necessário o estabelecimento de técnicas. Outra consequência da colaboração é a divisão de tarefas, que poderia ser inviável individualmente.

A mudança na forma de trabalho de um grupo de pesquisa é uma tarefa complexa. A proposta deste artigo exige um esforço inicial considerável frente às dificuldades cotidianas enfrentadas pelos pesquisadores. Porém, acreditamos que ganhos de eficiência e redução de problemas sejam compensadores. Esforços na formação de programadores ingressantes em projetos de pesquisa em técnicas e métodos de desenvolvimento de software podem ocupar algumas semanas mas economizar muitas horas de correção de defeitos e código emaranhado.

Agradecimentos

Os autores agradecem a todos os respondentes do questionário, no Workshop APPLETS e aos programadores entrevistados. Este trabalho teve apoio parcial de FAPESP (2011/10926-2) e CNPq (550449/2011-6). O autor Danilo L. Dalmon é financiado pela FAPESP (2010/06805-2).

Referências

- Ben-Menachem, M. & Marliss, G. S. (1997) "Software Quality, Producing Practical and Consistent Software". Thomson Computer Press.
- Boyle, T., (2003) "Design principles for authoring dynamic, reusable learning objects", Australian Journal of Educational Technology, vol.19, pp.46-58.
- Bridge, P. & Appleyard, R., (2008) "A comparison of electronic and paperbased assignment submission and feedback", British Journal of Educational Technology, vol.39, n.4, pp.644-650.
- Choquet, C. & Corbière, A. (2005) "Reengineering framework for systems in education". Educational Technology and Society, vol.9, pp.228-241.
- CMMI Product Team (2002). "Capability Maturity Model R Integration (CMMI), Version 1.1". Pittsburgh PA: SEI CMU.
- Kamiya, R. H. & Brandão, L. O. (2009) "iVProg - um sistema para introdução à programação através de um modelo visual na Internet". Anais do XX Simpósio Brasileiro de Informática na Educação.
- Koedinger, K. R., & Alevan, V. (2007) "Exploring the Assistance Dilemma in Experiments with Cognitive Tutors", Education Psychological Review, pp.239-264.
- Jones, M., (2010) "A CSCL Approach to Blended Learning in the Integration of Technology in Teaching", Interdisciplinary Journal of E-Learning and Learning Objects, vol.6, pp.103-113.
- Marrades, R., & Gutiérrez, A. (2001) "Proofs Produced by Secondary School Students Learning Geometry in a Dynamic Computer Environment", Educational Studies in Mathematics, pp.87-125.
- Mor, Y. & Winters, N. (2007) "Design approaches in technology-enhanced learning", Interactive Learning Environments, vol.15, pp.61-75.
- Richards, M. H., McGreal R., & Friesen, N., (2002) "The evolution of learning object repository technologies: Portals for on-line objects for learning", Journal of Distance Education, vol.17, pp.67-79.
- Tanbellini, M. J. G. S., Brandão, L. O. (2010) "O uso da Geometria Interativa como Facilitador no Ensino e Aprendizagem de Matemática", Anais do XVI Workshop de Informática na Educação, 2010.
- Tchounikine, P., (2011) "Computer Science and Educational Software Design", DOI 10.1007/978-3-642-20003-8_1, Springer-Verlag Berlin Heidelberg.
- van Dam, A. & Simpson, R. M. (2005) "Next-generation education software - why we need it and a research agenda for getting it". Educause Review, pp.26-43.