

FRAMES: Uma Proposta de *Framework* para o Ensino de Tópicos da Engenharia de Software

Carlos S. Portela¹, Alexandre M. L. Vasconcelos¹, Sandro R. B. Oliveira¹

¹Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – 50.740-560 – Recife – PE – Brasil
{csp3, amlv, srbo}@cin.ufpe.br

***Abstract.** The software industry is dissatisfied about the preparation level of recently graduated professionals that work in the Software Engineering (SE) area. Thus, companies invest in professional training for the teaching of specific practices. Based on industry training strategies and teaching methods focused on student, this paper presents a framework proposal for the SE teaching/learning. We expect that the use of this framework can help to reduce the shortage in training professionals and consequently meet the software industry's demand.*

***Resumo.** A indústria de software apresenta insatisfação quanto ao nível de preparação dos profissionais recém-formados que atuam na área de Engenharia de Software (ES). Assim, as empresas investem na capacitação profissional para repasse de práticas específicas da ES. Baseado em estratégias de capacitação da indústria e em métodos de ensino focados no aluno, este artigo apresenta uma proposta de framework para o ensino/aprendizado de ES. Espera-se que o uso deste framework possa ajudar a diminuir a carência na formação de profissionais e, conseqüentemente, atender a demanda da indústria de software.*

1. Introdução

A indústria de software se queixa de que os cursos de graduação não ensinam aos estudantes as competências necessárias para que eles possam começar a executar o seu trabalho com eficiência [Meira, 2015]. Desta forma, as empresas de software investem na capacitação dos seus profissionais para repasse técnico de práticas específicas que compreendem áreas da Engenharia de Software (ES). Esta carência na formação de profissionais graduados na área de ES pode ser resultado de uma educação inadequada [Lethbridge *et al.*, 2007].

Primeiramente, é inviável cobrir todos os tópicos sugeridos pelos currículos de referência da ACM/IEEE (2013) e da SBC (2005), especialmente considerando a carga horária disponível para a disciplina de ES [Wangenheim e Silva, 2009]. Em segundo lugar, as abordagens de ensino propostas não adotam estratégias que alterem este atual cenário do meio acadêmico [Meira, 2015]. Por fim, Wangenheim e Silva (2009) destacam diversas limitações quanto o desenvolvimento de competências profissionais dos estudantes no ambiente acadêmico. Estas limitações acabam por agravar a carência de profissionais habilitados na área de ES [Lethbridge *et al.*, 2007].

A fim de apoiar o fechamento destas lacunas, este artigo apresenta uma proposta de *framework* denominado FRAMES. O termo *framework* adotado neste trabalho define uma estrutura conceitual básica de itens de ensino/aprendizagem aplicada no domínio de Engenharia de Software, sendo adaptável e reutilizável no contexto de disciplinas da graduação em Ciência da Computação, fornecendo apoio a professores e alunos.

Além desta seção introdutória, a Seção 2 apresenta os trabalhos relacionados a esta proposta de *framework*. Os módulos e atividades do FRAMES são descritos na Seção 3. Por fim, as limitações, os resultados esperados a partir do uso deste em disciplinas de ES e as próximas etapas deste trabalho são apresentadas na Seção 4.

2. Trabalhos Relacionados

Hazzan e Dubinsky (2006) definiram um *framework* para o ensino de Métodos de Desenvolvimento de Software (MDS). Este *framework* define princípios conceituais e especifica práticas referentes às atividades de ensino que orientam os professores no processo de desenvolvimento da disciplina de ES e na avaliação dos estudantes.

A primeira prática é “Curso” que apresenta uma estrutura detalhada do curso, a segunda é “Papel” que apresenta um esquema detalhado do perfil dos envolvidos. A terceira prática é “Medição” que apresenta medidas que ajudam a avaliar o processo de desenvolvimento. A quarta prática é “*Coaching*” e a quinta prática é “Avaliação”.

O *framework* proposto por Hazzan e Dubinsky (2006) é bem compreensível e detalha como implementá-lo em cursos de graduação. No entanto seu foco é no ensino de MDS. O *framework* proposto nesta pesquisa se baseará na sua estrutura conceitual, definindo princípios e práticas. Como diferencial, poderá ser implementado no ensino de qualquer tópico das unidades de conhecimento da ES.

Outro *framework* foi proposto na área de ES por Sowe, Stamelos e Deligiannis (2006), que usa a metodologia de desenvolvimento de software *open source* e foi implementado na disciplina de “Introdução à Engenharia de Software” no departamento de Informática da *Aristotle University* na Grécia.

Este *framework* possui 3 fases, onde na Fase 1, os estudantes realizam 8 horas de leitura sobre os tópicos de desenvolvimento e testes. Na Fase 2, os estudantes aprendem a registrar os seus projetos em ferramentas de rastreamento de *bugs* e iniciam os testes nos softwares livres do *sourceforge.net*. Por fim, na Fase 3, baseado na apresentação dos projetos e nas atividades de teste, os estudantes são avaliados.

A principal contribuição desta proposta para o trabalho apresentado é a estruturação do *framework* em fases bem distintas. No entanto, esta proposta foca apenas na área de testes. O *framework* proposto neste artigo busca ser estruturado de forma a ser aplicado a qualquer unidade de conhecimento da ES. Além disso, incorporará práticas de capacitação da indústria no módulo do projeto prático.

3. Proposta de *Framework* para o Ensino

3.1. Princípios do *Framework*

A proposta do FRAMES se baseia em 4 (quatro) princípios derivados de trabalhos relacionados que abordam o uso de métodos focados no aluno no ensino de ES:

- I. O estudante deve ser o foco do processo de aprendizagem;
- II. A aprendizagem deve ser baseada na resolução de problemas;
- III. A realização de projetos práticos desenvolve competências e habilidades;
- IV. O modelo pedagógico deve seguir uma abordagem iterativa.

A partir destes princípios, definiram-se os módulos do FRAMES.

3.2. Módulos do *Framework*

Estruturalmente, o FRAMES possui 4 (quatro) módulos, conforme apresenta a Figura 1.



Figura 1. Módulos do *Framework* FRAMES

O módulo Plano de Ensino incorpora os tópicos das seguintes unidades de conhecimento: I) Engenharia de Requisitos; II) Processos de Software; III) Gerenciamento de Projetos de Software; IV) Projetos de Software; V) Verificação e Validação de Software; VI) Ferramentas e Ambientes. Estes tópicos foram identificados a partir de um *survey* realizado com professores sobre quais tópicos da ACM/IEEE (2013) e SBC (2005) são adotados na disciplina de ES [Portela, Vasconcelos e Oliveira, 2015]. Desta forma, pretende-se limitar a quantidade de tópicos ministrados, a fim de que estes possam ser adequar a carga horária disponível para a disciplina de ES.

O módulo Modelo Pedagógico foi definido a partir de métodos de ensino focados nos alunos, abordagens iterativas de ensino, práticas de capacitação da indústria e estratégias de avaliação. Este modelo pedagógico, representado na Figura 2, combina aprendizagem centrada em problema (PBL) [Meireles e Bonifácio, 2015], atividades em grupo como discussões e dinâmicas, reflexão e projeto prático.



Figura 2. Modelo Pedagógico do FRAMES

As práticas de capacitação identificadas junto a consultores da indústria e adaptadas para a academia foram *mentoring*, onde o professor compartilha seus conhecimentos com os alunos e os orienta na condução do projeto, e *coaching*, onde alunos veteranos realizam treinamentos em tópicos específicos de ES.

Já o módulo Projeto Prático sugere o uso de clientes/problemas reais, definição de prazos (cronograma), definição de papéis e responsabilidades pelos alunos. Para tal, disponibiliza um modelo de processo (<http://goo.gl/rKmctX>) que define atividades, papéis e artefatos de acordo com as 6 unidades de conhecimento do Plano de Ensino. Por fim, no módulo Avaliação de Aprendizagem, sugerem-se estratégias de avaliação que consistem na entrega de produtos de trabalho, participação individual dos alunos, questionários, dicionários de dados e mapas conceituais [Miranda *et al.*, 2015].

3.3. Instanciação e Uso

O *framework* FRAMES estará disponível em uma plataforma de *wiki*. Assim, sua instanciação deverá ocorrer em três etapas. Na *Etapa 1 - Configuração do Ambiente*, inicialmente o professor cria sua própria versão da *wiki*, instanciando-a para sua disciplina. Em seguida, o professor seleciona a unidade de conhecimento que deseja ministrar. Assim, deve instanciar os tópicos do módulo Plano de Ensino de acordo com a unidade de conhecimento selecionada.

Na *Etapa 2 - Aplicação do Framework*, o professor deverá fazer uso do módulo Modelo Pedagógico, onde terá acesso a uma base de vídeo-aulas, propostas de *workshops* e dinâmicas, bem como um formulário para registro das reflexões. No módulo Projeto Prático, os alunos terão acesso ao modelo de processo, para que possam instanciá-lo em seu projeto. Por fim, na *Etapa 3 - Aplicação do Framework*, o professor poderá instanciar as estratégias de avaliação do módulo Avaliação de Aprendizagem, onde terá acesso a *templates* e exemplos de itens de avaliação.

4. Considerações Finais

A proposta de adotar um *framework* permite lecionar uma determinada unidade de conhecimento da ES, a fim de melhor utilizar a carga horária da disciplina. No entanto, tempo continua sendo um fator crítico. A partir de um teste de aplicação, observou-se que se leva, em média, 3 aulas para lecionar os tópicos de uma unidade. Outra limitação do FRAMES consiste na falta de experiência de professores que não atuaram na indústria que podem vim a fazer uso deste *framework*. Para esta limitação, sugere-se a participação de um profissional da indústria para atuar na etapa de Discussão (Figura 2).

Apesar destas limitações, espera-se que a disponibilização do FRAMES em uma *wiki* permita aos professores instanciarem métodos de ensino que melhor desenvolvam competências profissionais nos alunos. Além disto, estes poderão contribuir com a sua evolução, através do envio de vídeos, artigos, além de sugestões de melhoria. Os alunos poderão usar esta *wiki* para registrarem seu aprendizado em cada uma das unidades de conhecimento, instanciarem o modelo de processo no projeto prático e estudarem fora do ambiente de sala de aula, através dos vídeos e outras ferramentas disponíveis.

A etapa atual desta pesquisa consiste na finalização da *wiki* do *framework*. Paralelamente, está sendo definido o protocolo do estudo de caso para aplicação deste

framework em 3 disciplinas da área de ES de instituições de ensino diferentes. Sendo assim, a próxima etapa desta pesquisa consiste na validação do uso do FRAMES.

Agradecimentos

Os autores gostariam de agradecer à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro ao desenvolvimento desta pesquisa.

Referências

- ACM/IEEE (2013) “Computer science curricula 2013 – Curriculum guidelines for undergraduate degree programs in Computer Science”, <https://www.acm.org>, March.
- Hazzan, O. e Dubinsky, Y. (2006). A Framework for Teaching Software Development Methods. *Proceedings of the 28th International Conference on Software Engineering*, pages 703-706. Shanghai, China.
- Lethbridge, T. et al. (2007). Improving software practice through education: Challenges and future trends. *Proceedings of the Conference Future of Software Engineering*, pages 12-28. Minneapolis, EUA.
- Meira, S. Sistemas de Informação e Engenharia de Software – Cadê as Escolas? Revista da SBC Engenharia de Software - Qual é o impacto da ES no mercado de Computação e na sociedade como um todo? 1ª. ed. Cap. 1, páginas 11-15. Porto Alegre, Brasil.
- Meireles, M. e Bonifácio, B. (2015). Uso de Métodos Ágeis e Aprendizagem Baseada em Problema no Ensino de Engenharia de Software: Um Relato de Experiência. Em *Anais do XXVI Simpósio Brasileiro de Informática na Educação*, páginas 180-188. Maceió, Brasil.
- Miranda, S. et al. (2015). Experimento de uma ferramenta de fórum baseado em mapas conceituais: uma nova forma de interação. Em *Anais do XXVI Simpósio Brasileiro de Informática na Educação*, páginas 917-926. Maceió, Brasil.
- Portela, C., Vasconcelos, A. e Oliveira, S. (2015). Análise da Relevância dos Tópicos e da Efetividade das Abordagens para o Ensino de Engenharia de Software: Resultados de um Survey com Professores e Alunos. Em *Anais do VIII Fórum de Educação em Engenharia de Software*, páginas 24-35. Belo Horizonte, Brasil.
- Prikladnicki, R. et al. (2009). Ensino de Engenharia de Software: Desafios, Estratégias de Ensino e Lições Aprendidas. Em *Anais do II Fórum de Educação em Engenharia de Software*. Fortaleza, Brasil.
- SBC (2005). Currículo de Referência da SBC para Cursos de Graduação em Bacharelado em Ciência da Computação e Engenharia de Computação, <http://www.sbc.org.br>, Abril.
- Sowe, S., Stamelos, I. e Deligiannis, I. (2006). A Framework for Teaching Software Testing using F/OSS Methodology. *Open Source Systems*. Chap. 6, pages 261-266.
- Wangenheim, C. e Silva, D. (2009). Qual Conhecimento de Engenharia de Software é Importante para um Profissional de Software? Em *Anais do II Fórum de Educação em Engenharia de Software*. Fortaleza, Brasil.