

Diretrizes para o Ensino Interdisciplinar de Engenharia de Software

Juliana Cristina Braga

Faculdade de Computação e Informática (FCI) – Universidade Presbiteriana Mackenzie
(UPM) – São Paulo – SP.
{j.cbraga}@mackenzie.com.br

Abstract. *This paper proposes guidelines for practicing the concepts of Software Engineering in several disciplines within the curriculum of undergraduate courses in Computer Science. These guidelines contain how practice will be, what its goals and knowledge necessary to perform it. This knowledge will be represented by video, text, wiki, and organized in a collaborative portal on the Internet. The methodology is based on management's knowledge and can be applied in other areas.*

Resumo. *O trabalho propõe diretrizes para aplicação dos conceitos de Engenharia de Software em várias disciplinas pertencentes ao currículo da maioria dos cursos de graduação em Ciência da Computação. Essas diretrizes contêm como essa aplicação deverá ser feita, qual o seu objetivo e o conhecimento necessário para realização da mesma. Este conhecimento será representado em vídeos, textos, wiki, e organizado em um portal colaborativo na Internet. A metodologia utilizada é baseada na gestão do conhecimento e poderá ser aplicada em outras áreas.*

1. Introdução

A Engenharia de Software (ES) é uma área da Ciência da Computação que contém conhecimento suficiente para auxiliar na criação de diversos tipos de sistemas, sejam eles simples ou complexos. No entanto, os desafios nessa área continuam, pois ainda existem projetos que atrasam, ultrapassam o orçamento, não produzem software de qualidade e nem atendem às necessidades do cliente. Segundo Lethbridge (2007) parte desses problemas de orçamentos, prazos e qualidade são causados por carência de profissionais habilitados nessa área. Essa carência, por sua vez, são resultados de uma educação inadequada, podendo ser comprovada pela pesquisa feita por Sargent (2004) e citada pelo mesmo autor. A pesquisa revela que: (i) apenas 40% dos profissionais da área de informática dos Estados Unidos possuem formação nessa área; (ii) 40% desses conhecem os principais campos da ES, tais como: requisitos, arquitetura, testes, fatores humanos e gestão de projetos; (iii) a maior parte é qualificada em programação e/ou são especialistas em determinados tipos de produtos como ferramentas para gerenciamento de dados e desenvolvimento web, etc; e (iv) existe uma parcela considerável de profissionais que não tem consciência do que não sabem. Apesar de não terem sido encontrados dados estatísticos a respeito no Brasil, acredita-se que a realidade dos profissionais de ES neste país não deve ser diferente. No meio acadêmico, dificuldades no ensino de ES já foram relatadas (Soares, 2000; Castro, 2000; e Hazzan, 2003) (i) muito conteúdo sendo dado em pouco tempo; (ii) baixa motivação que os alunos de ciência da computação possuem para estudar os conceitos teóricos de ES; (iii)

dificuldades em preparar os estudantes para a prática profissional dentro de ambientes acadêmicos. Esforços estão sendo realizados para diminuir essas dificuldades e esse trabalho encaixa-se nesse contexto.

2. Objetivos

Em vista da importância que o ensino da Engenharia de Software possui e das suas dificuldades, o presente trabalho tem como objetivo geral aumentar a prática da ES dentro das disciplinas oferecida nos cursos de Ciência da Computação e relacionados. Com isso, pretende-se consolidar os conceitos da ES, conscientizar os alunos sobre a sua importância e incentivar sua prática desde o início da graduação até o mercado de trabalho. Os objetivos específicos são: (i) Elaborar diretrizes para aplicação da ES em outras disciplinas; (ii) experimentar aplicar essas diretrizes em uma Universidade; e (iii) avaliar e divulgar os resultados e lições aprendidas desse experimento.

O artigo está organizado como segue: na Seção 3 é apresentada a metodologia utilizada. Na Seção 4 são mostrados os resultados parciais e esperados. Na Seção 5 descrevem-se os trabalhos relacionados e na Sessão 6 as considerações finais.

3. Metodologia

Conforme mencionado no objetivo, esse projeto deverá relacionar a Engenharia de Software (ES) com outras disciplinas. Segundo Piaget (1973), as relações entre as disciplinas podem ocorrer em três níveis: multidisciplinaridade, interdisciplinaridade e transdisciplinaridade. Neste trabalho será abordada a relação de interdisciplinaridade, pois uma interação entre duas ou mais disciplinas deverá ser estabelecida. A metodologia que será utilizada para estabelecer a relação da ES com outras disciplinas será baseada nos processos já consagrados de gestão do conhecimento presentes nos modelos de Liebowitz e Beckman (1998). A escolha da gestão do conhecimento é justificada pelo fato de se considerar as diretrizes propostas equivalentes a conhecimentos que deverão ser gerenciados, ou seja, criados, organizados, armazenados, atualizados e recuperados. Cada processo é representado por uma etapa da metodologia proposta, totalizando assim 7 etapas. Dentro de cada uma dessas etapas outros aspectos também serão considerados, como por exemplo, o gerenciamento de projetos, técnicas para elaboração e validação de conteúdos educacionais, etc. Esses aspectos serão estudados a medida que o projeto for sendo desenvolvido. A definição final da metodologia fará parte dos resultados desse trabalho. A seguir uma breve descrição das tarefas contidas em cada uma dessas 7 etapas:

Etapa 1 – Identificação, conceituação do conhecimento, definição do capital intelectual: Nesta etapa ocorre a identificação e conceituação do conhecimento necessário para aplicar a ES em outras disciplinas. Para isso devem-se descrever as diretrizes de como a ES será aplicada a cada uma das disciplinas, o objetivo pedagógico que se deseja alcançar, a definição do conteúdo a ser disponibilizado contendo as referências a serem utilizadas para a elaboração do mesmo.

Etapa 2 – Captura do capital intelectual, captura do conhecimento, coleta do conhecimento, importação de metodologias e tecnologias externas: reunião do conhecimento necessário para atender o que foi definido na etapa 1. Esse conhecimento poderá: já existir dentro da instituição em que a metodologia está sendo implantado, estar disponível externamente à instituição e/ou não existir e exigir desenvolvimento utilizando como ponto de partida as referências da Etapa 1. Nessa etapa é necessária vasta revisão bibliográfica, comunicação interna e colaboração externa.

Etapa 3 – Seleção e validação do conhecimento: seleção e validação, padronização e adequação dos conhecimentos reunidos na Etapa 2.

Etapa 4 - Organização e armazenagem do conhecimento, codificação do conhecimento, compilação e transformação do conhecimento: organização e classificação do conhecimento selecionado e validado na Etapa 3. Para essa organização deve-se definir ainda uma metodologia.

Etapa 5 - Compartilhamento do capital intelectual, acesso e disponibilização do conhecimento, transferência do conhecimento, externalização do conhecimento: Disponibilização e divulgação do conhecimento organizado na Etapa 5 através de um portal colaborativo na Internet. A partir da divulgação do conhecimento, as diretrizes já poderão ser aplicadas.

Etapa 6 - criação do conhecimento, geração do conhecimento, busca de soluções criativas, combinações de conhecimentos: o portal ficará preparado para receber outras colaborações, e espera-se seja ampliado englobando outras disciplinas e outras áreas do conhecimento.

Etapa 7 - Avaliação dos benefícios e do valor do conhecimento: monitoramento permanente deverá existir com objetivo de avaliar os benefícios do projeto.

4. Resultados Parciais e Esperados

De acordo com a metodologia implantada, alguns resultados parciais foram obtidos. Em linhas gerais, na Etapa 1 foram definidas as diretrizes para aplicação da ES em outras disciplinas. Na Etapa 2 algum conhecimento já foi reunido e gerado para possibilitar a execução do que foi definido Etapa 1. Na Etapa 3, o conhecimento reunido está sendo selecionado e validado. Na Etapa 4, o conhecimento que foi revisado está sendo organizado. Na Etapa 5, o conhecimento reunido está sendo disponibilizado em um portal de testes denominado Portal Educacional de Engenharia de Software Aplicada (PEESA). Ainda não foram obtidos resultados referentes às Etapas 6 e 7. A seguir uma descrição mais detalhada dos resultados obtidos e esperados em cada um das etapas.

4.1 Resultados parciais da Etapa 1

Nesta etapa foram escolhidas quais disciplinas, inicialmente, deverão aplicar a ES em seu conteúdo. Para cada uma delas foram definidas as diretrizes contendo como, o objetivo e o conteúdo das aplicações. As disciplinas foram escolhidas tomando-se como base o currículo do curso de Ciência da Computação da Universidade Mackenzie que está alinhado com os currículos de referência da Sociedade Brasileira de Computação e ACM/IEEE. Deve-se observar que os objetivos pedagógicos definidos estão sempre buscando criar no aluno o perfil desejável para um bom Engenheiro de Software (Lucena, 2008 e Bagert, 1999) e o hábito de sempre gerar software de qualidade. A seguir uma descrição breve de parte do já foi definida nesta etapa:

- **Língua Portuguesa - Como:** deve-se solicitar que o aluno faça uma atividade que documente os requisitos de um sistema. O professor dessa disciplina deverá corrigir os erros de português do documento, as frases ambíguas e sem sentido e as inconsistências. *Objetivo pedagógico:* aumentar a capacidade de especificar software, escrever documentos técnicos com clareza, e destacar que a língua portuguesa também é muito importante para a área de Ciência da Computação. *Conteúdo:* documento explicativo sobre a atividade, o protótipo do software que será descrito e o modelo de documentação que deverá ser seguido. O aluno não

precisa entender ainda detalhes sobre o modelo, mas já ficará familiarizado com modelos de especificação do sistema.

- **Ética e cidadania I e II** - *Como*: mostrar e discutir o código de ética e práticas profissionais baseadas em oito princípios desenvolvidos pela *IEEE Computer Society and the ACM* (IEEE-CS,1999). Ler documentos relacionados ao mesmo assunto. *Objetivo pedagógico*: promover a conscientização da importância da compreensão da ética dentro da ciência da computação. *Conteúdo*: documento explicativo sobre a atividade, código de ética e o seguinte artigo sugerido para leitura: “How the New Software Engineering Code of Ethics Affects you.” (Gotterbarn,1999).
- **Administração**: *Como*: no tópico competência comunicativa verbal deve-se mostrar um vídeo aula sobre os casos práticos da dificuldade existente na etapa de elicitação de requisitos. *Objetivo pedagógico*: é demonstrar a real importância da comunicação para os alunos da Ciência da Computação, que em sua maioria possuem dificuldade nesse aspecto, aumento da capacidade de comunicação. *Conteúdo*: Para essa atividade estará disponível a vídeo aula.
- **Introdução a Algoritmos e Computação** - *Como*: deve-se solicitar que o aluno utilize, além de outras notações, o diagrama de atividades UML para descrever um algoritmo. O professor deverá cobrar dos alunos a utilização de um guia de boas práticas de programação (modelo para documentar e organizar seus programas). *Objetivo pedagógico*: criação do hábito e a prática, desde o início de sua formação, de projetar software, utilizar boas práticas de programação, como por exemplo, comentar o código, dar nomes mnemônicos para variáveis, etc. Pretende-se assim já iniciar uma cultura de qualidade de software e também introduzir a notação UML. *Conteúdo*: documento explicativo sobre a atividade, guia de melhores práticas de programação (McConnell, 2005), vídeo aula explicando o diagrama de atividades dentro do conceito abordado.
- **Linguagem de Programação I e Estruturas de Dados** - *Como*: deve-se, em sala de aula, apresentar os exemplos de classes e objetos utilizando o diagrama de classes e objetos UML. No caso de linguagem de programação I, sugere-se um projeto integrado com a disciplina de Desenvolvimento Orientado a Objetos que é oferecida no mesmo período. Deve-se dar continuidade ao uso do guia de melhores práticas de programação e iniciar a utilização de um controlador de versão para os programas a serem desenvolvidos. *Objetivo pedagógico*: aumento da capacidade de especificar software, criação do hábito e da prática em projetar software, criação do hábito da programação com qualidade, familiarização com UML, incentivo ao trabalho em equipe. *Conteúdo*: documento explicativo sobre a atividade, exemplos de classes em UML com código associado e descrição do trabalho de integração com todo material necessário para sua realização.
- **Computação Gráfica** - *Como*: devem-se utilizar as ferramentas testes unitários para testar os algoritmos de computação gráfica. Sugere-se dar continuidade ao uso do guia de melhores práticas e do controlador de versão. *Objetivo pedagógico*: Mostrar a utilidade das ferramentas de testes, despertando o interesse do aluno para essa sub área de ES que cresce a cada dia, criação do hábito da programação com qualidade. *Conteúdo*: documento explicativo sobre a atividade, atividade prática do uso de testes unitários em algoritmos de computação gráfica, utilização de módulos educacionais no domínio de testes (Barbosa, 2004).

- **Banco de Dados** - *Como*: exibir a modelagem do banco também utilizando diagramas de classes persistentes em UML. Solicitar entrega de um projeto de banco de dados contendo modelagem e dicionário de dados. O objetivo pedagógico é mostrar a importância da documentação dos dados criando esse hábito nos alunos e aplicar os conceitos de orientação a objetos, e da prática em projetar software. *Conteúdo*: Para essa atividade encontra-se uma sugestão de trabalho prático que exige esses conceitos e também modelos de diagrama de classe e dicionário de dados.
- **Iteração Humana Máquina**: *Como*: aplicar as atividades práticas propostas no trabalho: Uma Metodologia de Ensino para a Disciplina Interação Humano-Computador (Braga, 2005). *Objetivo pedagógico*: praticar as técnicas de prototipação com foco na usabilidade, validação dos requisitos e testes de avaliação da usabilidade. *Conteúdo*: Para essa atividade está disponível todo o material necessário para a aplicação da metodologia proposta como, por exemplo: protótipos, fichas de avaliações, etc.
- **Trabalho de Graduação I e II**: *Como*: cobrar dos alunos que todos os projetos ligados ao desenvolvimento de software, passem pelas etapas (cabíveis ao projeto) do processo de ES. Deve-se deixar clara a importância da utilização das ferramentas de versionamento principalmente de trabalhos em equipe. Deve-se cobrar do aluno que o código obedeça ao guia de melhores práticas. *Objetivo pedagógico*: praticar e incentivar a utilização da ES durante o trabalho de graduação, aumento da capacidade de especificar software, criação do hábito e da prática em projetar software, criação do hábito da programação com qualidade, enfatizar a importância das atividades de manutenção de software, incentivo ao trabalho em equipe, aumento da capacidade de comunicação, escrever documentos técnicos com clareza. Para essa atividade será disponibilizado um exemplo de trabalho de graduação que utilizou essa modelagem.
- **Programação de Jogos** - *Como*: solicitar que o aluno faça plano de testes, testes unitários, testes de integração, casos de teste e relatório de testes. *Objetivo pedagógico*: mostrar a importância da área de testes para o desenvolvimento de jogos. *Conteúdo*: Para essa atividade será disponibilizado o modelo da documentação e o tutorial de instalação de plugins de testes unitários.

Diretrizes para outras disciplinas, detalhes sobre como o professor deverá avaliar cada uma das atividades e as referências que serão utilizadas para geração do conhecimento foram omitidos por restrição de espaço. Até o momento, observou-se um grande interesse dos professores de outras disciplinas em utilizar a ES em sua disciplina. Além disso, os professores de áreas diferentes da computação consideram a abordagem desse projeto uma maneira de motivar os alunos de CC no estudo de sua disciplina.

4.2 Resultados Obtidos e Esperados da Etapa 2

Nesta etapa foram reunidos conhecimentos internos e identificados, através de revisão bibliográfica, conhecimentos externos que servirão de apoio para a execução das aplicações sugeridas na Etapa 1. Essa etapa é contínua e um levantamento bibliográfico deverá ser exaustivo para poder descobrir quais metodologias e ferramentas na área de Educação podem ser aproveitadas para o escopo desse projeto.

4.3 Resultados Obtidos Esperados da Etapa 3 e 4 e 5

O conhecimento reunido na etapa 1 será disponibilizado no portal do projeto. A organização desse portal será, inicialmente, baseada no *Guide to the Software Engineering* (Bagert, 1999) que foi construído baseado no SWEBOK (Abran, et al., 2001).

4.4 Resultados Esperados da Etapa 6

Deste ponto em diante espera-se que o projeto adquira vida própria, onde os professores poderão estar em constante uso e colaboração do conhecimento através do portal.

4.5 Resultados da Etapa 7

O portal deverá possuir ferramentas de controle de acesso para controlar as colaborações. Paralelo a isso, serão disponibilizados questionários para coletar “feedbacks” dos professores que aplicaram a ES em suas disciplinas. Uma análise dos dados será feita e os resultados publicados. Pretende-se ainda investigar melhor as práticas utilizadas para organizar documentos educacionais, como por exemplo: Mecanismos e abordagens de apoio à modelagem de conteúdos educacionais (Barbosa, 2004). Pretende-se investigar também a utilização de ontologias para apoiar a aquisição, organização, reúso e compartilhamento de conhecimento sobre processos de softwares.

5. Trabalhos relacionados e Abrangência

Assim como este trabalho, diferentes abordagens têm sido adotadas na tentativa de melhorar o ensino de Engenharia de Software (ES), dentre elas encontram-se: metodologias de ensino (Soares, 2000; Castro, 2000; e Hazzan, 2003), Jogos e/ou ferramentas (Tao e Qing, 2009; Baker et al., 2003), estudos de caso (Barros e Araujo, 2008; Hilburn, 2007), estudos empíricos (Santos, et al., 2008; Hilburn, 2006). A principal diferença desses trabalhos para o presente, é que aqueles não utilizam a interdisciplinaridade. Outros trabalhos abordam a ES juntamente com outras disciplinas (Cunha, et al., 2008, Silva, 2008; Hilburn, 2006; Santos, 2007; Alves, 2006). No entanto, a ES é utilizada, mas não é desenvolvida dentro delas como é o caso deste trabalho. Além disso, as outras disciplinas estão sempre dentro da área de exatas, sendo que neste trabalho sugere-se relacionar a ES também com disciplinas pertencentes ao domínio de humanas como: Português, Administração e Ética. Esses trabalhos relacionados podem ser utilizados como referências nas diretrizes do presente trabalho. O trabalho apresentado por Werneck (2008) aborda a adoção dos conceitos de ES em outras disciplinas e também pretende obter resultados dessa abordagem. No entanto, o mesmo trabalho não adota os processos de gestão de conhecimento e não propõe a elaboração de diretrizes. Os resultados deste trabalho e o proposto por Wenerck podem ser complementares. O trabalho de Prikladnicki (2006) objetivou discutir a construção do conhecimento na Engenharia de Software, e entender como pesquisas multi, inter e transdisciplinares podem contribuir para a evolução das pesquisas neste contexto. Esse trabalho está sendo utilizado como fundamentação teórica para este. O trabalho mais próximo até o momento encontrado foi o proposto por Hilburn (2002), que sugere a aplicação da qualidade de software durante todo o currículo. O presente trabalho também tem o mesmo propósito, com as diferenças de fornecer a base de conhecimento e abordar outros conceitos de ES e não somente a qualidade de software.

6. Considerações Finais

Este trabalho objetiva melhorar o ensino de ES através da aplicação de seus conceitos em outras disciplinas do curso de Ciência da Computação. Para isso, ele propõe diretrizes que indicarão, para cada disciplina, *como* aplicar a ES, qual o *objetivo* pedagógico dessa aplicação, e o *conhecimento* necessário para que a aplicação seja realizada com êxito. Esse conhecimento será representado por vídeos, transparências, textos, wiki (Leuf, 2001), e referências externas que serão organizadas e disponibilizadas em um portal colaborativo na Internet. O portal também receberá “feedbacks” das aplicações das diretrizes propostas que futuramente serão analisados e publicados. A metodologia desse projeto é genérica e poderá ser estendida para a integração entre outras disciplinas diferentes das que foram mencionadas e entre disciplinas de cursos distintos. O projeto possui baixo custo de implantação, pois não exige acréscimos de horas e nem criação de novas disciplinas no currículo. As principais contribuições desse trabalho são: (i) fornecer diretrizes associadas à conteúdo didático para serem aplicadas, sendo este o principal incentivador da utilização dessas diretrizes; (ii) fornecer resultados empíricos da interdisciplinaridade, o que poderá ajudar em outras pesquisas; (iii) incentivar a interdisciplinaridade na Ciência da Computação sendo esta apontada como fator importante no documento *Grandes Desafios da Pesquisa em Computação no Brasil* (SBC, 2004).

7. Referências

- Abran, A., Bourque P., Dupuis R. and Moore. J. D. (2001), “Guide to the Software Engineering Body of Knowledge – SWEBOK”, IEEE Press.
- Bagert D.J. et al. (1999), “Guidelines for Software Engineering Education”, Version 1.0, Software Eng. Inst., Carnegie Mellon Univ., Pittsburgh, Nov.
- Baker, A.; Navarro, E.O.; van der Hoek. (2003), A. “An experimental card game for teaching software engineering”, Software Engineering Education and Training, Proceedings. 16th Conference on Volume , Issue , 20-22 March 2003 Page(s): 216 – 223.
- Barbosa, E. F. (2004), “Uma contribuição ao processo de desenvolvimento e modelagem de módulos educacionais”, Tese de Doutorado - Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos - SP, Brasil. Orientador: José Carlos Maldonado.
- Barros, M. O., Araújo, R. M. (2008), “Ensinando Construção de Software Aplicada a Sistemas de Informação do Mundo Real”, In: Fórum de Educação em Engenharia de Software Outubro.
- Braga, J. C., Marucci, R. A. (2006), “Proposta de uma Metodologia de Ensino para a Disciplina Interação Humano-Computador”. In: WEI - Workshop de Educação em Computação, 2006, Campo Grande. Anais do XXVI Congresso da SBC p. 215-225.
- Castro, J.F.B, Gimenes, I.M.S, Maldonado, J.C. (2000), “Uma proposta de Plano Pedagógico para a matéria Engenharia de Software”. In: II curso de qualidade de cursos de graduação da área de Computação e Informática, Curitiba, pp. 251-270.
- Cunha, A.M., et al. (2008), “Estudo de Caso abrangendo o Ensino Interdisciplinar de Engenharia de Software” , Fórum de Educação em Engenharia de Software (FEES), Campinas – SP, out.
- Lucena, F. N. (2008), “Projeto Pedagógico do Curso Bacharelado em Engenharia de Software – Universidade Federal de Goiás – UFG”.
- Hazzan, O., Dubinsky, Y., (2003). “Teaching a Software Development Methodology: The case of Extreme Programming”, Proceedings of the 16th Conference on Software Engineering Education and Training (CSEE&T 2003), Espanha, pp. 176-184.
- Hilburn, T.B.; Towhidnejad, M., (2007), “Case for Software Engineering”, Software Engineering Education & Training, 20th Conference on Volume , Issue , 3-5 July 2007 Page(s):107 – 114

- Hilburn, Y. Towhidnejad, M. (2002), "Software quality across the curriculum", in: *Frontiers in Education*, 2002. FIE 2002. 32nd Annual, 6-9 Nov. 2002, Volume: 3, pg, S1G-18- S1G-23 vol.3
- Hilburn, T.B.; Towhidnejad, M.; Nangia, Shen, S., L., (2006), "A Case Study Project for Software Engineering Education Frontiers". *Education Conference*, 36th Annual Volume , Issue , 27-31 Oct. Page(s):1 – 5
- IEEE-CS, (1999), "Software Engineering Code of Ethics and Professional Practice," IEEE-CS/ACM.
- Lethbridge, T.C.; Diaz-Herrera, J.; LeBlanc, R.J.; Thompson, J.B., (2007), "Improving software practice through education: Challenges and future trends", *Future of Software Engineering*, 2007. FOSE apos;07, Volume , Issue , 23-25 May, Page(s):12 – 28.
- Leuf, B., Cunningham, W., (2001), "The Wiki Way", *Quick Collaboration of the Web*. Addison-Wesley.
- Liebowitz J. and Beckman T., (1998), "Knowledge Organizations: What Every Manager Should Know", St. Lucie Press, Boca Raton, Fla.
- Mcconnell, S., (2005), "Code Complete: um Guia Prático para a Construção de Software", 2a edição, Bookman.
- Meyer, B., (2001), "Software Engineering in the Academy Computer", *Los Alamitos*, v. 34, n. 5, pp. 28-35, May.
- Piaget, Jean., (1973) "The epistemology of interdisciplinary relationships". In *Interdisciplinarity*, pp. 127-39, and *Main Trends in Inter-Disciplinary Research*, New York: Harperan Row.
- Prikladnicki, Rafael (2006), Audy, Jorge Luis Nicolas . "Construção de Conhecimento e Complexidade na área de Engenharia de Software". In: *WOSES 2006 - II Workshop Um Olhar Sócio-Técnico sobre a Engenharia de Software*, Vila Velha. p. 51-64.
- Santos, R. P., et al. (2008), "Utilizando Experimentação para Apoiar a Pesquisa em Educação em Engenharia de Software no Brasil", *Fórum de Educação em Engenharia de Software (FEES)*, Campinas – SP, out.
- Sargent, J., (2004) "An Overview of Past and Projected Employment Changes in the Professional *IT* Occupations". *Computing Research News*, 16, 3, May, pp. 1, 21.
- Soares, M. S., (2008) "Uma experiência de ensino de Engenharia de Software orientada a trabalhos práticos". *Fórum de Educação em Engenharia de Software (FEES)*, Campinas – SP, out.
- Sociedade Brasileira de Computação (SBC), (1999), "Currículo de Referência da SBC para Cursos de Graduação em Computação e Informática". Sociedade Brasileira de Computação – SBC.Set.
- Sociedade Brasileira de Computação (SBC), (2004), *Grandes Desafios da Pesquisa em Computação no Brasil 2006 até 2016*, Sociedade Brasileira de Computação - SBC, 2004. 22p.
- Tao W., Qing Z., (2009), "A Software Engineering Education Game in a 3-D Online Virtual Environment", vol. 2, pp.708-710, *First International Workshop on Education Technology and Computer Science*.
- Don Gotterbarn, (1999), "How the New Software Engineering Code of Ethics Affects You", *IEEE Software*, vol. 16, no. 6, pp. 58-64, Nov./Dec.
- Werneck, V., M., B., et al., (2008), "Engenharia de Software no Curso de Ciência da Computação", *Fórum de Educação em Engenharia de Software (FEES)*, Campinas – SP, out.