

# Qual Conhecimento de Engenharia de Software é Importante para um Profissional de Software?

Christiane Gresse von Wangenheim<sup>1</sup>, Djoni Antonio da Silva<sup>2</sup>

<sup>1</sup>PPGCC/INE/UFSC – Federal University of Santa Catarina

<sup>2</sup>CTTMAR/UNIVALI – Universidade do Vale do Itajaí

gresse@gmail.com, djonisilva@gmail.com

**Abstract.** *We present the results of a survey conducted to identify the importance and amount learned of Computer Science topics, principally, focusing on Software Engineering, in the career of software professionals, repeating an adapted version of a survey conducted in 1998 by T. Lethbridge. The survey has been conducted in 2008 among Computer Science graduates in Brazil. The results suggest that various topics may be under-represented in Computer Science courses, whereas others may be less relevant for Computer Science curriculums.*

**Resumo.** *Apresentamos resultados de um estudo realizado para identificar a importância e a quantidade de conhecimentos aprendidos por profissionais da área de software em tópicos dos Cursos de Ciência da Computação, dando ênfase para a Engenharia de Software, repetindo e adaptando uma versão de uma pesquisa realizada em 1998 por T. Lethbridge. A pesquisa foi realizada em 2008 entre graduados em Ciência da Computação no Brasil. Os resultados sugerem que vários tópicos podem estar sendo abordados de maneira insuficiente nos cursos de Ciência da Computação, enquanto outros poderiam ser considerados menos relevantes nos currículos.*

## 1. Introdução

Tipicamente, profissionais de software são formados em cursos de graduação como modo de preparação para atuar na área. O curso de graduação, que se destaca nesta área é o de Ciência da Computação, além dos cursos de Engenharia da Computação e de Sistemas de Informação. No entanto, há anos a indústria se queixa de que os cursos de graduação não ensinam aos alunos as competências necessárias para que eles possam começar a executar o seu trabalho com eficiência (Gudivada, 2003) (Callahan & Pedigo, 2002). Uma das críticas é a falta de atenção com o ensino de Engenharia de Software (ES) na graduação (Mead *et al.*, 1997). Além disso, o papel de um profissional de software deixou de limitar-se somente a conhecimentos técnicos, passando a exigir um vasto conjunto de conhecimentos e habilidades como comunicação, trabalho em equipe, etc. Neste contexto, o ensino adequado de ES é importante para melhorar o estado atual de desenvolvimento de software e ajudar a mitigar muitos dos problemas tradicionais associados à indústria de software.

Contudo, especialmente nos cursos de graduação, os tópicos de ES são normalmente ensinados de forma bastante superficial, abordando essas questões dentro de poucas disciplinas no curso (Mead *et al.*, 1997). E, embora o desenvolvimento do SWEBOOK (IEEE, 2004) e das Diretrizes Curriculares para Graduação (IEEE/ACM, 2005) tenham

sido um avanço significativo, ainda se pergunta se os tópicos de ES são ensinados de maneira adequada no Curso de Ciência da Computação para preparar adequadamente os profissionais desta área.

Esta questão tem sido de grande interesse nos últimos anos e foram realizadas algumas tentativas de investigar o que os profissionais e as empresas esperam do ensino do Curso de Ciência da Computação (CC). Em 1998, Lethbridge (Lethbridge, 2000) realizou uma pesquisa com profissionais da área de software constatando que alguns tópicos da graduação foram considerados menos úteis, enquanto se teve a impressão que para outros tópicos considerados importantes, não foi dada a devida atenção no ensino. Este estudo foi repetido em 2002 por Kitchenham (Kitchenham *et al.*, 2005) com resultados muito semelhantes. Portanto, dez anos depois da primeira pesquisa nos propomos repetir o estudo, buscando obter informações sobre a relevância dos temas abordados nos cursos de ciência da computação segundo opinião dos profissionais desta área (especificamente, no que diz respeito à ES), visando contribuir para a melhoria dos currículos atuais.

## 2. A Pesquisa

Nossa pesquisa é fortemente baseada na pesquisa realizada por (Lethbridge, 2000) (Lethbridge, 1999) e na pesquisa realizada por (Kitchenham *et al.*, 2005).

**Questionário.** Revendo o questionário utilizado por (Lethbridge, 1999) e (Kitchenham *et al.*, 2005), definimos um questionário com 69 tópicos agrupados em 6 categorias (Tabela 1) levando em consideração as recomendações para currículos de cursos de Ciência da Computação (IEEE/ACM, 2001), (IEEE/ACM, 2004), (MEC, 1999) e principalmente, relacionados à ES, o SWEBOK (IEEE, 2004) e o *framework* CMMI (CMMI Product Team, 2006).

Tabela 1. Lista de tópicos e categorias

<b>Software</b> Estruturas de dados e algoritmos Gerência de arquivos Banco de dados Linguagens de programação específicas Teoria de linguagem de programação Medição e análise de desempenho Análise de complexidade computacional e de algoritmo Inteligência artificial Reconhecimento de padrões e de processamento de imagens Computação gráfica Segurança e criptografia Sistemas operacionais Transmissão de dados e redes Processamento paralelo e distribuído Sistemas embarcados Programação Web Ergonomia	<b>Engenharia de Software</b> Gerência de projetos Desenvolvimento de requisitos Gerência de requisitos Métodos formais de especificação Conceitos e tecnologias orientadas a objetos Arquitetura de software Software design/projeto e padrões Teste de software Inspeção e revisão de software Garantia da qualidade de software Gerência de configuração de software Manutenção, reengenharia e engenharia reversa Métricas de software Confiabilidade de software e tolerância à falhas Estimativa de custo/ esforço de software Processo de software e melhoria de processo (CMMI, MPS.BR) Ferramentas de engenharia de software
<b>Hardware</b> Eletrônica digital e lógica digital Arquitetura de microprocessador Arquitetura de sistema de computador Eletrônica analógica Robótica Processamento digital de sinais Projeto VLSI	<b>Matemática</b> Cálculo diferencial e integral Equações diferenciais Álgebra linear e matrizes Probabilidades e estatística Lógica matemática Teoria de grafos Análise combinatória Funções, relações e conjuntos
<b>Áreas Afins</b> Física Química Economia Contabilidade Marketing Gerenciamento Empreendedorismo Psicologia	<b>Ciência Computacional</b> Análise numérica Pesquisa operacional Modelagem e simulação

Filosofia Redação Técnica Oratória Habilidades de trabalho em equipe Liderança Negociação Legal / profissionalismo, ética e sociedade Controle estatístico de processos Metodologia científica Segunda língua	
--	--

Seguindo (Lethbridge, 1999), utilizamos as mesmas quatro perguntas para cada tema (Tabela 2). Cada uma destas questões tem respostas associadas a uma pontuação, que varia de 0 a 5.

Tabela 2. Questões e respostas.

Q1. Quanto você aprendeu sobre isso durante o curso de ciência da computação? 0=Aprendi nada 1 = Tomei-me vagamente familiar 2=Aprendi o básico 3=Me tornei funcional (conhecimento moderado) 4= Aprendi muito 5 =Aprendi em profundidade, me tornei um especialista (aprendi quase tudo)	Q2. Qual é o seu atual conhecimento sobre isso, considerando que você aprendeu isso no emprego e quanto você esqueceu-se desde a sua graduação? 0=Sei nada 1=Estou vagamente familiarizado 2=Sei o básico 3=Estou funcional (conhecimento moderado) 4=Sei muito 5=Sei em profundidade, sou um especialista (sei quase tudo)	Q3. Este assunto específico foi útil para você na sua carreira? 0=Completamente inútil 1=Quase nunca usei 2=Ocasionalmente útil 3=Moderadamente útil - em determinadas atividades 4=Muito útil 5=Essencial	Q4. Quanto seria útil aprender mais sobre este assunto (por exemplo, em cursos complementares)? 0=Inútil aprender mais 1=Difícilmente útil 2=Possivelmente útil 3= Moderadamente útil 4=Importante aprender mais 5=Crítico de aprender mais
---	---	--	---

Além disso, incluímos questões demográficas a fim de caracterizar os participantes. Também utilizamos um conjunto de perguntas como filtro no início do questionário. Duas versões do questionário (em Português e Inglês) foram desenvolvidas e disponibilizadas na web utilizando a ferramenta de código aberto *LimeSurvey* (<http://www.limesurvey.org>). Para reduzir a influência do cansaço nas respostas, a ordem dos grupos e dos tópicos é alterada automaticamente a cada acesso. Antes da coleta de dados, o questionário foi testado dentro do nosso grupo de pesquisa.

**Público-alvo.** Como público-alvo do nosso estudo, definimos Bacharéis em Ciência da Computação que se formaram entre 1998 - 2005 em um curso de CC no Brasil, e que trabalham como profissionais da área de software. Este público alvo foi selecionado com o objetivo de incluir profissionais que tiveram tempo suficiente para adquirir experiências na prática. Por outro lado, foram excluídos profissionais que se formaram há muito tempo seguindo um currículo desatualizado. Como nossa pesquisa está focada em Bacharéis em CC, excluímos graduados de outros cursos como Engenharia da Computação e de Sistemas de Informação. Realizamos o convite para a participação da pesquisa por meio de listas de email, grupos de discussão e fóruns.

**Execução da Pesquisa.** Dados foram coletados durante o período de 20 de Outubro de 2008 a 1 de Dezembro de 2008. Durante esse período, recebemos respostas completas de 48 participantes que trabalham como profissionais na área de software. Entre os participantes estão bacharéis de Ciência da Computação de universidades públicas e privadas do Brasil. Os participantes da pesquisa trabalham com diferentes tipos de softwares, sendo que a maioria trabalha com gestão de informação ou aplicações de software e uma minoria com sistemas embutidos. Eles estão envolvidos em diferentes atividades no processo de software com uma forte concentração nas atividades técnicas.

### 3. Resultados

Uma análise completa e detalhada da pesquisa pode ser encontrada em (Wangenheim & Silva, 2009). Aqui, em razão do foco deste trabalho, apresentamos resultados específicos para a Engenharia de Software.

#### 3.1 Quais tópicos de ES são mais importantes para os profissionais da área de software?

A importância dos tópicos foi avaliada baseando-se no valor da mediana das respostas à pergunta 3: Este assunto específico foi útil para você na sua carreira? Comparado com outros tópicos de Cursos de CC, os tópicos de ES estão entre os mais importantes, todos com uma importância de no mínimo 3. Na Figura 1 é mostrada a ordem dos tópicos de ES por importância.

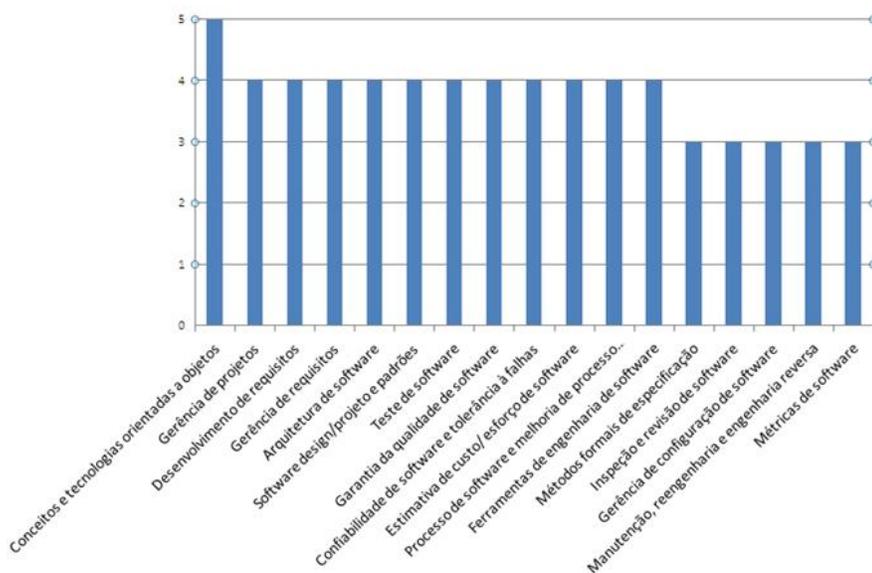


Figura 1. Classificação dos tópicos de ES ordenados pela importância

Esta importância dos tópicos relacionados à ES é enfatizada também pelo fato que, considerando todos os tópicos de Cursos de CC desta pesquisa, aqueles relativos à ES ficaram entre os 25 mais importantes (Wangenheim & Silva, 2009).

#### 3.2 A quantidade do ensino na graduação corresponde à importância dada aos tópicos de ES?

Adaptando a definição utilizada em (Kitchenham *et al.*, 2005), que define o *gap* de conhecimento como sendo a diferença entre a importância e o aprendizado na graduação:  $Gap$  de Conhecimento = (mediana das respostas da pergunta Q3) - (mediana das respostas da pergunta Q1) (Figura 2). Assim, um valor negativo indica que a graduação deixou a desejar em relação à importância dada ao tópico.

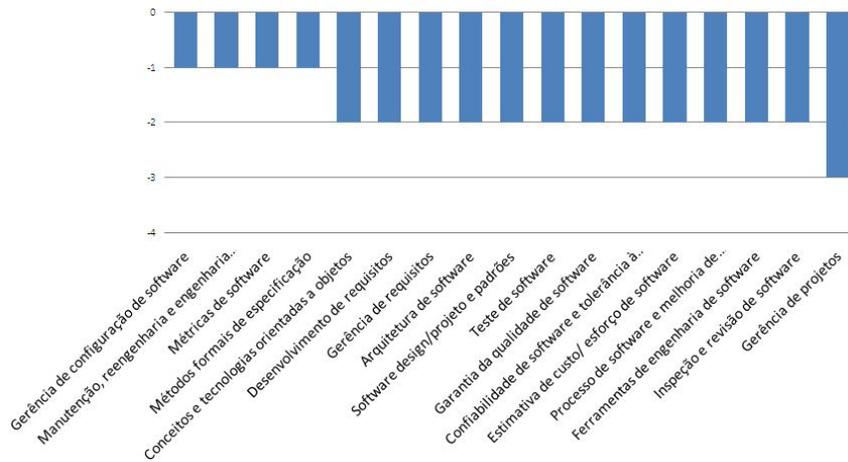


Figura 2. *Gap* de conhecimento por tópicos de ES

Analisando o *gap* de conhecimento, podemos observar que todos os tópicos de ES parecem não ser tratados suficientemente na graduação considerando a importância dada aos mesmos, e isto, especialmente em relação á gerência de projetos.

### 3.3 Os profissionais da área de software aprendem no trabalho o que é necessário?

De acordo com (Lethbridge, 1999), analisamos a quantidade aprendida referente a cada tópicos no trabalho (ou esquecida desde a graduação) com base na diferença entre as respostas das perguntas 1 e 2 (Figura 3).

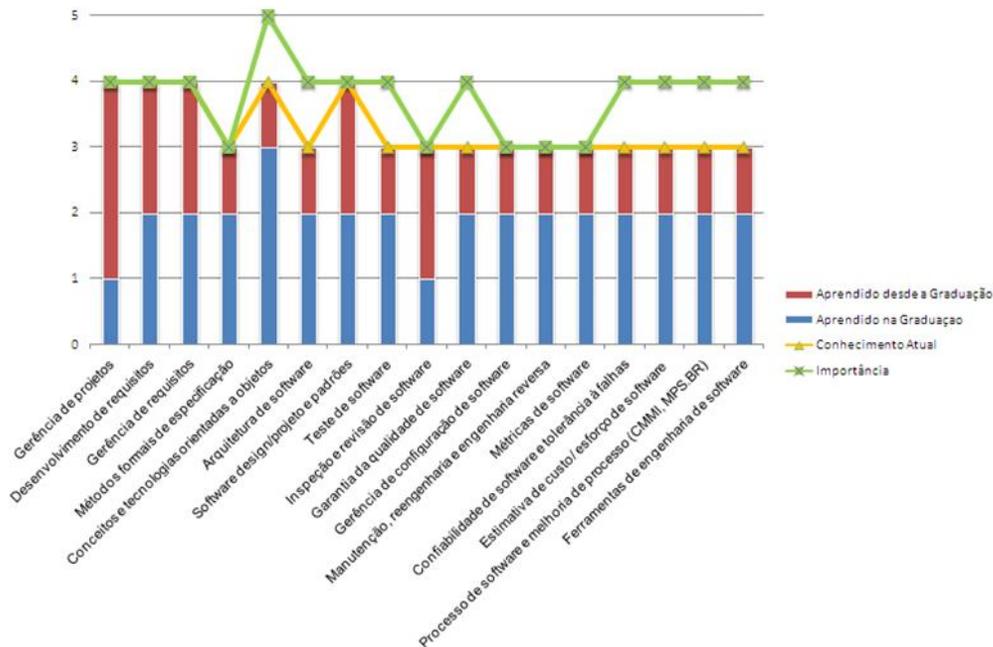


Figura 3. Quantidade aprendido nos tópicos de ES em relação a educação formal e treinamentos

Os profissionais confirmaram que aprenderam mais sobre tópicos de ES depois da graduação. Sobretudo, o tópico de gerência de projetos é enfatizado como um dos temas que os profissionais mais tiveram que aprender depois de se formarem. No entanto, o conhecimento atual dos participantes ainda não atinge o nível de importância

dado a vários tópicos de ES, salientando, mais uma vez, o *gap* de conhecimento existente no que diz respeito ao aprendizado dos tópicos de ES, tanto na graduação, quanto na formação profissional.

### 3.4 A importância dos tópicos mudou em comparação com as outras pesquisas?

A fim de analisar esta questão, comparamos a classificação dos tópicos ordenados pela importância em nossa pesquisa com a classificação das pesquisas realizadas por Lethbridge e Kitchenham. No entanto, é importante salientar que a importância foi avaliada de forma diferente em cada uma das pesquisas. Em nosso estudo, analisamos importância considerando o valor da mediana para as respostas da pergunta 3. No estudo Lethbridge, várias maneiras de avaliar a importância foram aplicadas (Lethbridge, 1999). No entanto, a apresentação da lista completa dos tópicos está disposta pelo cálculo do valor médio das respostas das questões 3 e 4. No estudo de Kitchenham, a importância é calculada sobre a proporção de respostas dos participantes, que receberam três pontos (indicando, que a utilidade do tópico era no mínimo moderada) ou mais para a questão 3. Outro aspecto a ser considerado é a variação dos tópicos considerados em cada uma das pesquisas. Embora, fortemente baseado no estudo de Lethbridge, modificamos os tópicos para ajustá-los as nossas necessidades e aos avanços da área, incluindo, por exemplo, o tópico de programação web. Por outro lado, a pesquisa realizada por Kitchenham abrange os grupos de software e ES apenas, não incluindo, por exemplo, áreas afins ou *hardware*. Em razão das quantidades diferentes de tópicos nas pesquisas, consideramos então os tópicos de ES que estavam entre os 15 mais e os 15 menos importantes da pesquisa de Kitchenham, e os tópicos que estavam entre os 25 mais e 25 menos importantes com relação aos nossos resultados e os do Lethbridge.

Tabela 3. Comparação do ranking de importância (*os tópicos presentes em todas as três pesquisas são marcados em azul, os tópicos presentes em duas pesquisas são marcados em verde*).

Nossa pesquisa	(Lethbridge, 1999)	(Kitchenham et al., 2005)
<b>Tópico de SE entre os tópicos mais importantes</b>		
Conceitos e tecnologias orientadas a objetos	Software <i>design</i> /projeto e padrões	Gerência de projetos
Gerência de projetos	Arquitetura de software	Levantamento e análise de requisitos
Desenvolvimento de requisitos	Levantamento e análise de requisitos	Arquitetura de software
Gerência de requisitos	Conceitos e tecnologias orientadas a objetos	Métodos de projetos e análise
Arquitetura de software	Métodos de projetos e análise	Testes, verificação & garantia da Qualidade
Software design/projeto e padrões	Gerência de projetos	Práticas de projetos de software
Teste de software	Testes, verificação & garantia da Qualidade	Conceitos e tecnologias orientadas a objetos
Garantia da qualidade de software	Gerência de liberação e configuração	
Estimativa de custo/ esforço de software	Confiabilidade de software e tolerância a falhas	
Processo de software e melhoria de processo (CMMI, MPS.BR)		
Ferramentas de engenharia de software		
<b>Tópico de SE entre os tópicos menos importantes</b>		
Nenhum	Nenhum	Manutenção, reengenharia e engenharia reversa
		Métodos de especificação formal
		Estimativa de custo de software
		Confiabilidade de software e tolerância a falhas
		Normas e processos (CMM/ISO 9000 etc.)
		Métricas de software

Comparando os resultados dos três estudos, podemos observar que praticamente todos os tópicos de ES citados entre os mais importantes da pesquisa Lethbridge e Kitchenham

também continuam entre os temas mais importantes no nosso estudo. No entanto, o fato do nosso estudo ter mais tópicos de ES citados entre os mais importantes pode indicar uma tendência de aumento da importância dos tópicos de ES. Isso inclui até mesmo tópicos como melhoria de processo de software e estimativa dos custos, que, no estudo de Kitchenham, ficaram classificados entre os tópicos menos importantes. Uma observação interessante é que no nosso estudo (bem como no estudo Lethbridge) nenhum tópico de ES foi citado entre os menos importantes, ao contrário do estudo de Kitchenham, em que alguns tópicos de ES foram citados entre os menos importantes.

#### 4. Implicações para as orientações curriculares

Embora tenhamos de ter muito cuidado ao interpretar os resultados, devido ao pequeno tamanho da amostra e a sua restrição a profissionais brasileiros, os resultados podem ser relevantes como base para reexaminar as competências profissionais necessárias para ter sucesso na área de software, especialmente em comparação com os outros estudos. A Tabela 4 mostra uma comparação da classificação dos tópicos de ES por importância. Esta comparação é resultante da nossa pesquisa com o resumo do corpo de conhecimento de Cursos de CC, conforme apresentado nas diretrizes curriculares para a graduação em computação (IEEE/ACM, 2001), indicando também o número mínimo de horas sugerido para cada tópico.

Tabela 4. Mapeamento dos tópicos de ES ao corpo de conhecimento de Cursos de CC

Nossa Pesquisa		IEEE/ACM CS Curriculum guidelines	
Grau da Importância	Tópicos de Engenharia de Software	Corpo de conhecimentos de Cursos de Ciência da Computação	Número mínimo sugerido de horas *
1	Conceitos e tecnologias orientadas a objetos	PL6. Programação orientada a objeto	10
2	Gerência de projetos	SE8. Gerenciamento de Projetos de Software	3
	Desenvolvimento de requisitos	SE5. Especificações e requisitos de software	4
	Gerência de requisitos		
	Arquitetura de software	SE1. Projeto de Software	8
	Software design/projeto e padrões		
	Teste de software	SE6. Validação de Software	3
	Garantia da qualidade de software		
	Confiabilidade de software e tolerância à falhas	SE11. Confiabilidade de Software	-
	Estimativa de custo/ esforço de software		
	Processo de software e melhoria de processo (CMMI, MPS.BR)	SE4. Processos de Software	2
	Ferramentas de engenharia de software	SE3. Ambientes e Ferramentas de Software	3
	Métodos de especificação formal	SE10. Métodos Formais	-
	Inspeção e revisão de software	<i>Como parte da validação de software(SE6)</i>	
	Gerência de configuração de software		
Manutenção, reengenharia e engenharia reversa	SE7. Evolução de Software	3	
Métricas de software			

\*O número representa o número mínimo de horas necessárias para atingir o padrão do curso conforme sugerido pelo (IEEE/ACM, 2001). A não alocação do número de horas indica que o conteúdo do curso não possui número mínimo de horas indicado na especificação.

Esta comparação pode indicar uma falta de dedicação a certos tópicos de ES considerados importantes. Para alguns temas podemos identificar uma falta total de consideração, como, por exemplo, gerenciamento de configuração de software, que na prática é considerada como uma base essencial, não só para engenheiros de software, mas para qualquer profissional de software. E, em geral, considerando a sugestão de um total de no mínimo 280 horas para um curso de CC, a atribuição de cerca 36 horas aos tópicos de ES não parece corresponder com a percepção da importância destes tópicos.

#### 5. Conclusões

Nossa pesquisa parece confirmar as críticas de que as competências de ES, necessárias aos profissionais de software, muitas vezes não estão sendo adequadamente abordadas nos cursos de CC. Comparando nossos resultados com estudos anteriores, parece que a importância dos tópicos SE cresceu ainda mais hoje em dia. É claro, que o desenvolvimento de um currículo de CC adequado e atualizado não é trivial. Um problema central é que é simplesmente impossível cobrir todo o material potencialmente relevante, especialmente considerando a variedade de empregos e domínios em que profissionais de software atuam. Mesmo assim, acreditamos que os resultados da nossa pesquisa podem contribuir como um *feedback* às práticas educacionais atuais.

### **Agradecimentos**

Gostaríamos de agradecer o T. Lethbridge por fornecer o material utilizado na pesquisa em 1998 e o grupo de pesquisa Cyclops/UFSC por hospedar o sistema de pesquisa. Este trabalho foi apoiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq – Brasil e a UNIVALI – Universidade do Vale do Itajaí/Brasil.

### **Referências**

- Callahan, D., Pedigo, B. Educating Experienced IT Professionals by Addressing Industry's Needs. IEEE Software, 19/5, 2002.
- CMMI Product Team. CMMI for Development, Version 1.2. Technical Report CMU/SEI-2006-TR-008, Carnegie Mellon University/Software Engineering Institute, 2006
- Gudivada, V. N. The Computing Profession at a Crossroads. IEEE Computer, 36/5, 2003.
- IEEE CS. SWEBOK - Guide to the Software Engineering Body of Knowledge, 2004.
- Joint Task Force on Computing Curricula IEEE/ACM. Computing Curricula 2001 - Computer Science. Final Report, 2001.
- Joint Task Force on Computing Curricula IEEE/ACM. Software Engineering 2004 - Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, 2004.
- Joint Task Force for Computing Curricula ACM/AIS/IEEE-CS. Computing Curricula 2005 - The Overview Report, 2005.
- Kitchenham, B. *et al.* An Investigation of Software Engineering Curricula. Journal of Systems and Software, 74/3, 2005.
- Lethbridge, T. C. The Relevance of Education to Software Practitioners: Data from the 1998 Survey. Technical Report TR-99-06, University of Ottawa - Computer Science, 1999.
- Lethbridge, T. C. What Knowledge Is Important to a Software Professional? IEEE Computer, 33/5, 2000.
- Mead, N., *et al.* The State of Software Engineering Education and Training. IEEE Software, vol. 14, no. 6, 1997.
- Ministério de Educação. Currículo de Referência da SBC para Cursos de Graduação em Computação e Informática, 1999.
- Wangenheim, C. Gresse von, da Silva, D. A. Survey on the Relevance of Topics in Computer Science Education. Technical Report LQPS001.09E, Universidade do Vale do Itajaí, 2009. ([http://www.inf.ufsc.br/~gresse/download/LQPS001\\_09E.pdf](http://www.inf.ufsc.br/~gresse/download/LQPS001_09E.pdf))