

Uma Metodologia de Ensino das Técnicas de Qualidade, Confiabilidade e Segurança (*Safety*) de Software Aplicadas num Protótipo de Sistema de Software de Computador Embarcado de Tempo Real no Instituto Tecnológico de Aeronáutica – ITA.

Francisco S. Marcondes¹, Adilson M. Cunha¹, Luiz Alberto V. Dias¹

¹Instituto Tecnológico de Aeronáutica
Praça Marechal Eduardo Gomes, 50
12228-900 São José dos Campos, SP.

yehaaain@gmail.com, {cunha,vdias}@ita.br

Abstract. *This paper describes the application of Problem Based Learning – PBL as an instructional methodology to teach concepts and techniques about software quality, reliability and safety, taught at the Brazilian Aeronautics Institute of Technology (ITA) on the 2nd semester of 2006. It includes a brief description of a case study based upon an ongoing project at the ITA. The case study was adopted on different courses for the development of an embedded real time software system, using an Integrated Computer Aided Software Engineering Environment for quality measurements.*

Resumo. *Este artigo descreve a aplicação da metodologia de Aprendizado Orientado a Problemas (Problem Based Learning - PBL) para o ensino de conceitos e técnicas sobre qualidade, confiabilidade e segurança (safety) de software, ministrados no Instituto Tecnológico de Aeronáutica (ITA), no 2º semestre de 2006. Ele inclui uma breve descrição de um estudo de caso baseado num projeto em andamento no ITA. O estudo de caso foi adotado em diferentes disciplinas para o desenvolvimento de um sistema de software embarcado de tempo real, utilizando ferramentas de um ambiente integrado de engenharia de software ajudada por computador para medições de qualidade.*

1. Introdução

Segundo Cunha (2006a, 2006b), a obtenção da qualidade pode ser entendida como um método gerencial que utiliza processos e procedimentos disseminados por organizações que, ao longo do tempo, buscam uma posição competitiva para propiciar a satisfação da sociedade.

Em sistemas de Tempo Real, entende-se como qualidade a conformidade com requisitos, isto é, a garantia de que o software não esteja defeituoso, não contenha erros, nem apresente falhas. Confiabilidade aqui é definida como a qualidade no tempo, isto é, a capacidade do sistema atingir a meta de se manter em conformidade com os requisitos previamente estabelecidos, num determinado tempo, ou seja, não se deteriorando com o tempo. O primeiro significado do termo segurança (*safety*) de software refere-se a

confiabilidade no uso, isto é, o risco de danos físicos ou lógicos que o sistema de software possa apresentar durante o seu uso, envolvendo quem o desenvolve, o utiliza, ou até mesmo quem pode ser afetado direta ou indiretamente pela sua utilização. O outro significado do termo segurança (*security*) de software refere-se ao risco que o sistema de software oferece quanto ao seu uso não autorizado. Neste artigo, apenas o primeiro significado do termo segurança (*security*) de software é considerado [Cunha 2006a, 2006b].

Este artigo descreve a experiência prática na utilização da metodologia de ensino nas disciplinas CES-32 e CE-230 Qualidade, Confiabilidade e Segurança de Software ministradas, respectivamente, para os alunos do 5º ano do Programa de Graduação em Engenharia da Computação e do Programa de Pós-graduação em Engenharia Eletrônica e Computação, na Área de Informática (PG/EEC-I) do ITA, durante o 2º semestre de 2006.

A maior contribuição acadêmica e tecnológica dessas duas disciplinas foi a de propiciar a capacitação de seus alunos na auditoria de um protótipo de Sistema de Software de Computador (SSC) para um Veículo Aéreo Não Tripulado. Este protótipo, denominado VANT-EC-SAME, operando a partir de uma Estação de Controle e de um Satélite de Monitoramento Ecológico, utilizou o Processo Unificado da Rational (*Rational Unified Process – RUP*), o Padrão da Linguagem de Modelagem Unificada (*Unified Modeling Language – UML*) e as ferramentas de um Ambiente Integrado de Engenharia de Software Ajudada por Computador (*Integrated Computer Aided Software Engineering Environment – I-CASE-E*).

O ensino baseou-se no Aprendizado Orientado a Problemas - AOP (*Problem Based Learning – PBL*) de [Thomas et al. 1999], onde os alunos trabalharam em grupos, buscando resolver problemas reais e gerar relatórios formais, dividindo e compartilhando o que foi aprendido. Esta abordagem trouxe muitos benefícios aos alunos, incluindo maior aprofundamento no conteúdo das disciplinas, maior independência, aumento de motivação e melhoria das características das pesquisas e das buscas pela solução de problemas [Jones & Rasmussen 1997].

2. Aprendizado Orientado a Problemas

A PBL foi utilizada pela primeira vez, na década de 60, na Universidade de MacMaster no Canadá. Charles Engel appud [Boud & Feletti, 1991] indica que a PBL está mais para uma abordagem de aprendizado do que para uma técnica de ensino, pois as técnicas de ensino tradicionais exigem a memorização de grande quantidade de conteúdo, enquanto a PBL busca estimular nos alunos a capacidade de desenvolver o próprio aprendizado.

"Essa metodologia apresenta como características principais o fato de ser centrada no aluno, se desenvolver em pequenos grupos tutoriais, apresentar problemas num dado contexto, ser um processo ativo, cooperativo, integrado e interdisciplinar e orientada para a aprendizagem do adulto." [MedSobral, 2006]

Através de um dado problema, os alunos devem se reunir em grupos pequenos que buscam a solução do problema proposto através da orientação do tutor. Na busca da solução, os alunos são induzidos a trocar experiências, fazer pesquisas aplicadas visando resolver o problema e discutir aquilo que foi aprendido, propondo a melhor solução. A busca da solução tem intervenção do tutor que direciona o aprendizado,

fornecendo o conceito em sala, a bibliografia e a orientação individual para cada grupo. A estratégia utilizada depende de cada grupo e do problema abordado.

A PBL vem sendo utilizada com sucesso nas disciplinas CES-63 Sistemas Embarcados, CE-235 Sistemas Embarcados de Tempo Real e CES-32 e CE-230 Qualidade, Confiabilidade e Segurança de Software, ao longo dos últimos cinco anos, no ITA. Segundo a maioria dos alunos, o conteúdo e os detalhes da matéria são muitas vezes esquecidos, restando apenas a aplicação dos conceitos na solução dos problemas nos laboratórios. Acredita-se que isso se deva ao fato dos alunos terem participado da construção do conhecimento em laboratório, por meio de debates, aplicações e relatórios, permitindo sedimentações e disseminações dinâmicas.

Esta metodologia de ensino tem-se mostrado muito eficaz, quando utilizada para a capacitação de alunos com elevado grau de maturidade e alguma experiência de mercado, como tem sido o caso dos alunos dessas disciplinas. Além de estimular a utilização e a troca de experiências pessoais para aplicação num mesmo problema, o tutor ou professor faz com que sejam aplicadas na prática novos conceitos e novas métodos ensinados em sala.

3. Metodologia Utilizada

Inicialmente, foi atribuída a cada aluno a responsabilidade de definir e auditar as normas de qualidade de software utilizadas para verificar uma Unidade do Protótipo do Sistema de Software de Computador (SSC) do Projeto VANT-EC-SAME construído pelos alunos das disciplinas CES-63 e CE-235 [Cunha 2006a, 2006b].

Entre as quatro disciplinas ministradas (CES-32, CE-230, CES-63 e CE-235), conseguiu-se emular, em nível acadêmico, o ambiente de uma *software house*, onde o auditor da qualidade não representava a mesma pessoa que desenvolvia o Projeto [Dos Santos & Cunha 2005]. O processo de desenvolvimento de software utilizado foi o *Rational Unified Process (RUP)* da IBM / Rational [IBM 2007].

No primeiro nível de integração, com duração de 4 semanas, cada aluno, individualmente, preencheu os artefatos Plano de Garantia da Qualidade, Plano de Teste e Caso de Teste previstos no RUP [IBM 2007, Kruchten 2000].

Neste nível de integração foi utilizado o MS Project para apoiar a aplicação de um Plano de Garantia da Qualidade de Software para as Unidades de Software de Computador (USC) alocadas aos alunos [Dos Santos & Cunha 2005].

No segundo nível de integração, com duração de 8 semanas, cada aluno, individualmente, elaborou uma estimativa de esforços necessários para o desenvolvimento do protótipo do sistema, baseando-se na técnica de Pontos por Casos de Uso - PCU e utilizando o "Programa Estimativa" [Silva & Cunha, 2006].

Os auditores das USC agruparam-se para auditar os Componentes de Softwares de Computador (CSC), onde os artefatos gerados no primeiro nível de integração foram unificados e consolidados, formando os artefatos do CSC [Dos Santos & Cunha 2005].

Numa avaliação intermediária de desempenho, os alunos utilizaram em laboratório, as ferramentas *Rational Quality Architect (RQA)* e *Rational Test Real Time (RTRT)* [IBM 2007], para praticar a compreensão dos conceitos nos seus CSCs.

Com base no protótipo do software desenvolvido nas disciplinas CES-63 e CE-235, cada grupo de alunos: 1) executou o *Harness Test* da ferramenta RQA [IBM 2007]; 2) auditou e consistiu o referido código-fonte com os artefatos inicialmente desenvolvidos; e 3) realizou uma revisão das estimativas de tempos e esforços necessários ao desenvolvimento, visando a sua adequação aos prazos da disciplina.

No terceiro nível de integração, com duração de 4 semanas, foi realizada a integração dos Itens de Configuração de Software de Computador (ICSC), onde os alunos integraram os artefatos dos CSCs e os consolidaram nos artefatos dos ICSCs [Dos Santos & Cunha 2005].

A avaliação final de desempenho dos alunos nas disciplinas foi constituída: 1) da integração dos ICSCs no Protótipo do Sistema de Software do Projeto VANT-EC-SAME (SSC); 2) de uma análise de sensibilidade do código-fonte gerado, utilizando pelo menos três métricas de qualidade de software; 3) de uma avaliação da estimativa de esforços por PCU; 4) de uma revisão dos artefatos utilizados; e 5) da utilização da ferramenta de documentação SoDA IBM / Rational, para geração automática de Relatórios [IBM 2007].

Tais relatórios foram publicados nas páginas individuais dos alunos e apresentados num seminário para compartilhamento dos conhecimentos adquiridos.

4. Estudo de Caso

O Problema apresentado como Estudo de Caso das disciplinas CES-32 e CE-230 foi o do Protótipo do Sistema de Software do Projeto VANT-EC-SAME. Ele consistiu em se avaliar o software desenvolvido nas disciplinas CES-63 e CE-235, que foram ministradas concomitantemente pelo mesmo professor. A idéia foi a de se medir, parcialmente e sem completeza, a implementação acadêmica de um protótipo de SSC capaz de propiciar o monitoramento das águas da Bacia Amazônica.

Os dados de entrada do protótipo seriam obtidos a partir de plataformas fixas (Pontos de Coleta de Dados - PCDs) e de plataformas móveis (Veículos Aéreos Não Tripulados - VANTs). Esses dados seriam filtrados e enviados para uma Sala de Situação e complementados com dados históricos provenientes de bancos de dados e de modelos hidrológicos previamente desenvolvidos por terceiros, para servir na previsão de desastres naturais tais como enchentes e secas.

Caso os resultados dos modelos hidrológicos propiciassem previsões de situações de alerta ou emergência, alarmes poderiam ser acionados. O tomador de decisões neste caso poderia então tomar as providências apropriadas para minorar os principais efeitos e impactos da situação hidrológica [Cunha 2006a, 2006b].

5. Principais Resultados Obtidos

a. Resultados Práticos

A metodologia empregada possibilitou aos alunos das disciplinas um contato com os conceitos e as diretrizes principais de qualidade, confiabilidade e segurança (*safety*) de software implementadas no RUP, bem como a realização de experimentos práticos em laboratório, utilizando a garantia da qualidade num Protótipo de SSC.

No primeiro nível de integração, os alunos realizaram, individualmente, um planejamento de atividades no MS Project e preencheram alguns artefatos considerados essenciais para propiciar a qualidade do software. Esta experiência possibilitou atuarem como Gerentes de Qualidade, definindo normas e métricas utilizadas para cada USC.

Para preencher os artefatos selecionados do RUP, os alunos aprimoraram habilidades de pesquisa buscando as métricas de maior relevância para a aplicação em suas USCs. Além dos modelos fornecidos pelo RUP, foram proporcionados também exemplos de artefatos previamente preenchidos, como referência para a preparação dos documentos.

Os artefatos gerados foram: 1) o Plano de Garantia da Qualidade utilizado para registrar a política de garantia da qualidade do protótipo; 2) o Plano de Teste que descreveu a política de testes do protótipo; e 3) o Caso de Teste que, com base na descrição dos Casos de Uso, propiciou a sua derivação em casos de teste, representando a verificação da suas conformidades com os Casos de Uso efetivamente elaborados.

No segundo nível de integração, as USCs foram consolidados em CSCs por grupos formados por 5 mais ou menos 2 alunos, após discussões sobre documentos, planejamentos e definições das principais normas e métricas aplicáveis aos artefatos.

Essas práticas propiciaram o desenvolvimento, a consolidação, a aplicação e a difusão dos conhecimentos teóricos adquiridos em sala de aula nos exercícios de laboratório. Além de consistir e consolidar as estimativas realizadas no MS Project, essas práticas também propiciaram o confronto de planejamentos e o ajuste de cronogramas, de acordo com as experiências adquiridas.

A partir dos artefatos gerados nas disciplinas CES-63 e CE-235, uma estimativa de esforços com base na técnica de Pontos de Casos de Uso (PCU) foi elaborada para os CSCs e utilizada como base para as revisões de planejamento. As estimativas de PCU foram elaboradas utilizando-se o “Programa Estimativa”, apresentado na Figura 1 [Silva & Cunha, 2006]. Neste programa, os alunos puderam configurar os principais parâmetros previstos pela técnica, compreendendo a importância da utilização desta técnica, bem como a facilidade do uso de ferramentas automatizadas.

Nome	Complexidade	Estimativa	
Comunicação	Medio		
Total de Pesos Não Ajustados dos Atores: 1 Total de Pesos Não Ajustados dos Casos de Uso: 40 Pontos Totais Não Ajustados: 41 Fator de Complexidade Técnica: 1,020 Eficiência do Fator Ambiental: 0,920 Pontos Totais de Casos de Uso Ajustados: 38,474		Porcentagem de Segurança [%]: 20 UCP + Margem de Segurança: 923,38 Quantidade de Membros na Equipe: 3 Dedicação Mensal de cada membro: 80 Dedicação Mensal da Equipe: 240 Total de Meses: 3,85 Total de Anos: 0,32	
Horas gasta por pessoa para cada PCU. 20 769,48			
Informações dos Atores Atores Simples 1 Atores Médio 0 Atores Complexo 0 Total de Atores 1		Informações dos Casos de Uso Casos de Uso Simples 4 Casos de Uso Médio 2 Casos de Uso Complexo 0 Total de Casos de Uso 6	

Figura 1. – Exemplo de estimativa no “Programa Estimativa”

Em paralelo com as aulas teóricas, foram apresentadas as ferramentas I-CASE-E voltadas para a realização de testes e medições de qualidade. Como avaliação intermediária de desempenho, cada aluno se utilizou das ferramentas RQA e RTRT, mostrando a compreensão do seu uso.

No terceiro nível de integração, os grupos de CSCs foram consolidados em ICSCs e testados, utilizando o RQA. A Figura 2 mostra um exemplo da aplicação de um teste de caixa-preta onde o responsável pelo teste define a previsão de funcionamento da integração e utiliza o RQA para validar essa previsão. É possível então, verificar o funcionamento, auditar as normas de qualidade utilizadas, revisar os artefatos e estimativas atualizadas do cronograma e comparar o esforço previsto com o realizado.

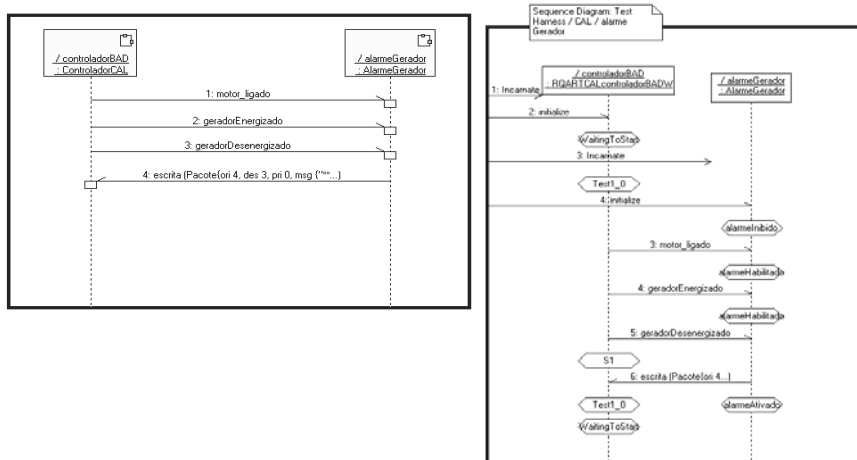


Figura 2 – Teste de caixa-preta no RQA

Esperava-se que os testes dos ICSCs no RQA não fossem apresentar falhas, pois uma vez realizados com sucesso os testes em nível anterior, os de nível posterior deveriam refletir tais características. Em alguns casos, isso foi verdade, em outros não.

Nos casos em que não foi observado o comportamento esperado, pôde-se detectar dois tipos de falhas. No primeiro tipo, os testes do CSC em nível anterior encontravam-se incorretos, e uma vez o software corrigido, tornou-se possível sanar o problema do ICSC. No segundo tipo de falha, a integração dos CSCs foi realizada de maneira equivocada, e o problema pôde ser logo identificado e corrigido.

Para a avaliação final de desempenho dos alunos nas disciplinas, com base nas experiências obtidas, foram consolidados, no curto espaço de tempo de cerca de 2 semanas, os principais artefatos dos ICSCs. Neste mesmo espaço de tempo, foi auditado o SSC com base nos artefatos integrados e cada grupo ficou responsável por auditar o seu próprio CSC.

Nesta auditoria, foram verificados os cumprimentos do cronograma e da estimativa de PCU, avaliados os principais resultados e encontradas as principais falhas. Por fim, utilizando-se o RTRT mostrado na Figura 3 para avaliar os CSCs, foram apresentadas as análises de pelo menos três métricas de avaliação da qualidade do protótipo do sistema de software (SSC) produzido.

FILE: CAL.cpp		FILE: BAD.cpp	
Halstead Metrics		Halstead Metrics	
Difficulty	39	Difficulty	28
Effort	206793	Effort	120754
Errors Estimation	1.77	Errors Estimation	1.43
Size	793	Size	630
Testing Time	3 hrs 11 min 28 sec	Testing Time	1 hrs 51 min 48 sec
Volume	5302.38	Volume	4312.66
Vocabulary	103	Vocabulary	115
Lines & Comments		Lines & Comments	
Comments only Lines	44	Comments only Lines	26
Comments	38	Comments	28
Empty Lines	26	Empty Lines	25
Source only Lines	282	Source only Lines	187
Source & Comments Lines	4	Source & Comments Lines	4
Lines	356	Lines	242
Comment Rate	13.48 %	Comment Rate	12.39 %
Statements & Levels		Statements & Levels	
Maximum of Nested Level	8	Maximum of Nested Level	8
Sum of Nested Level	20	Sum of Nested Level	17
Maximum Statements	34	Maximum Statements	24
Total Statements	56	Total Statements	50
V(g)		V(g)	
Maximum of V(g)	22	Maximum of V(g)	16
Mean of V(g)	5.71	Mean of V(g)	3.22
Standard Deviation of V(g)	7.16	Standard Deviation of V(g)	4.77
Sum of V(g)	40	Sum of V(g)	29

Figura 3 – Exemplo de resultados obtidos nos testes efetuados pelo RTRT

Após cada nível de integração, foram gerados pelos alunos relatórios de atividades, e publicados em suas páginas individuais, os principais resultados apresentados sob a forma de seminários para difundir o conhecimento obtido.

b. Principais Resultados da Aprendizagem

A divisão da atividade de garantia da qualidade proposta pelo RUP, dividida em grupos alunos, propiciou o desenvolvimento de diversas capacitações durante a execução de tarefas individuais, buscando-se objetivos práticos e viabilizando-se constantes e intensas trocas de informações e refinamentos de aplicações de conceitos.

O principal resultado de ensino foi a obtenção de soluções diferentes aplicadas a problemas semelhantes e integrantes, sempre com um número limitado de requisitos num protótipo reduzido sem a necessidade de completeza, propiciando o uso de criatividade para o desenvolvimento de soluções.

A utilização de um estudo de caso real permitiu a motivação dos alunos num ambiente acadêmico e o desenvolvimento de capacitações possíveis de serem aplicadas em Projetos de médio para grande porte, atendendo a requisitos de qualidade, que embora incompletos, puderam ser implementados com grande sucesso.

A obtenção desses resultados deveu-se a participação dos alunos das disciplinas CES-63 e CE-235, que trouxeram um grau de realismo apropriado para o ambiente acadêmico, emulando o que ocorre numa *software house*, onde auditores de qualidade fazem parte de um grupo diferente dos desenvolvedores.

Ao final desta experiência, constatou-se que cerca de 80 alunos envolvidos em quatro disciplinas, foram capazes de produzir, individualmente, 40 USCs, gerando e

auditando, aproximadamente, 140 mil linhas de código em C++, quanto a qualidade, confiabilidade e segurança, em apenas 17 semanas.

6. Conclusão

O protótipo de sistema de software do projeto VANT-EC-SAME propiciou a apresentação e aplicação dos conceitos de Engenharia de Software e das ferramentas utilizadas no desenvolvimento de um Protótipo Acadêmico de Projeto de médio para grande porte. Com esta experiência os alunos foram capacitados a contribuir na implementação e auditoria de qualidade em Projetos de desenvolvimento no mercado.

O uso de I-CASE-E proporcionou o aumento de produtividade tanto na implementação quanto na auditoria de qualidade, mostrando a importância do uso dessas ferramentas no apoio aos projetos de desenvolvimento. Devido a estas facilidades, diferentes opções puderam ser exploradas pelos alunos, possibilitando soluções bastante criativas.

A PBL mostrou-se eficaz para o ensino de Engenharia de Software, como uma forma de capacitar, por meio da troca de experiências, os profissionais para atuação em Projetos de Desenvolvimento assim que encerrado o treinamento nessas disciplinas.

Esta iniciativa acadêmica realizada com sucesso, no 2º semestre de 2006, representou um importante aumento na qualidade do ensino-aprendizado em Engenharia de Software que vem sendo realizado no âmbito da Divisão de Ciência da Computação do Instituto Tecnológico de Aeronáutica - ITA, contribuindo de maneira efetiva para o aumento do nível de capacitação de mão de obra altamente especializada no Brasil.

Bibliografia

- Cunha, A. M. (2006a) "Software Quality, Reliability and Safety (CE-230)" Lecture Notes, Brazilian Aeronautical Institute of Technology (ITA), www.comp.ita.br/cunha.
- Cunha, A. M. (2006b) "Real Time Embedded Systems (CE-235)" Lecture Notes, Brazilian Aeronautical Institute of Technology (ITA), www.comp.ita.br/cunha.
- Dos Santos, W. A., Cunha, A. M. (2005) "An MDA Approach for a Multi-Layered Satellite On-Board Software Architecture", Proceedings of Fifth Working IEEE/IFIP Conference on Software Architecture (WICSA 5), Pittsburgh, USA, November.
- IBM Software (2007)- Rational Suite - Product Overview - available in <http://www.ibm.com/software/awdtools/suite/>, March.
- Jones, B. F., Rasmussen, C. M. (1997) "Moffitt, M. C. Real-life problem solving: A collaborative approach to interdisciplinary learning", Washington, DC: American Psychological Association.
- Kruchten, K. C. (2000) "The Rational Unified Process - An Introduction", 2nd ed., New York, NY: Addison-Wesley.
- Silva, C. M. B., Cunha, A. M. (2006) "Programa Estimativa - Versão 1.0", São José dos Campos, SP: Instituto Tecnológico de Aeronáutica.
- Thomas, J. W., Mergendoller, J. R., and Michaelson, A. (1999) "Project-based learning: A handbook for middle and high school teachers", Novato, CA: The Buck Institute for Education.
- Boud, D; Feletti, G. (1991) "The challenge of problem based learning", Kogan Page, 2nd Edition.
- MedSobral (2006) "Problem Based Learning" <http://www.medsobral.ufc.br/pbl/pbl.htm> visualizado em 25/05/2007.