

# A Practical Approach for Software Engineering Teaching: A Case Study for Real Time System Development

Henrique Fernandes de Campos, Breno Lisi Romano, Gláucia Braga e Silva,  
Ricardo Godoi Vieira and Adilson Marques da Cunha

Divisão de Ciência da Computação (IEC), Instituto Tecnológico de Aeronáutica (ITA),  
12.228-900, São José dos Campos, São Paulo, Brazil

{hfcampos, blromano, glaucia, vieira, cunha}@ita.br

***Abstract.** This paper reports an academic experience at the Instituto Tecnológico de Aeronáutica - ITA in real time embedded system development. The adopted methodology involves tailoring of the IBM Rational Unified Process (RUP) supported by Software Engineering recommended practices. A case study related to a hydrometeorological context is used to demonstrate the application of this methodology together with the main artifacts produced during some development process activities.*

## 1. Introduction

Nowadays, Software Engineering teaching has been improved by increasing practical classes in undergrad and graduate courses in Computer Science areas. Generally, practical activities involve the use of academic projects towards the application of methods, techniques and tools to the software development.

During the second semester of 2007 in the Electronics and Computer Engineering graduate program at the *Instituto Tecnológico de Aeronáutica - ITA*, a case study related to hydrometeorological context was adopted as an academic project for a course named Real Time Embedded Systems (*Sistemas Embarcados de Tempo Real – CE-235*).

This paper tackles a software development methodology and shows the main applied techniques and some Computer Aided Software Engineering (CASE) tools used during the development of process activities in a real time embedded system prototype.

The remaining of this article is organized as follows. The second section shows some related works. The third describes an adopted case study. The development methodology is addressed in the fourth section together with some used techniques and RUP tailoring. The fifth shows its application within a case study and also some results. Finally, the sixth section presents some conclusions.

## 2. Related Works

According to a well succeeded academic experience background, practical activities related with the Software Engineering teaching has been studied at ITA [Cunha 2008]. These experiences explore the benefits obtained at teaching-learning process from the use of technological resources together with teaching methodologies like Problem Based Learning (PBL) [Nobre 2006].

Martins et al. [Martins 2005] and Cunha et al. [Cunha 2007] investigations relate the results from applying PBL methodology within the scope of the CE-235 discipline. The last research addresses the integration strategy by levels with tailoring of the IBM Rational Unified Process (RUP) for small projects and the use of CASE tools.

Other works have been adopted practical-theoretical teaching methodologies which consist of balancing theory and practice on software project development including group activities, CASE tools, software development processes and specific examinations [Andrade 2008].

### **3. Case Study**

The VANT-EC-SAME case study is a Computer Software System (CSS) comprised of: a software for Data Collection Platforms (*Plataformas de Coleta de Dados - PCDs*); a software for a station of hydrometeorological data acquisition, monitoring, controlling, and decision support (*Estação de Monitoramento, Controle e Apoio a Decisão – EMCAD*); and a software for Ecological Monitoring Satellites. This case study was adopted during a 17 weeks course named CE-235 on the second academic semester of 2007 at ITA.

This case study represents an abstraction of a realistic Project named Amazon Integration and Cooperation for Modernization of Hydrological Monitoring (*Projeto de Integração e Cooperação Amazônica para Modernização do Monitoramento Hidrológico - ICA-MMH*). It has been developed in a collaborative work involving ITA and the Brazilian National Water Agency (*Agência Nacional de Águas – ANA*), supported by the Brazilian Projects and Studies Foundation (*Financiadora de Estudos e Projetos - FINEP*), in the *Projeto 5206/06*.

### **4. The Development Methodology**

According to experiences in previous courses, the adoption of methodologies has supported teachers in the management of academic activities related to the software project development.

The adopted methodology follows the best practices on Software Engineering [Pressman 2002], ensuring its software quality as products, processes, and services.

Next sections present its used techniques and a RUP tailoring applied to the VANT-EC-SAME CSS development.

#### **4.1. Used Techniques**

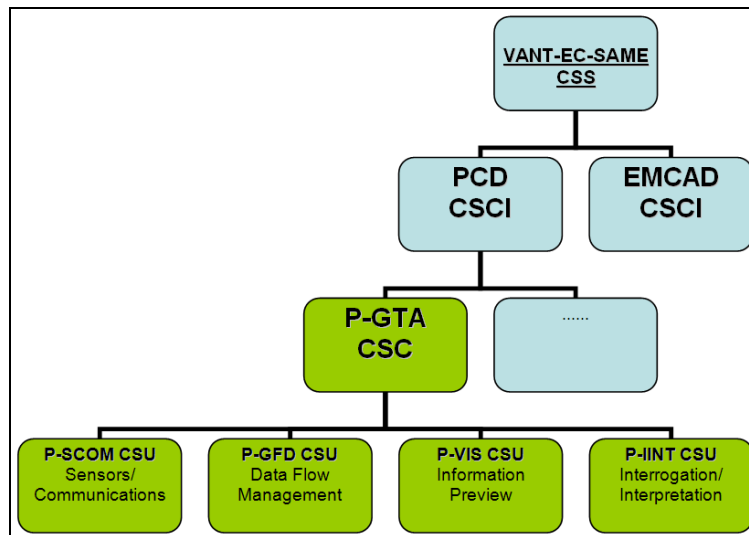
This section shows two fundamental techniques applied into the case study successful development. The first one tackles iterative and incremental development according to some integration levels. While the second technique shows the effort estimation for the development based on use cases.

##### **4.1.1. Integration levels**

The methodology applied to the case study development has divided the project into small applications assigned to each student. Using a bottom-up approach, individual applications were integrated within similar ones. Three other integration levels have

occurred until the prototype was consolidated into a new computerized software system, as shown in Figure 1.

Four integration levels were defined, as suggested by [U.S. Department of Defense 1994]: Computer Software or Hardware Units (CSU/CHU); Computer Software Components (CSC) or Computer Hardware Components (CHC); Computer Software Configuration Items (CSCI) or Computer Hardware Configuration Items (CHCI); and to integrate all the previous ones, just a Computer Software System (CSS).



**Figure 1. VANT-EC-SAME CSS – Integration Levels.**

The VANT-EC-SAME CSS was divided into two CSCI representing the PCD and the EMCAD. Each CSCI was formed from its CSC, which in its turn has generated their respective CSU or has encapsulated into a Computer Hardware Unit – CHU, as shown in Figure 1.

In this context, each student has the opportunity to apply the theoretical concepts into different scenarios: individually on his own application (CSU) in the first integration level; and in teams on the advanced integration levels, where they can share experiences, knowledge, and prepare themselves for their professional life.

Considering the resulting integration levels and its applications, the work breakdown was based on course schedule and students’ technical skills. And, in order to define the scope of each small application and the involved academic tests, an effort estimation technique was used and it will be presented in the next section.

#### **4.1.2. Effort Estimation by Use Case Points (EEUCP)**

The cost estimation technique of use case points was used to evaluate the following factors in order to determine an overall cost estimation:

- 1) Technical implementation factors;
- 2) Environmental factors;
- 3) Use cases’ quantity and complexity; and
- 4) Actors’ quantity and complexity.

The effort estimation of the case study development was established by the EEUCP technique, applied on a free software tool developed at ITA [Silva 2008], providing information to the CE-235 detailed course planning. This planning covers the work balance considering available time and human resources involved.

Other factors also considered on this estimation concern the formal process used and automated tools supporting development process activities.

This technique was applied on the use cases, previously identified on the VANT-EC-SAME CSS, in order to evaluate the Case Study applicability at the available time.

## 4.2. RUP Tailoring

The development process applied to the VANT-EC-SAME CSS was based upon the RUP tailoring to small projects. Figure 2 shows the phases and activities selected from academic customization as well as the involved iterations and artifacts to be produced.

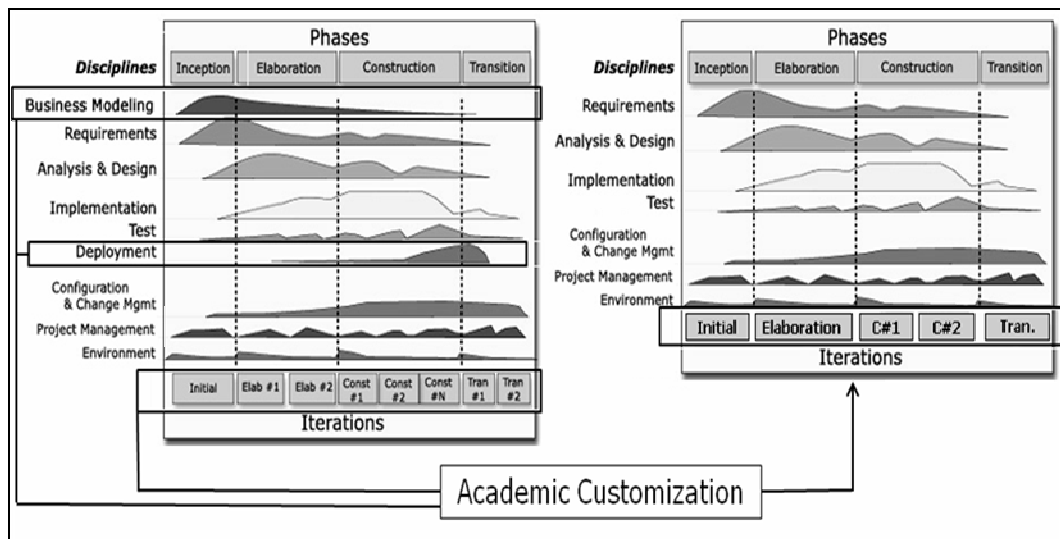
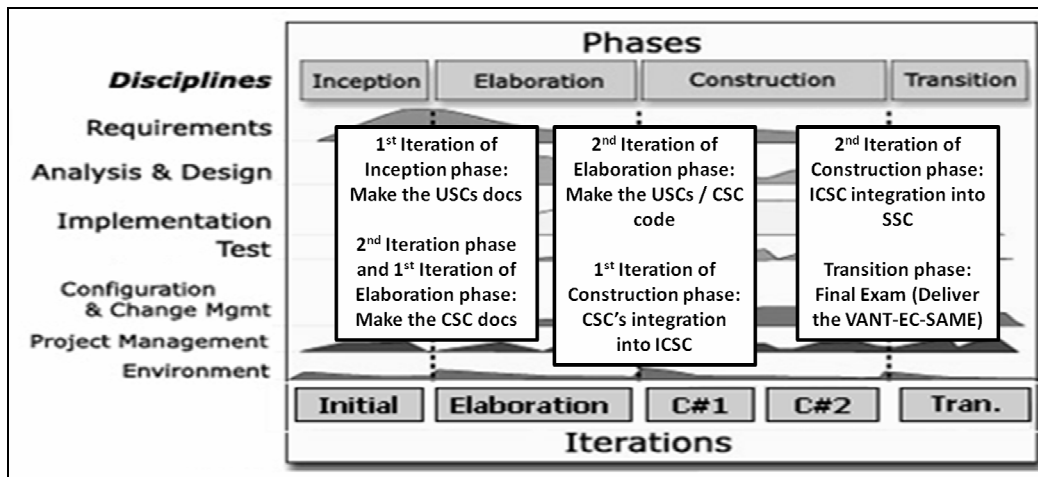


Figure 2. RUP Tailoring.

During the case study development and according to its integration level, students have taken different roles in the process, working as analysts, designers, programmers, testers, among others suggested by RUP.

Regarding artifacts elaboration, a subset of documents, models, source-codes, graphical interfaces and reports were selected and mapped according to RUP phases and iterations. Each RUP phase had at least two iterations, allowing the project incremental elaboration, as shown in Figure 3.



**Figure 3. RUP Iteration.**

A timetable was built using MS Project to represent the Work Breakdown Structure (WBS) based on RUP tailoring, in order to show integration levels and pre-established milestones.

These milestones represent the defined academic tests divided into two types. The first is the theoretical test in which the learnt concepts are checked. In the second, practical exercises include artifacts produced through the iterations of each phase, as shown in Figure 3.

#### 4.3. Tools

During this software development, some CASE tools were used [Santos 1994] in order to support software construction to improve the VANT-EC-SAME CSS quality development (Table 1).

**Table 1. Used tools at the VANT-EC-SAME CSS.**

Discipline	Purpose	Tool
Project Management	Planning and traceability to support project manager.	MS Project
	Effort estimation based on the use case in order to measure the development time.	Estimation Automated Tool
Requirements and Business Modeling	Software requirements management.	IBM Rational RequisitePro
Requirements, Analysis, Design and Business Modeling	RealTime system development based upon the UML to allow source code generation and run the model.	IBM Rational Rose RealTime
Implementation	Source code development	Borland C++ Builder
Testing	Cross-platform solution for component	IBM Rational Test

	testing and runtime analysis.	RealTime
	Software support verification by means of automatic test generation from sequence diagrams.	IBM Rational Quality Architect - RealTime Edition
Configuration Management and Change Management	Version control management.	IBM Rational ClearCase
	Change requests traceability.	IBM Rational ClearQuest
All disciplines	Automatic documents and reports elaboration from developed models and source codes.	IBM Rational Software Documentation Automation

The use of these tools was made possible due to previous academic agreements between the ITA and some companies such as IBM Rational, Microsoft and Borland. On the other hand, the Effort Estimation Automated Tool (EAT) was developed by a researcher currently developing his work at the ITA Software Engineering Research Group (*Grupo de Pesquisa em Engenharia de Software - GPES*) [Silva 2008].

Looking at the major benefits of the use of CASE tools, an increase of productivity and quality was observed due to automated code generation, use of standards, automated tests and documentation of the entire process.

On the other hand, even though the theoretical concepts had been taught through tutorials and quick references, one of the main disadvantages is related to the small period of time available to comprehend the use of the adopted tools.

## 5. Applying the Methodology

In order to demonstrate the proposed methodology applied in the VANT-EC-SAME CSS, based on integration levels (Figure 1), the Tasks Management CSC named *Gerenciamento de Tarefas* (P-GTA), part of PCD CSCI, was selected due to a scope reduction.

The P-GTA CSC has provided the real time hydrometeorological data acquisition from watersheds and also the telecommunication management between the EMCAD CSCI and the PCD CSCI.

On this context, the P-GTA CSC was composed of the following four CSU: Information Preview (*Pré-Visualização de Informações – P-VIS*); Interrogation/ Interpretation (*Interrogação/ Interpretação – P-IINT*); Data Flow Management (*Gerenciamento do Fluxo de Dados – P-GFD*); and Sensors / Communications (*Sensores / Comunicação – P-SCOM*).

The aim of the EMCAD CSCI was to receive hydrometeorological data such as: temperature, atmospheric pressure, and water quality. These data are usually acquired from the PCD CSCI installed on river margins at the Brazilian Amazon Region, and can be transmitted on three different ways: Satellite, Radio Frequency, or Cellular Phone.

P-GTA CSC development was planned according to the EEUCP technique results. Regarding the P-GTA use cases complexity, the cost estimation obtained in terms of time, was about seventeen weeks with a four student's team.

Figure 4 shows an overview of P-GTA CSC functionalities represented by a Unified Modeling Language (UML) use case diagram. Some examples of existing functionalities are sensing data acquisition, communication direction management, and *in loco* acquired data visualization.

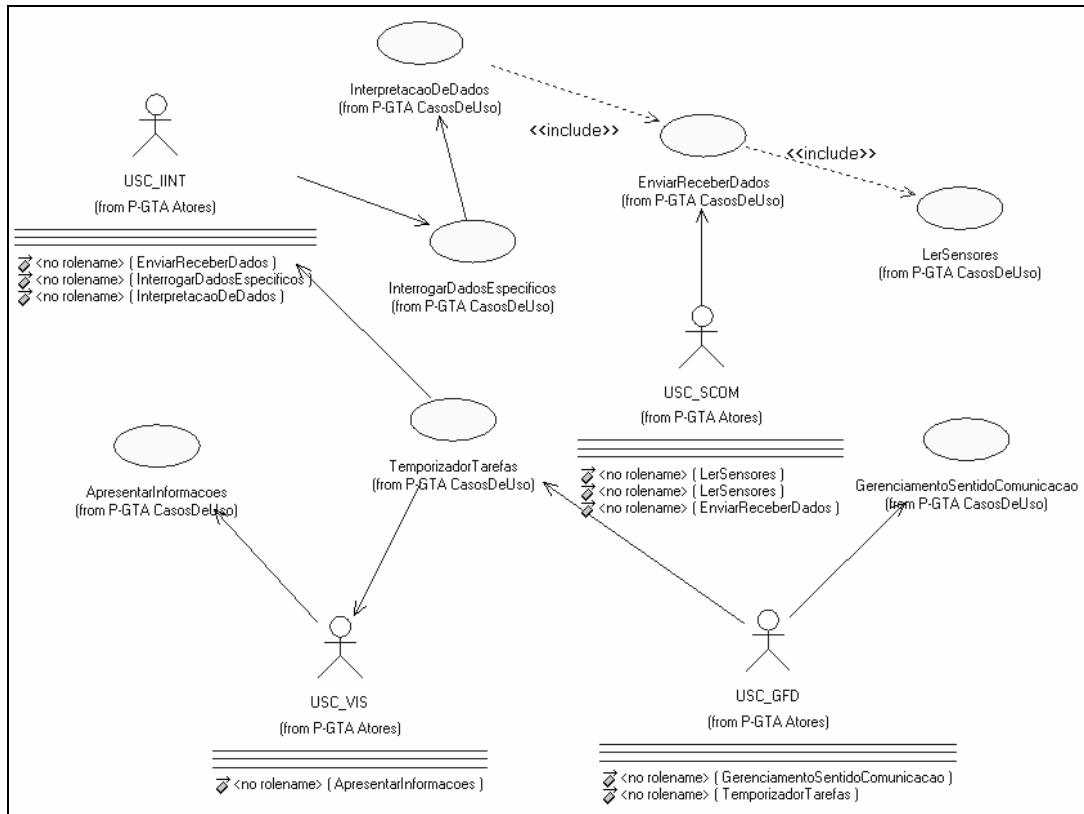


Figure 4. P-GTA CSC Use Case Diagram.

### 5.1. RUP phases iterations

In the first iteration at the RUP Inception phase, functional and non-functional requirements of P-IINT CSU, P-GFD CSU, and P-VIS CSU were surveyed and documented in artifacts, allowing use cases modeling, as shown in Figure 4. In the second iteration of the first RUP phase, the selected CSU artifacts were integrated, increasing the development complexity.

The Rational Rose RealTime (RRRT) CASE tool was used on the RUP Elaboration phase, in order to model the CSUs until the P-GTA CSC RUP Construction phase.

In the RUP Construction phase, the P-GTA CSC and the P-COM CSC were integrated into the PCD CSCI. Their structure and state diagrams were aggregated, in order to facilitate components understanding and visualizations. At this point, classes, capsules, protocols, and associated diagrams were organized into packages. Finally, in

the second iteration of this RUP Construction phase, the P-GTA CSCI and EMCAD CSCI were integrated into the VANT-EC-SAME CSS.

All the related phases above have included the elaboration of the following RUP documentation artifacts: Development Case; Development Plan; Vision; Stakeholders Requests; Supplementary Specifications; Glossary; Risk List; Iteration Plan; Use Case Model; Requirements Management Plan; Test Plan; Use Case Test; and Quality Warranty Plan.

At the end, in the RUP Transition phase, the P-GTA CSC, integrated into the VANT-EC-SAME CSS, was adjusted to fit identified needs such as defect corrections, features implementations, and the official deliver.

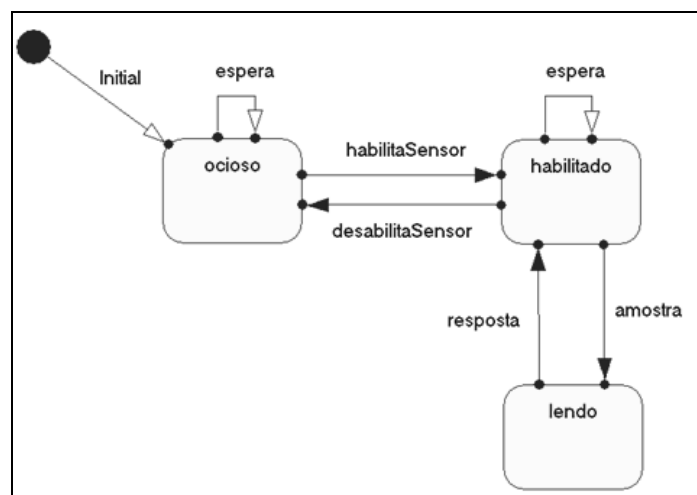
The following section describes P-GTA CSC analysis and design steps.

## 5.2. P-GTA produced artifacts

Within the artifacts set in Elaboration and Construction phases of tailored RUP, the mainly UML logical and physical models were produced supported by CASE tools, according to analysis and design disciplines.

The use case diagram was developed to show all P-GTA CSC functionalities and CSU relationships overview. Class diagrams and appropriate protocols were defined to establish communications between capsules and classes.

Thus, a structure diagram was produced to represent the messages exchange between capsules and classes in the P-GTA CSC class diagram. After that, signals, established ports, and communication direction were defined. On the next step, the P-GTA CSC behavior diagram was developed through states machines, as shown Figure 5.



**Figure 5. P-SCOM behavior diagram.**

The P-GTA CSC Analysis and Design disciplines were improved by applying the RRRT tool in order to support automated code generation, as applied by [Loubach 2008]. In this step, about 15.000 source-code lines in C++ Language were automatically generated. It was produced, based upon modelling and target software platform configuration parameters.



During the Construction phase, the generated code was checked by a test set using the IBM Rational Test RealTime (RTRT) and the IBM Rational Quality Architect - RealTime Edition (RQA-RT).

## 6. Teaching-learning process results

This practical approach for teaching the classes increases productivity and motivation of the students because they realize the importance of methods, techniques and CASE tools in the Software Engineering context and can by analogy use them in real problems.

As an educational method, the proposed methodology requires experience, dedication and available time from the teacher. Three aspects can be highlighted for the application of this methodology: 1) Knowledge and experience to define and plan the Case Study; 2) Leadership to manage students working in teams; e 3) Maturity to evaluate students' participation individually and in teams.

As a continuous improvement of the adopted teaching practices, the teacher receives feedback from students mainly by e-mail or in person, providing suggestions, recommendations and also examples of the use of the same practices at work.

## 7. Conclusion

Nowadays, the success of developing academic case studies directly depends on the use of a methodology and the Software Engineering best practices.

The methodology basis for the case study on this paper was established by applying some known techniques such as integration levels strategy and effort estimation, useful on the activities' planning and management.

The use of case studies as academic projects allows students to experience situations that they could only witness if they were working in a company.

Given the successful history of the presented practical teaching approach, the authors strongly recommend the adoption of this methodology in other areas of knowledge because it enriches the content of classes and increases the motivation, participation, and involvement of students and teachers in the teaching-learning process.

## References

- Andrade, R. M. C., Marinho, F. G., Leitão, V. L., Rocha, L. S. (2008). "*Uma Proposta de Metodologia para o Ensino de Engenharia de Software*", In: Fórum de Educação em Engenharia de Software, Campinas - SP. Monografias em Ciência da Computação - PUC RJ. Rio de Janeiro: Pontifícia Universidade Católica do Rio de Janeiro, v. 43/08.
- Cunha, A.M., Braga e Silva, G. Monte-Mor, J. A., Domiciano, M. A. P., and Vieira, R. G. (2008). "*Estudo de Caso abrangendo o Ensino Interdisciplinar de Engenharia de Software*", In: Fórum de Educação em Engenharia de Software, Campinas - SP. Monografias em Ciência da Computação - PUC RJ. Rio de Janeiro: Pontifícia Universidade Católica do Rio de Janeiro, v. 43/08.

- Cunha, A.M., Santos, W.A., Loubach, D. S., Nascimento, M. R., Nobre, J.C.S. (2007). “Applying model driven development in aerospace software prototype with IBM Rational Rose Real-Time”, In Proceedings of the IBM Rational Software Development Conference, Orlando.
- Loubach, D.S., Ramos, D.B., Saotome, O., Cunha, A.M. (2008). “Comparing source codes generated by case tools with hand coded”, In Proceedings of 5th International Conference on Information Technology (ITNG): New Generations, Las Vegas.
- Martins, O., Santos, W.A., Ferreira, A.S., Anjos, L.S., Cunha, A.M. (2005). “A strategy for teaching real time embedded systems at the Brazilian Aeronautics Institute of Technology – ITA”, In Proceedings of 3rd International Conference on Education and Information Systems (EISTA): Technologies and Applications, Orlando.
- Nobre, J.C.S., Loubach, D.S., Cunha, A.M., and Vieira Dias, L.A. (2006). “Aprendizagem Baseada em Projeto (Project-Based Learning – PBL) aplicada a software embarcado e de tempo real”, SBIE, SBC.
- Pressman, R.S. (2002). “Software Engineering: A Practitioners Approach”, McGraw Hill.
- Santos, W.A. and Cunha, A.M. (2005). “Exploring Round-Trip Engineering Capabilities for Satellite Flight Software Projects” 24th Digital Avionics Systems Conference – DASC.
- Silva, C. M. B. da, Loubach, D. S., Cunha, A. M. da. (2008). “Applying the Use Case Points Effort Estimation Technique to Avionics Systems”, In: 27th IEEE/AIAA Digital Avionics Systems Conference (DASC), Saint Paul. Proceedings of 27th Digital Avionics Systems Conference.
- U.S. Department of Defense. (1994). “Military Standard Software Development and Documentation”, MIL-STD-498.