Using Best Practices of Software Engineering into a Real Time System Development

Ricardo Godoi Vieira, Breno Lisi Romano, Gláucia Braga e Silva, Henrique Fernandes de Campos, Adilson Marques da Cunha
Brazilian Aeronautics Institute of Technology
São José dos Campos, São Paulo, Brazil

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

Key Words: artifacts, CASE tools, effort estimation, integration level, RUP tailoring.

1. Introduction
Nowadays, the Software Engineering teaching has been improved by increasing practical classes in undergraduate 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 of the Brazilian Aeronautics Institute of Technology (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).

In this context this paper tackles a development methodology and shows the main applied techniques and some Computer Aided Software Engineering (CASE) 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. The fifth demonstrates its application into a case study. Finally, the sixth and last 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 [1]. 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) [2].

Martins’ et al. [3] and Cunha’s et al. [4] researches relate the results from applying PBL methodology within the scope of CE-235 discipline. The last research addresses the integration strategy by levels with a tailored IBM Rational Unified Process (RUP) for small projects and the use of CASE tools.

3. Case Study
The VANT-EC-SAME case study is a Computer Software System (CSS) comprised of: software for Data Collection Platform (Plataformas de Coleta de Dados - PCDs); software for hydrometeorological data acquisition, monitoring, controlling, and decision support station (Estação de Monitoramento, Controle e APOio a Decisão – EMCAD); and 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 Amazonônica 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).

4. The Development Methodology

The used development methodology follows the best practices on Software Engineering [5], 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 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 [6]: 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 previous, just one Computer Software System (CSS). 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 (Figure 1).

4.1.2. Effort Estimation by Use Case Points (EEUCP). The cost estimation technique of use case points evaluates 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 EEUCP technique, applied on a free software tool developed at ITA, 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.

4.2. RUP Tailoring

The development process applied to the VANT-EC-SAME CSS was based upon a 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.

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](image)

A timetable was built using MS Project including activities, artifacts and source code to be produced and delivered according to integration levels or pre-established milestones.

4.3. Tools

During this software development some CASE tools were used [7] in order to support software construction to improve the VANT-EC-SAME CSS development quality (Table 1).

<table>
<thead>
<tr>
<th>Discipline</th>
<th>Purpose</th>
<th>Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Management</td>
<td>Planning and traceability to support project manager.</td>
<td>MS Project</td>
</tr>
<tr>
<td></td>
<td>Effort estimation based on the use case in order to measure the development time.</td>
<td>Estimation Automated Tool</td>
</tr>
<tr>
<td>Requirements and Business Modeling</td>
<td>Software requirements management.</td>
<td>IBM Rational RequisitePro</td>
</tr>
<tr>
<td></td>
<td>RealTime system development based upon the UML allows source code generation and runs the model.</td>
<td>IBM Rational Rose RealTime</td>
</tr>
<tr>
<td>Implementation</td>
<td>Source code development</td>
<td>Borland C++ Builder</td>
</tr>
<tr>
<td>Testing</td>
<td>Cross-platform solution for component testing and runtime analysis.</td>
<td>IBM Rational Test RealTime</td>
</tr>
<tr>
<td></td>
<td>Supports software verification by means of the tests automatic generation from sequence diagrams.</td>
<td>IBM Rational Quality Architect - RealTime Edition</td>
</tr>
</tbody>
</table>

| 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 the developed models and source code. | 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 an ITA researcher currently developing his work at the ITA Software Engineering Research Group (Grupo de Pesquisa em Engenharia de Software - GPES).

5. Applying the Methodology

In order to apply the proposed methodology, the Tasks Management CSC named Gerenciamento de Tarefas (P-GTA) was selected. The P-GTA CSC has provided the real time hydrometeorological data acquisition from water masses and also the telecommunication management between the EMCAD CSCI and the PCD CSCI.

In this context, the P-GTA CSC was composed of the following four CSU: Information Preview (Pré-Visualização de Informações – P-VIS); Interrogation and 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. Therefore, the scope was reduced to the Radio Frequency communication due to academic time constraints.

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 sensoring data acquisition, communication direction management, and in loco acquired data visualization.

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 preliminary use cases modeling, as shown in Figure 4. In the second iteration of this first RUP phase, the selected CSU artifacts were integrated, increasing the development complexity.

Finally, in the RUP Transition phase, 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 Analysis and Design

The P-GTA CSC Analysis and Design steps were improved by applying the RRRT tool in order to support automated code generating, as applied by [8]. This time around 15,000 source-code lines were automatically CASE generated.

The use case diagram was produced to represent all P-GTA CSC functionalities and CSU relationships overview. So, a class diagram and appropriate protocols were developed to establish communication between capsules and classes.

Thus, a structure diagram was produced to define the messages exchange between capsules and classes in the P-GTA CSC class diagram. After, signals, established ports, and communication direction were defined.

Next step, the P-GTA CSC behavior diagram was developed through states machines.

Finally, in the RUP Transition phase, the VANT-EC-SAME CSS was adjusted to fit identified needs such as defect corrections, features implementations, and the official deliver.

5.3. Simulation Environment

The Simulation Environment was composed by HCI P-VIS, P-GTA CSC modeling and hardware as shown in Figure 6. This composition was defined due the fact that HCI P-VIS has been developed in Borland C++ Builder.
tool and P-GTA CSC has been modeled in RRRT CASE tool, without a native interface for communication.

So, to provide an interface for communication it was used the Transmission Control Protocol/Internet Protocol (TCP/IP), implemented through sockets. The P-GTA CSC modeling was embedded in hardware with the purpose of emulating data acquisition from sensors and data sending to the HCI P-VIS.

![Figure 6: Simulation Environment Overview](image)

After the communication between the P-GTA CSC and the HCI P-VIS some information such as: air relative humidity, external temperature, and luminosity could be visualized, as shown in Figure 7.

![Figure 7: P-VIS Human-Computer Interface](image)

6. Conclusion

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

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.

Besides applying a methodology into a Real Time Embedded System case study, this paper has shown a way of checking its efficiency and identifying some of its improvements.

At the end, the main gains obtained from the use of automated tools for code generation have justified the dispended effort on the model elaboration.

Some suggested future works are: changing the hardware emulation for an embedded version; and applying this proposed methodology into other case studies.

7. References


