Applying Agile Method on Academic Access and Fraud Control System

Gustavo Ravanhani Matuck, Rodrigo Luiz de Oliveira, Tharcius Augusto Pivetta, Luiz Alberto Vieira Dias, Adilson Marques da Cunha
Computer Science Division
Brazilian Aeronautics Institute of Technology (ITA)
Sao Jose dos Campos - SP - Brazil
gmatuck@ita.br, ruizol20@gmail.com, tpivetta@gmail.com, vdiastita.br, cunha@ita.br

Abstract – This paper presents an application of agile method on academic access and fraud control system. It describes the conceptualization, development, and testing process of a case study, using Problem Based Learning (PBL) and some collaborative tools. It was developed, on the 1st semester of 2012, at the Brazilian Aeronautics Institute of Technology (Instituto Tecnologico de Aeronautica – ITA) by graduate students during three different multidisciplinary courses involving geographically distributed teams. Its main results are presented, discussed, and analyzed, providing an overview of a successful new academic application development.

Keywords - Model-based software development, Scrum agile development, Problem-based learning – PBL, Collaborative tools, Interdisciplinary teams, Banking fraud transactions.

I. INTRODUCTION

Nowadays, the competition between countries means that the human knowledge areas seek for innovative and complex solutions that meet the needs of an increasingly globalized market [1]. The changing world of Information Technology (IT) has represented the main foundation for supporting various knowledge domains [2].

On the academic arena, the main process of teaching and learning presents difficulties and involves political, social, and cultural challenging elements, among others. The dynamics of a modern world requires closer relationships between academic teaching and learning processes and professional market needs [3].

On the financial arena, the global competition has focused on customers and results, taking into account mainly banking fraud and access controls [4]. This has been causing relevant investments from financial institutions and their partners on advanced technologies and capabilities. And it is getting worst in so far, as emerging technologies to support financial areas have been also serving to support and give power to bad intentions of fraudsters mainly involving banking transaction operations.

On both educational and financial fields, Information Technology (IT) based project developments have been looking for solutions that increasingly meet stringent demands by actual society.

However, the traditional methods for software development have not been able to meet the needs with required efficiency [5,6].

In order to meet customers’ needs, agile projects’ developments are gaining strength in all modern society sectors, as news solutions to solve old problems [7].

Other important elements guiding the IT field and software projects developments have been using resources involving geographically distributed multidisciplinary teams [7].

Within this context, communications among individuals are becoming essential to ensure desired results and mitigate obstacles that make projects unfeasible or affect customers’ expectations.

The use of collaborative tools, mainly through the Internet, has been providing an efficient way to increase communication and productivity. This scenario has eliminated barriers that until then have been impacting software projects’ developments.

Considering this scenario, some research areas still need to be investigated:

- The educational area – needs to be properly structured to provide better ways of meeting market needs;
- The Financial area – needs initiatives involving Research & Development (R&D) to provide more appropriate solutions for banking fraud and access controls and preventions; and
- The IT area – needs more suitable new strategies for software projects’ developments than the previous traditional prescriptive methods applied to address customers’ expectations.

Collaborations among geographically distributed multidisciplinary teams need support based on collaborative tools, aiming to increase productivity. And reduce waste of resources.
II. WORK STRUCTURE

This research aims to investigate, conceive, implement, and test an academic system prototype for a Banking Fraud and Access Control System, named in Portuguese Sistema de Controle de Acessos e Fraudes - SCAF, by applying Scrum agile method development, its best practices and some collaborative tools.

This prototype was developed, during the 1st semester of 2012, by students of three different courses from the Electronics Engineering and Computer Science Graduate Program at the Brazilian Aeronautics Institute of Technology (Instituto Tecnologico de Aeronautica - ITA): CE-240 Database System Project, CE-245 Information Technologies, and CE-229 Software Testing.

The developed prototype tackled the main pros and cons of academically using the Scrum agile method, its best practices, and some collaborative tools.

III. DEVELOPMENT AND MAIN RESULTS

Initially, a Database structure, some User Stories (USs), and Students responsible for the development of the SCAF prototype were divided into two Corporate Database Systems teams: the first team was assigned to develop a Corporate Database Subsystem for Access Control (Subsistema de Controle de Acessos Indevidos – SCAI) and the second team was assigned to develop a Corporate Database Subsystem for Fraud Operating Control (Subsistema de Controle Operacional de Fraudes – SCOF).

As expected, the initial idea of this work has evolved several times, during the academic semester and its project development. The two Corporate Database Subsystems (SCAI and SCOF) were subdivided again into four Subject Database Subsystems.

Figure 1 shows the SCAF academic project prototype overview with the SCAI Corporate Database subdivided into GIC and AAI Subject Database Subsystems and the SCOF Corporate Database subdivided into GDI and AAI Subject Database Subsystems.

Within this context, each Subject Database Subsystem was again subdivided into four Application Database Subsystems, each one of them was individually assigned to just one student from the CE-240 Database System Project Course.

CE-240 The Database System Project Course

At the end of each sprint development, each designed and implemented Database Subsystem had to show compliance with some specification requirements. This process was conducted according to four integration levels: 1) the Application Database Subsystem Level, also called “zero integration level”; 2) the Subject Database Subsystem Level, also called “1st integration level”; 3) the Corporate Database Subsystem Level, also called “2nd integration level”; and 4) the Holding Database System Level, also called “3rd and final integration level”.

1) The Application Database Subsystem Level

This integration level was also called zero integration level, mainly because no integration effort had to be done at this time. In order to build Application Database Subsystems, only individual development effort was needed from each student team member.

This level brought together the minimum necessary amount of data and/or information to met requirements.

At this zero integration level, there was no need of direct dialogue among team members from other Application Database Subsystems being constructed.

2) The Subject Database Subsystem Level

A process of initial re-work and fusion of some database entities and its attributes characterized this 1st integration level.

At this time, common identified data and information from different Application Databases Subsystems belonging to the same Subject Database Subsystem had to be integrated and layered in order to avoid redundancy.

The starting point of simultaneous information exchanges and collaborations characterized this process, when the database development had to be parallelized within the four Subject Database Subsystems.

Figure 1 shows this 1st integration level with the parallelization development process per each Corporate Database Subsystem (SCAI and SCOF).

For the SCAI Corporate Database Subsystem, within its GIC Subject Database Subsystem, the following Application Database Subsystems were simultaneously integrated in parallel: GVS, GDI, GDP, and GSV.

Also for the SCAI Corporate Database Subsystem, within its AAI Subject Database Subsystem, the following Application Database Subsystems were simultaneously integrated in parallel: GDM, GRI, GMW, and GRA.
For the SCOF Corporate Database Subsystem, within its GDI Subject Database Subsystem, the following Application Database Subsystems were simultaneously integrated in parallel: GPC, GDA, GDH, and GDF.

Also for the SCOF Corporate Database Subsystem, within its AAI Subject Database Subsystem, the following Application Database Subsystems were simultaneously integrated in parallel: GCD, GPD, GMC, and RDO.

Figure 2 shows the SCAF Holding Database System Entity-relationship Model.

3) The Corporate Database Subsystem Level
A re-working and fusion continuation process of some database entities and its attributes from different Subject Database Subsystems has characterized this 2nd integration level, aiming to obtain the necessary elements to result in a database’s corporation.

This integration level demanded lower effort than the previous one, since the Subject Database Subsystems were already well structured.

At the end of this corporate level of integration, it was possible to release two Corporate Database Subsystem (SCAI and SCOF), as working prototypes to be tested by the Application Program Interface (API) Development Team.

The API Development Team was composed by the students from the CE-245 Information Technologies Course.

Students from the CE-240 Database System Project Course have been working on the Persistence Layer (Database Layer) and also on the 1st half of the Business Layer of the entire SCAF System Project.

At the same time and in parallel, students from the CE-245 Information Technologies Course have been working on the 2nd part of the Business Layer and on the Interface Layer of the entire SCAF System Project.

This strategy has allowed foreseeing what would be in near future the final aspect of the SCAF project academic prototype.

Figure 3 shows SCAI and SCOF Corporate Database Subsystems after the 2nd integration level.

4) The Holding Database System Level
Finally, the 3rd integration level was performed representing the final modeling prototype of the SCAF project database.

The final transition from SCAI and SCOF Corporate Database Subsystems to the SCAF Holding Database System has required a major academic effort, due to the needs of fusions of some entities to be properly integrated.

Figure 4 shows the SCAI and SCOF Corporate Database Subsystems Models after the 2nd integration level and Figure 5 shows the final result of the SCAF Holding Database System Model after the 3rd integration level.
Table 1 summarizes the main figures of the SCAF Holding Database System Project development from its integration levels.

<table>
<thead>
<tr>
<th>Levels</th>
<th>Entity</th>
<th>Attributes</th>
<th>Source-code Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application DB</td>
<td>70</td>
<td>315</td>
<td>1,489</td>
</tr>
<tr>
<td>Subject DB</td>
<td>47</td>
<td>214</td>
<td>1,368</td>
</tr>
<tr>
<td>Corporate DB</td>
<td>45</td>
<td>203</td>
<td>1,243</td>
</tr>
<tr>
<td>Holding DB</td>
<td>38</td>
<td>174</td>
<td>1,072</td>
</tr>
</tbody>
</table>

CE-245 The Information Technologies Course

Students from CE-245 The Information Technologies Course were responsible for building the Application Program Interface (API) layer as the 2nd part of the SCAF Project Business Layer.

The SCAF API was built based upon Scrum Agile Method and its best practices, considering User Stories (USs) from the Subject Database Subsystems Models, as part of the 1st integration level.

This process was performed using the agile development management tool named Pivotal Tracker® [9]. On this scenario, it was possible to continuously develop and manage through the Internet this academic project, in an appropriate asynchronous and flexible way, reallocating responsibilities to team members as needed.

Figure 6 shows a screen shot of the SCAF project using Scrum Agile Method and Pivotal Tracker® tool.

During the SCAF project development, the most frequent interactions between team members from CE-245 and CE-240 courses on the SCAF have occurred only after the 1st integration level.

The process intensification was carried out mainly because, at the Application Database Level (zero integration level), there was not enough visibility yet of all product features and how they would integrate with each other.

Changes made from Database Subsystems Models have to be also replicated in the SCAF Graphic User Interface (GUI). At this time, team members had to actively increase their communications to perform its appropriate necessary changes.

An Internet portal was built, as a simple and visible way, to create access interface to databases.

Figure 7 – A SCAF Project screen shot with clustering information about frauds operations on credit card.
Data and information intersections were conducted in order to detect credit card frauds’ operations, providing several views and interpretations, aiming to facilitate the definition of: risk areas; establishments with high fraud levels; among others.

Figure 7 provides a screen shot of the SCAF Project. It shows an example of frauds operations on credit card, by clustering information about a risk area integrated with Google Maps® tool [10].

**The CE-229 Software Testing Course**

The software testing team members from the CE-229 course started their activities together with the IT team members from the CE-245 course and the CE-240 team members, as the main database developers.

The testing team members were responsible for building a Test Plan for each previously defined Corporate Database Subsystem (SCAI and SCOF), in a collaborative way, during some planning meetings.

The two Corporate Test Plans aimed to gather all necessary information for the SCAF project prototype overall test planning, describing their test approach.

A top-level Test Plan was generated and used to coordinate all test level efforts, considering the following: Items to be tested; Risks that may affect Test Plan realization; Features to be tested; Features NOT to be tested; Testing approach to be used; Acceptance/failure criteria; Input/output criteria; Cutting and returning requirements criteria; List of resulting deliverables from Test Plan; Test remaining tasks; Environment needs; Staffing and training needs; Team members responsibilities on Test Plan; Planning and contingency risks; and Test Plan glossary.

The testing activity has resulted in reduction or elimination of risks that software might suffer during its development. Thus, the risks analysis has become a primary activity in the Test Plan phase of this SCAF project. Figure 8 shows an example of the SCAF project top-level Test Plan.

During the Test Plan elaboration, testing team members have identified, as a potential difficulty, the lack of pattern and the best practices for source-code development, considering a scenario involving different team members geographically distributed.

To reduce the effects of this problem, the Model View Control (MVC) architecture [8] was used, mainly because it was well known by almost all developer team members and provided the test division in layers.

The MVC architecture has been used in Software Engineering as a development reference model also known as a “pattern architecture”. This model has isolated the SCAF project application logics from user interfaces, allowing development, edition, and testing each part of its development.

Figure 9 shows an example of an MVC diagram with solid lines meaning direct associations and dashed lines meaning indirect associations.

![Figure 9 – An Example of an MVC diagram.](image)

In this scenario, by using MVC architectures, the proposed approach facilitated the directories structuring, benefiting the integration occurrence in development time not just at the end of the project. Therefore, the project directory structure was created according to the Corporate Database Subsystems subdivision and the team members integrating the SCAF project workgroups.

Initially, some directories were considered as a project pattern and were shared with other modules. However, each module had its individuality, according to business rules, shared or not with other modules.

At this time, the individual access permissions had to be assigned for each team member of workgroups, to avoid unauthorized source-code changes of each other.

To ensure the code integrity produced, it was adopted as a solution the use of Subversion (SVN), which has allowed user’s identification and their work operations.

![Figure 10 – A fragment of SCAF project directory.](image)
Figure 10 shows a fragment of the SCAF project directory structure.

After some development cycles and the Test Plan execution, it was created a Test Iteration Report. This report referred as SCAF project modules, aimed to gather all information regarding planning, effort control, and testing executions.

This artifact has described a software test approach. It represents a planning used by some involved team members to coordinate the testing efforts.

The elaboration of this Test Iteration Report has supported the following goals:
- Identification of inspected items by tests;
- Approach test description used;
- Elements description deliverable by test project; and
- Results presentation of test runs.

This Test Iteration Report has considered 25 test cases, sub-divided into 40 cases. From these 40 cases, 21 have represented applications of Database Tests (TDB) and 19 integration tests between layers (TIC). Table 2 shows some general test results conducted for the SCAF project.

<table>
<thead>
<tr>
<th></th>
<th>Success</th>
<th>Failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD</td>
<td>16 (76.19 %)</td>
<td>5 (23.81 %)</td>
</tr>
<tr>
<td>TIC</td>
<td>6 (31.58 %)</td>
<td>13 (68.42 %)</td>
</tr>
</tbody>
</table>

IV. CONCLUSIONS AND RECOMMENDATIONS

This paper presented an application of Scrum agile method and collaborative tools on an academic access and fraud control system, named in Portuguese, Sistema de Controle de Acessos e Fraudes - SCAF.

It described the conceptualization, development, and testing process of a case study, using Problem Based Learning (PBL) and some collaborative tools.

The application of Scrum agile method and its best practices on the SCAF project management has provided a better flexibility towards involved team members as well as better visibility for its end users.

During the SCAF project development, team members could not only fit the scope of requested changes but also overcome most encountered difficulties.

Comparing with traditional software development projects, at the end of this SCAF project, it was possible to deliver more aggregated value to its end-users, as well as provide greater interactivity between all its involved team members.

The use of collaborative tools during the SCAF project prototype development has provided better effectiveness, mainly due to the use of distributed teams.

The integration among the team members from three different graduate courses from the Brazilian Aeronautics Institute of Technology (ITA) has shown an interesting academic challenge and, at same time, an experience with great aggregated value.

The effort for synchronizing all academic activities related to the SCAF project prototype has demanded an extra attention, in order to meet the schedule, the initially defined scope, students iterations, and teacher’s availability.

The main obtained results from the development of the SCAF project have provided a better understanding about credit cards banking frauds and access controls at corporate level, as well as the conceptualization and development of computational methods and techniques directed to a real and complex problem.

For the continuation of this SCAF project prototype, it is recommended that some research should be done, involving application of agile methods and best practices, and new database, programming and testing techniques.

ACKNOWLEDGMENTS

The authors thank the Brazilian Aeronautics Institute of Technology (Instituto Tecnologico de Aeronautica - ITA), its Research Group of Software Engineering (Grupo de Pesquisa em Engenharia de Software - GPES), and the 2RP Net Brazilian Enterprise, for their contributions, cooperation, and research support.

REFERENCES