SOFTWARE DEVELOPMENT PROCESS AT LEI-ISEP

Angelo Martins

Instituto Superior de Engenharia do Porto, Portugal

Alberto Sampaio

Instituto Superior de Engenharia do Porto, Portugal

António Costa

Instituto Superior de Engenharia do Porto, Portugal

ABSTRACT

The software development process is the backbone of any Informatics Engineering program. This paper presents the LEI-ISEP (Informatics Engineering program at Instituto Superior de Engenharia do Porto) software development process, based on the well known Unified Process, and the way it is iteratively and incrementally introduced to the students over 5 semesters.

KEYWORDS

Unified Process, CDIO, RUP, Software Process

INTRODUCTION

A key requirement of CDIO is the students' ability to autonomously conceive and implement systems, using industry best practices. To this effect, several engineering Programs have defined process models to be used in project courses (e.g. Linkönping Project Management Model – LIPS [1]).

A small market (employer) inquest proved that the lack software development process knowledge and practice was the most common weak spot of LEI-ISEP's graduates, as well as the graduates from every other major university in the North of Portugal (the inquest was restricted to the North of Portugal). Thus, the Informatics Engineering Department decided to define as a priority the teaching of an iterative software/system development process in LEI-ISEP, which should be used by all courses.

LEI-ISEP PROGRAM AND THE CDIO APPROACH

LEI-ISEP is a 6 semester bachelor program in Informatics Engineering, a common type of program in Southern Europe that combines computer science and computer engineering. LEI-ISEP adopted the CDIO principles [2] as the reference for the learning process, notably the introduction of a laboratory-project course in each semester and a capstone project in the last semester. About 85% of the students opt to implement the capstone project in an outside organization/enterprise.

Capitalizing on the experience of DTU (Technical University of Denmark), the semester at LEI-ISEP is divided in two time slots:

- Week 1 to 12, the students have classes of 4 or 5 "common/classical" courses. Most courses include have a final exam (ranging from 40 to 60% of the final grade) and assessment by tests, small projects, etc.

- Week 13 to 16, the students have only a laboratory-project course in which they have to implement a "large" project from a given set of requirements (the project complexity increases from semester to semester). The project to implement is related to the subjects learned on the first 12 weeks, thus allowing the students to reflectively improve their knowledge on the subjects.

The rationale behind this spiral approach is to present the student increasingly difficult challenges that he has to solve as a member of a team, using the competences acquired in ordinary courses and in the previous projects. One can define three levels:

- Application development, encompassing the 1st and 2nd semester, where general introductory subjects are introduced: Math and Programming. In the 2nd semester, a specialized course in Software Engineering provides the students with the competences in the software development process needed to step to the next level.

- System development, encompassing the 3rd and 4th semesters. In this intermediate level, core informatics engineering subjects are introduced. These subjects, which any professional in this field must master, include operation systems and networks, advanced OO modeling and algorithms, data base programming and software development. The students are also introduced to the system concept, which is composed of several components, and introduces new concepts such as component integration, component redesign/replacement, system reliability and individual component failure, etc.

- Project management, encompassing the 5th and 6th semesters. In this advanced level, specialized informatics engineering and management (e.g. enterprise management, project management, etc.) subjects are introduced. The software and computing worlds change at a fast pace and some courses have to be updated almost yearly, in order for the students to be able to have successful capstone project/internship.

The learning process of LEI-ISEP's first 5 semesters is summarized in Figure 1. It's worth mentioning that each LAPR (Laboratory-Project) course has a soft skills module (16 hours/semester) focusing on complementary subjects such as writing skills, teamwork, etc. The students are allocated a lab and there is always a teacher present to monitor the work and provide help. Thus more hours allocated to the initial LAPR, while in the last ones the students are expected to develop their project almost autonomously. Actually, in LAPR5 the teacher's role is mostly to act as the client. The "official" classroom hours of the actual project development component of each LAPR ranges between 24 hours/week (LAPR1 and LAPR2) and 16 hours/week (LAPR4 and LAPR5) over 4 weeks. This is enough for the students to implement the first year's projects, but the actual workload in the other LAPR courses is much higher.

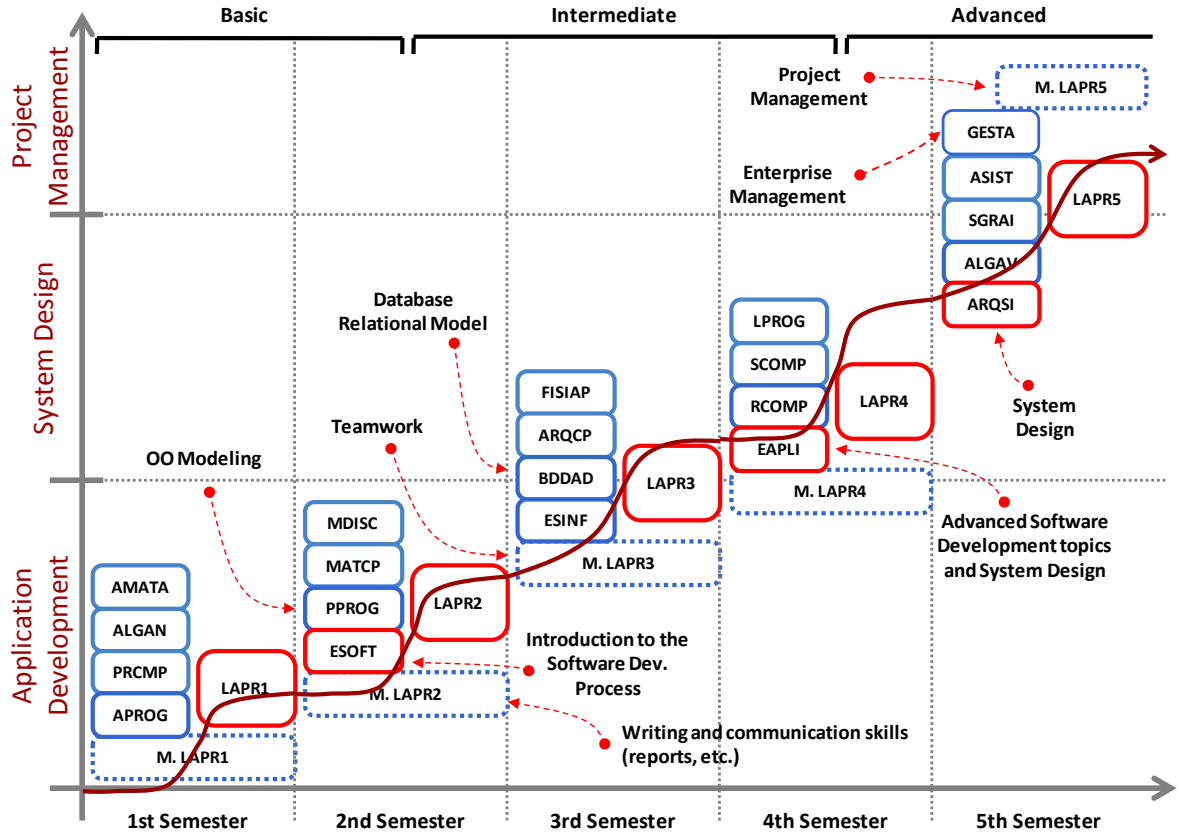

Figure 1 - LEI-ISEP learning process

SOFTWARE DEVELOPMENT PROCESS

The IEEE Computer Society defines software engineering as “the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.” [3]

While many advances have been made in the computer science and software engineering fields, general software practice still produces insecure, unsafe, or unusable programs that are delivered late and over budget. Today, in spite of the best efforts of the computer science and software engineering communities, software developers do not regularly use the best available known methods. Many follow a largely intuitive craft-like practice by picking up new methods from their peers and by using techniques that have long been discredited.

A software development process is a structure imposed on the development of a software product/system. There are several process models, each describing approaches to a variety of tasks or activities that take place during the process [4]. There are many processes and one cannot definitely choose one as the best, as the choice of the process to use is very much dependent on the type of system to be developed, the tools used in its development and the size and internal organization of the development team.

One widely used development process is the Unified Process (UP) [5], an extensible framework for the iterative and incremental development of systems that can be customized for specific organizations or projects. With iterative development, the project is divided into small parts. This allows the development team to demonstrate results earlier on in the process and obtain valuable feedback from system users. Often, each iteration is actually a mini sequential process with the feedback from one phase providing vital information for the

design of the next phase. There are many commercial development processes based and/or inspired on the UP, being the Rational Unified Process (RUP) [6] one of the most widely used. OpenUP [7], from the Eclipse Foundation is a lightweight/agile version of the UP that can be used by small teams and in small projects.

There was an internal argument on the choosing of software development process to use at LEI-ISEP, being the faculty divided between an iterative model (UP based) or an agile one (Scrum, XP, etc.). In the end, a UP based process was chosen, because it provides a structured approach to software development, which may be easier for the beginner (student) to follow. RUP was the processes of choice, as it is a refinement of the UP and the Informatics Engineering Department (DEI) has a close relation with IBM (RUP is a product of Rational, a division of IBM), having access to IBM's catalogue of teaching and e-learning materials. This is paramount in order to train the faculty and thus speed the adoption of the development process.

RUP is based on a set of building blocks, describing what is to be produced, the skills required and the step-by-step explanation describing how specific development goals are achieved. The main building blocks are:

- Roles (who) – A Role defines a set of related skills, competences, and responsibilities.

- Work Products (what) – A Work Product represents something resulting from a task, including all the documents and models produced while working through the process.

- Tasks (how) – A Task describes a unit of work assigned to a Role that provides a meaningful result.

A project is implemented in several iterations and, within each iteration, the tasks are categorized into nine disciplines, six engineering disciplines (Business Modeling, Requirements, Analysis and Design, Implementation, Test, Deployment) and three supporting disciplines (Configuration and Change Management, Project Management, Environment). The relative importance of each discipline changes from iteration to iteration, as the project moves from design to deployment. RUP has determined a project lifecycle consisting of four phases, as depicted in Figure 2:

- Inception - The primary objective is to scope the system adequately as a basis for validating initial costing and budgets.

- Elaboration - In this phase the problem domain analysis is made and the architecture of the project gets its basic form. The key domain analysis for the elaboration is system architecture.

- Construction - In this phase, the main focus goes to the development of components and other features of the system being designed.

- Transition - The primary objective is to 'transition' the system from the development into production, making it available to and understood by the end user. The activities of this phase include training of the end users and maintainers and beta testing of the system to validate it against the end users' expectations.

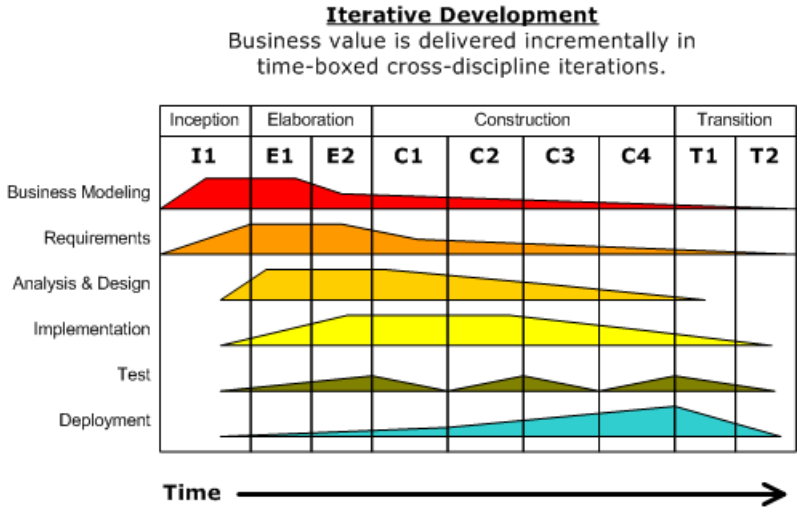

Figure 2 – UP Iterative development and the 6 main disciplines

LEARNING THE SOFTWARE DEVELOPMENT PROCESSES IN LEI-ISEP

RUP is a very large framework and many disciplines are too complex to be included in a bachelor degree, thus it was decided that some disciplines should only be studied in depth on a master degree. Furthermore, it was decided that the proficiency level the students should attain on the core disciplines of the process should be somewhere between Analysis and Synthesis (Bloom's scale) and that they should use the process on their capstone project. This level of proficiency requires a prolonged and repetitive use of the process in several courses, especially each semester's LAPR course. The solution was the development of an iterative approach for the teaching of the software development process, encompassing five semesters and three proficiency levels, compatible with LEI-ISEP's learning process depicted in Figure 1.

Basic Process

The Basic Process is to be used only in the 1st semester's project course, LAPR1, which is designed to meet CDIO standard 4 - Introduction to Engineering, and in the first 12 weeks of the 2nd semester. This process provides an overview of the iterative software development process at a level that can be understood by 1st year students with little knowledge about software development. There is no explicit reference to RUP's disciplines, as it is depicted in Figure 3.

Intermediate Process

The Intermediate Process is to be used in the 2nd (after week 12) and 3rd semesters. The students are introduced to the RUP in the 2nd semester, in a course named "Software Engineering" (ESOFT), thus this process includes four RUP disciplines. Of the two left out, Business Modeling is somewhat included in the Advanced Process, while Deployment is not covered at all in LEI-ISEP.

While the Basic Process is supposed to be used only with the very simple of the problems, where requirements are clearly identified in the problems wording, the Intermediate Process is to be used in more demanding projects, thus the inclusion of the Requirements and Analysis and Design disciplines. Though the process is the same for two semesters, 3rd semester's projects (LAPR3) are naturally more demanding and complex.

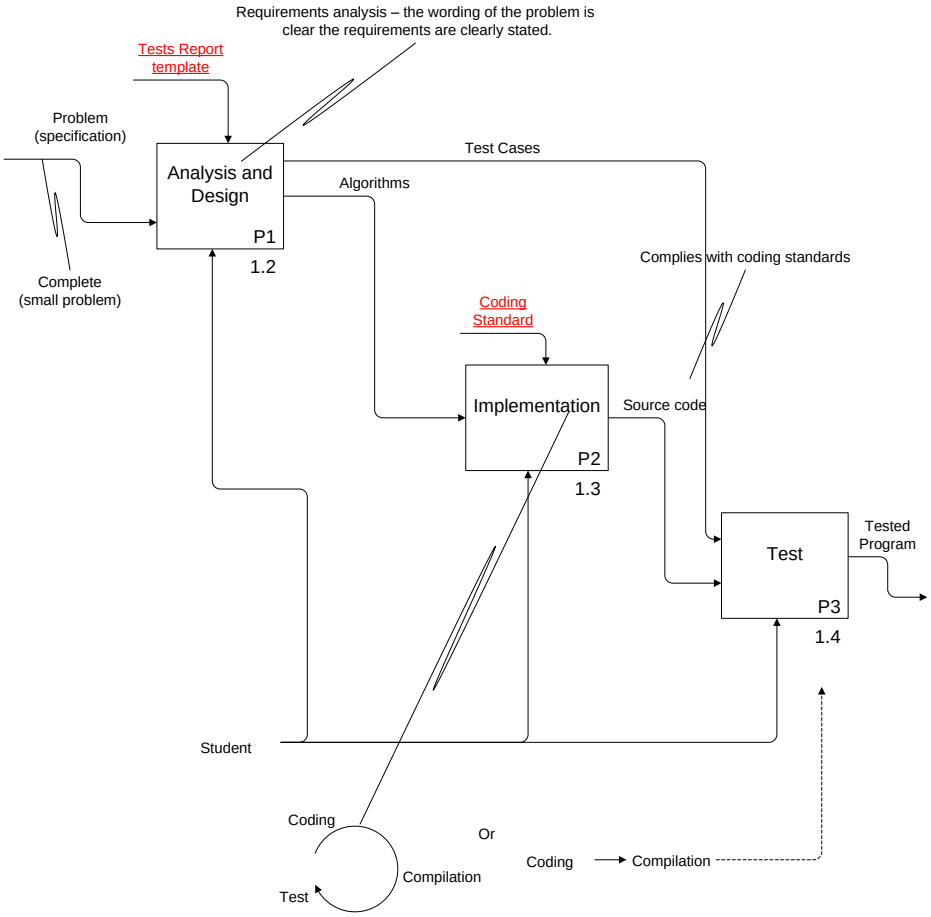

Figure 3 – LEI-ISEP's Basic Process

Advanced Process

The Advanced Process is to be used from the 4th semester (after week 12) onwards, including in the capstone project (6th semester). At the moment there are actually two versions of this process, as Project Management is only introduced in the 5th semester, in LAPR5. This is going to change in the near future, as project Management will be moved to LAPR4 (4th semester).

The advanced development process topics are introduced in two courses, EAPLI (Advanced Software Engineering) in the 4th semester and ARQSI (System Engineering) in the 5th. This iterative and incremental topic introduction, combined with a project course each semester, allows the student to fully understand (reflective observation [8]) the importance of the software development process in his professional life. We believe ISEP's graduates are starting to be agents of change in the Portuguese software industry and the industry seems to recognize that: this school year there were 230 internship offers by the industry while there were only 140 eligible students.

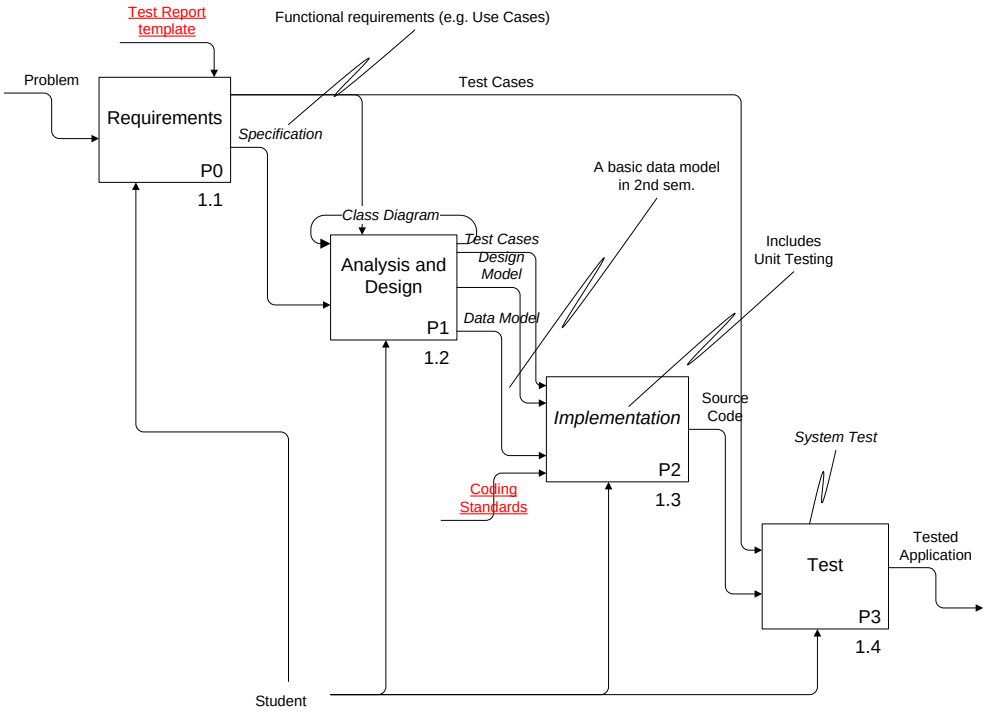

Figure 4 – LEI-ISEP's Intermediate Process

Faculty training

Faculty training is a key success factor for this endeavor. Unfortunately, the software development process hasn't been a popular research topic among the faculty. ISEP has to renowned research centers in the computing area, one on Artificial Intelligence (agents, decision support systems, etc.) and another one on Real-time Systems and Networks. Hence, most masters and PhDs have focused on these topics.

The lack of alignment between faculty interests and the LEI-ISEP Program's it's not exclusive to ISEP. The broader the theme the worse it is, as research nowadays is much focused and results oriented. The software development process is a broad subject and it requires a lot of practice in real projects. The inspiration for a fast solution for this problem come from the six-sigma business management strategy, especially the martial arts inspired implementation roles [9]. Under the direction of the LEI-ISEP's Program director, a *Champion* was chosen to lead the process of defining a software process and a task force was assembled (black belts in six-sigma terminology), composed of specialists in the field, most of them responsible for the courses most related to the subject.

The task force developed a development process proposal and some of the related support documentation (document templates, usage guides, etc.). The process was then introduced in the core courses: the three software engineering related courses and the five project courses. These last courses present an interesting challenge, as they include teaching staff from all other courses of the same semester. In some courses there are over 30 teachers involved. The only solution to disseminate the process practices in an efficient manner was the creation of an intermediate level of responsibility, the "green belts" that were trained and act as disseminators and "experts" for the remaining staff. At the time of the writing of this paper it is too soon to present a reliable assessment of the enforcement of this strategy. Nevertheless, the first results are encouraging.

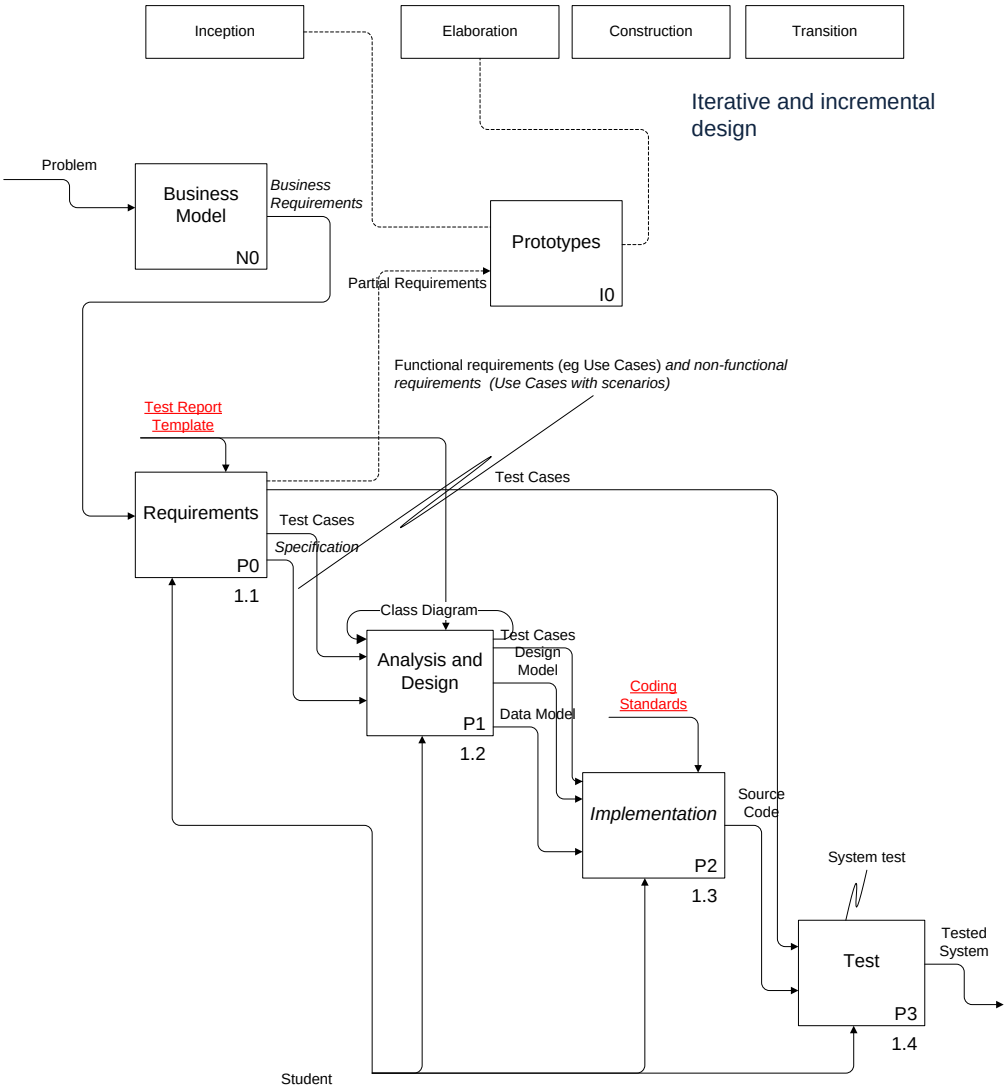

Figure 5 – LEI-ISEP's Advanced Process

CONCLUSION

ISEP identified that the practice of a Software Development Process would be a key differentiator for its graduates in Informatics Engineering, a program already recognized to be in the top 5 in Portugal. Nevertheless, learning a Software Development Process is a prolonged process by itself. This lead to the development of an iterative and incremental approach for introducing the process to LEI-ISEP's students.

The results of this process will only be fully measurable in a couple of years, when current students finish their bachelor degrees and we start having feedback from the industry, but the process has already provided very interesting results, especially in faculty involvement and training.

REFERENCES

[1] Edward Crawley, et al, Rethinking Engineering Education – The CDIO Approach, Springer, 2007, ISBN 978-0-387-38287-6, pp 111.

[2] Edward Crawley, et al, Rethinking Engineering Education – The CDIO Approach, Springer, 2007, ISBN 978-0-387-38287-6, pp 35.

[3] IEEE Standard Glossary of Software Engineering Terminology, IEEE, std 610.12-1990, 1990.

[4] http://en.wikipedia.org/wiki/Software_development_process

[5] Ivar Jacobson, Grady Booch and James Rumbaugh, Unified Software Development Process, Addison-Wesley, 1999, ISBN 0-201-57169-2.

[6] http://www-01.ibm.com/software/awdtools/rup/

[7] http://epf.eclipse.org/wikis/openup/index.htm

[8] Edward Crawley, et al, Rethinking Engineering Education – The CDIO Approach, Springer, 2007, ISBN 978-0-387-38287-6, pp 144.

[9] George Eckes, Six Sigma Team Dynamics - The Elusive Key to Project Success, John Wiley & Sons, 2003. ISBN 0-471-22277-1

Acknowledgements

The authors would like to thank the members of the task force responsible the process development, whose effort was essential for the success of this endeavor:

- Alexandre Bragança (atb@isep.ipp.pt)
- José Miguel Losa (jal@isep.ipp.pt)
- Nuno Silva (nps@isep.ipp.pt)
- Paula Escudeiro (pmo@isep.ipp.pt)
- Paulo Sousa (pag@isep.ipp.pt)

Biographical Information

Angelo Martins is the Program Director of LEI-ISEP at the Informatics Engineering Department of Instituto Superior de Engenharia do Porto, Portugal. He is a Telecommunications and Computer Engineer and has a PhD in Software Engineering. His main research activities are focused on the development of software for large scale health monitoring systems.

Alberto Sampaio is a Professor at the Informatics Engineering Department of Instituto Superior de Engenharia do Porto, Portugal. His current research focuses software process assessment and improvement. He leads the task force responsible for the development of LEI-ISPE's Software Development Process.

Antonio Costa is the Vice-president of the Scientific Counsil of of Instituto Superior de Engenharia do Porto, Portugal, and a Professor at the Informatics Engineering Department. He was the former Program Director of LEI-ISEP and the responsible for the current program structure of LEI-ISEP. He is the major disseminator of CDIO ideals at ISEP.

Corresponding author

Angelo Martins
Instituto Superior de Engenharia do Porto
Rua Dr. Antonio Bernardino de Almeida, 431
P-4200-072 Porto
+351 965 142 065
amm@isep.ipp.pt