STEPS FOR ITERATING DESIGN-IMPLEMENT EXPERIENCES INTO A CDIO COURSE

Sergi Bermejo, Miguel Ángel García-González, Ramon Bragós

Telecom BCN, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain

Núria Montesinos

Bellvitge University Hospital, L'Hospitalet de Llobregat (BCN), Spain

Montserrat Ballarín

Bellvitge Biomedical Research Institute (IDIBELL), L'Hospitalet del Llobregat (BCN), Spain

ABSTRACT

This paper describes how a particular form of iterative design-implement experiences were driven in the fourth-year Advanced Project in Science and Telecommunications Technology of the Degree in Sciences and Telecommunication Technologies taught at the Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona (Telecom BCN), in which students were faced with the incremental development of an innovative mirroring tool for facial paralysis rehabilitation. This work was done in the research framework established by Telecom BCN, Bellvitge University Hospital and Bellvitge Biomedical Research Institute (IDIBELL).

KEYWORDS

Design-Implement Project, Incremental Development, Telecommunication Engineering, Standards: 5, 11

INTRODUCTION

The CDIO approach meets the challenge of increasing the quality of engineering education by providing a learning environment in which students understand how to Conceive-Design-Implement-Operate (CDIO) complex technological products (Crawley et al., 2014). Accordingly, CDIO functions as a lifecycle model of these products where Design-Implement (D-I) remain as their key stages and usually involve an iterative loop to meet certain constraints and criteria in a process of refinement (Cross, 2008).

Iterative procedures wherein enhancements in some parts lead to modifications in others are actually inherent to the very nature of D-I. For building meaningful D-I experiences, implying practical hands-on activities that generate real-world verifiable results (Crawley et al., 2014), a kind of “Reflection-in-Action” (Schön, 1984) should be carry out, in which knowing and doing are inseparable and thus problem understanding and its solution have to be developed side-by-side. In a search for systematization different approaches to iterative, evolutionary and incremental development has been conceptualized and proposed, particularly in fields like software engineering (Larman, 2003).

This work shows how a particular realization of such iterative D-I experiences was introduced in the fourth-year Advanced Project in Science and Telecommunications Technology (12 ECTS credits) of the Degree in Sciences and Telecommunication Technologies taught at Telecom BCN. As the basis for engineering-based active and experiential learning in the context of a design-implement project, students were faced with a process of refining the design and coding a novel and pioneering software tool based on computer vision and machine learning (Prince, 2012) for aiding the neuromuscular re-education of hospital patients with facial paralysis (VanSwearingen, 2008) using a mirror therapy. This work was done in the cooperative research framework recently formed by Telecom BCN and IDIBELL (Domingo, et al., 2015) and looks upon some previously neglected issues such as the product usability tested in the intended target user.

The paper is organized as follows. In the next section, some key concepts on iterative D-I experiences are introduced and several proposals for their practical realization in the context of the Design-Build courses path at Telecom BCN are presented. The course Advanced Project in Science and Telecommunications Technology taught at Telecom BCN in collaboration with IDIBELL is then reviewed. Finally, conclusions are summarized in the light of the previous sections.

ITERATIVE DESIGN-IMPLEMENT EXPERIENCES: CONCEPTS AND PROPOSALS

Design-Implement experiences as central to Telecom-BCN curricula

The term Design-Implement experience is used to denote a whole range of engineering activities central to the process of developing novel products; while *design-* emphasizes plan definition, block diagrams and algorithms that describe the product, *-implement* refers to hardware building, software coding, testing, validation and any other stage involved to obtain an operating product from a design. Accordingly, D-I experiences necessarily include practical hands-on activities for students to design, build, generate, test, and operate a real product as a counterpart to theory for supporting, enhancing and deepening active and experiential learning in which students mimic professional engineering practice (Crawley et al., 2014). In this way, they remain crucial to project-based courses and to programs in general.

The need of iterative and incremental approaches to Design-Implement

Reflection-in-action in engineering design

Engineering design is similar to problem-solving since it involves a solution proposal in the form of an artifact (or a product) based on a conceptualization and understanding of a problem to be faced. As a consequence of this fact, problem and solution have to be evolved side-by-side (Cross, 2008; Koh et al., 2015) and engineering design can be then viewed as a "reflective conversation with the materials of a situation", wherein the designer frames and re-frames the problem yielding new discoveries which call for new reflection-in-action, forming an indissoluble knowing and doing; thus, "the process spirals through stages of appreciation, action, and reappreciation" (Schön, 1984). This reflection-in-action, which implies that knowing and doing are inseparable, involves a reflective practicum in academy, i.e. a learning-by-doing within a reciprocal dialogue between students and teachers (Binder et al., 2011) through practical hands-on activities that produce real-world demonstrable results (Crawley et al., 2014).

Iterative and incremental approaches to D-I

The spiral process of reflection-in-action is, in fact, that of Design-Implement experiences. Iterative procedures, in which enhancements in some parts lead to modifications in others, are very common in the stages of D-I, usually involving an iterative loop to meet certain constraints and criteria in a refinement process (Cross, 2008). Although a solution could be obtained using a waterfall model, i.e. through the execution of a convenient number of sequential stages, as soon as any of its details are not clearly defined, some form of iterative project management life cycle (PMLC) model should be employed instead (Wysocki, 2013).

The basic idea behind iterative PMLC (Figure 1) is to obtain a product through repeated (i.e. iterative) cycles and in smaller portions (i.e. incremental) at a time, allowing engineers to exploit what was learned during earlier stages of the cycle and, more specifically, through previous versions of the product. Enhancement and refinement comes then from both product development and use in a way that changes and new functional abilities are considered at each step. For (software) development projects, the most popular iterative PMLC models are, among others, Evolutionary Development (or, in short, Modified) Waterfall, Scrum and Rational Unified Process (RUP) (Larman, 2003; Wysocki, 2013):

- The modified waterfall model was proposed in the seventies and became a highly influential refinement of the simple stage-wise waterfall model, providing recognition of the feedback loops between stages, and a guideline to confine the feedback loops to successive stages to minimize feedback rework (Boehm, 1988).
- Scrum was originally formulated as a holistic approach with six features that join together in jigsaw puzzle-like fashion: built-in instability, self-organizing project teams, overlapping development phases, "multilearning," subtle control, and organizational transfer of learning (Takeuchi & Nonaka, 1986).
- The RUP (Kroll & Kruchten, 2003) has four project life-cycle phases –inception, elaboration, construction and transition– that are similar in presentation to a 'waterfall'-styled project and interact all of them. RUP uses three building blocks for describing what is to be produced (work products), the skills required (roles) and how development goals are to be achieved (tasks). Within each iteration, the tasks are categorized into six engineering disciplines –business modelling, requirements, analysis and design, implementation, test, Deployment– and three supporting disciplines –Configuration and change management, Project management and Environment–.

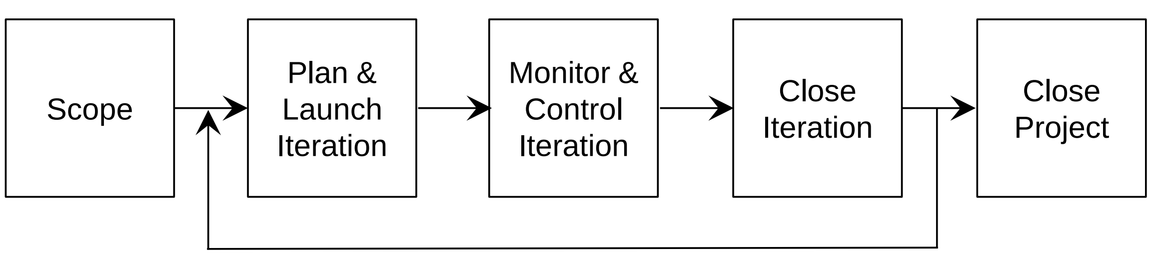


Figure 1. Iterative PMLC cycle (adapted from Wysocki, 2013)

Steps for an iterative proposal to Design-Implement in CDIO courses

A simple iterative model of D-I (Figure 2) is introduced in this section. It is not intended to provide a complete framework for managing highly complex projects but only to suggest tentatively a very preliminary model, loosely constructed upon the PMLC models reviewed above and suitable to be used in Design-Build courses. The basic features of the proposed model can be briefly summarized as follows:

- It is a spiral-based iterative model (Boehm, 1988) divided in three phases or cycles – inception, elaboration and construction–. Each cycle of the spiral starts with planning and requirements definition/modification followed by design and implement –divided in hardware work/software coding, testing and evaluation– and ends with a tollgate, measured objectively through a set of deliverables (see Table 2). The final prototype is deployed at the end of the third cycle.
- It is user/client-driven (Gould & Lewis, 1985) since meetings with clients and potential users are conducted to (re-)define product requirements and test and evaluate early and mid-term prototypes, thus allowing their refinement and enhancement along the process.

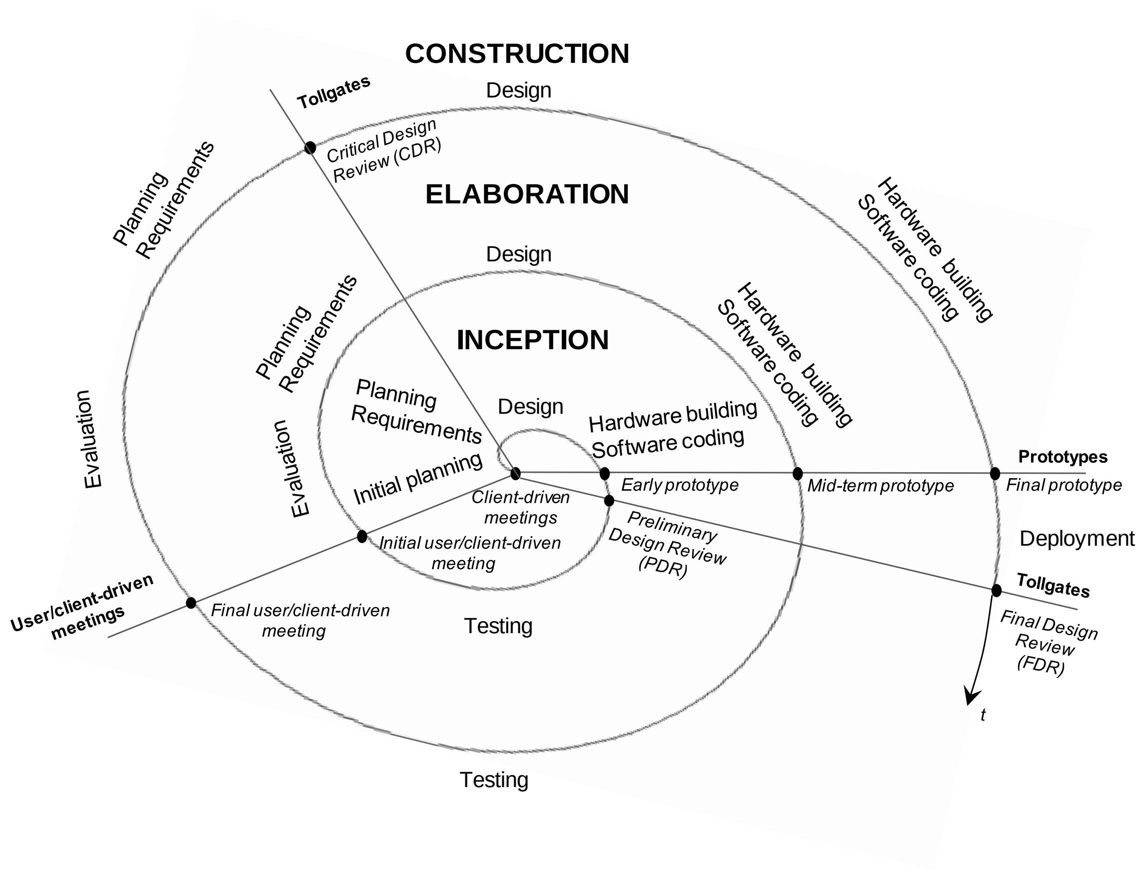


Figure 2. The suggested spiral model for iterative design-implement

Table 1. Tollgates and deliverables

# week	Tollgate	Deliverables
3	Preliminary Design Review (PDR)	- Project Charter (PMBOK Guide; Project Management Institute, 2013) - Project Management Plan
9	Critical Design Review (CDR)	- Reviewed Project - Management Plan
12	Preliminary Final Design Review (FDR)	
13	FDR	- Reviewed Project Plan - Project Report
13+	Project presentation + Demo	- Poster - Final Report - Business Plan - Presentation
weekly	Meeting minutes Lab logbook	

DEVELOPING A MIRRORING TOOL FOR FACIAL PARALYSIS REHABILITATION: A CASE STUDY IN ITERATING DESIGN-IMPLEMENT EXPERIENCES

Course Outline and Aims for 2015 Edition

The course in the academic plan

The fourth-year Advanced Project (AP) in Science and Telecommunications Technology (12 ECTS credits) is intended as a capstone course for students of the Degree in Sciences and Telecommunication Technologies taught at Telecom-BCN (UPC). It aims to provide students with a significant design experience and integration of knowledge from several courses for culminating conception, design and implementation of a complex system, and also a means to practice system thinking, project management, technical writing, and technical presentation skills. AP students were previously enrolled in two other Design-Build courses.

Course aims for the 2015 Edition

AP students were challenged with the design and building of a new and inventive software tool that relies on computer vision and machine learning (Prince, 2012) for aiding, through a mirror therapy, the neuromuscular re-education of hospital patients with facial paralysis (VanSwearingen, 2008). The software shows in real time to the patient a video where the affected half of the face is substituted by a mirroring of the healthy side. Moreover, the software takes measures of characteristic points over the face in order to track the progress of the patient while trying to do some facial movements such as smiling, mimicking a kiss, etc. The course was conducted in cooperation with the Rehabilitation Unit of the Bellvitge University Hospital and IDIBELL that worked as the client in the D-I model. This is one of the nine projects offered this semester.

Results and Discussion

Course preliminaries

Students were first divided in two groups (of 8-9 members each) with a leader and assigned to the same project to be developed in laboratory sessions for a total of 90 hours. Teachers acted essentially as technical consultants in these sessions and also monitored their group dynamics attending to weekly group meetings. Students were provided with a very rough version of the intended software written in C++ that make use of several machine learning and computer vision toolboxes (e.g. OpenCV) and a well-known proprietary integrated development environment.

The key role of user/client-driven meetings

At the beginning of the *inception phase*, two meetings between clients, students and teachers were done in order to define the basic requirements and the general functionality of the application; the first meeting was face to face and the second one by e-mail through a questionnaire. Later, two face meetings of four hours each were conducted in the rehabilitation unit of the Bellvitge Hospital between group leaders, a teacher, two physiotherapists (clients) and several voluntary patients affected by facial paralysis (users) at the end of *inception and elaboration phases*. In these meetings, software prototypes were extensively tested by users giving their detailed impressions about them. Also, clients gave important insights on the clinical framework in which these aiding tools will be incorporated.

Early, middle-term and final prototypes

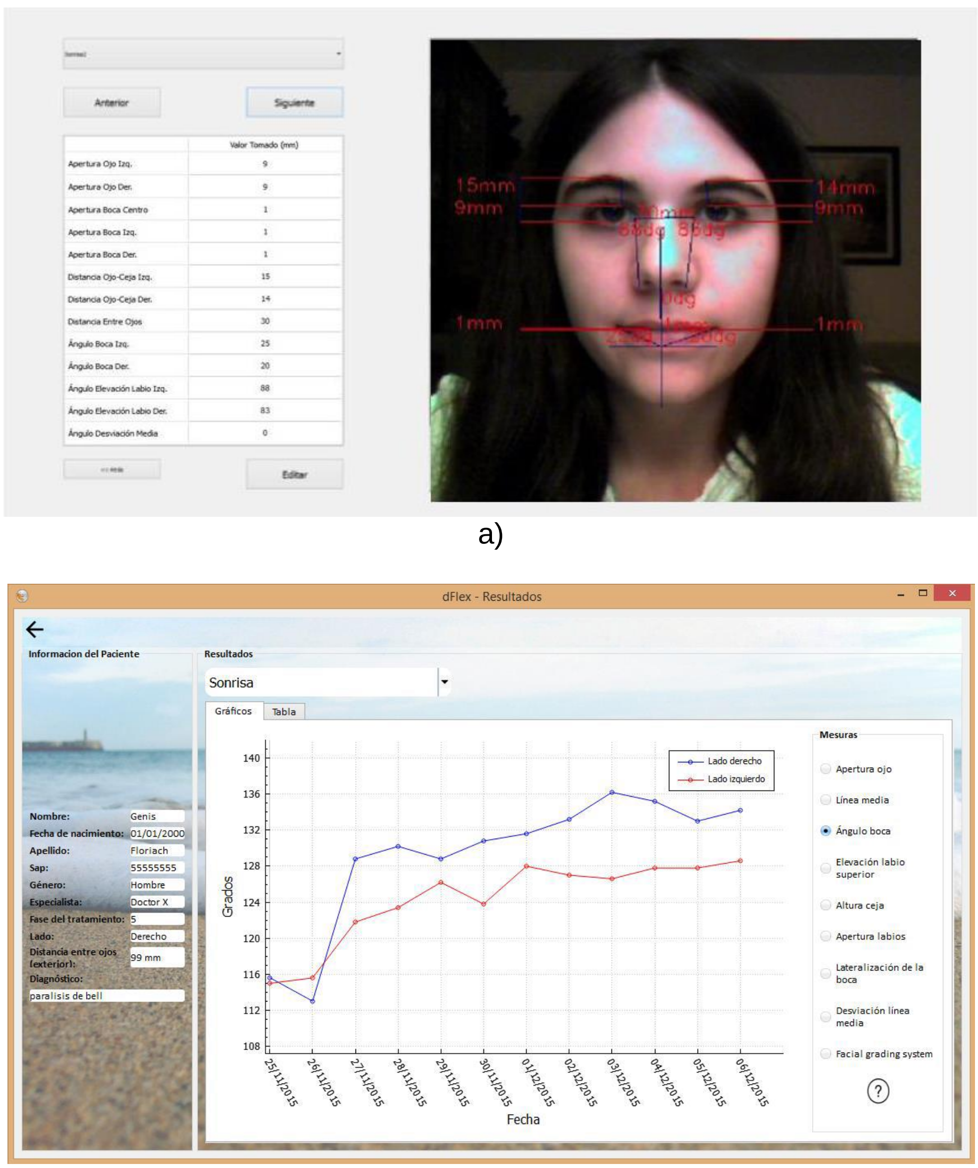
A clear differentiation in depth and sophistication between prototypes was present. Early software prototypes included only a basic version of the real-time mirroring facility that made possible a very fruitful user/client-driven meeting since their test and evaluation reveal relevant bugs to be solved and reveal new requirements. On the other hand, mid-term prototypes included important improvements of the basic mirroring algorithm in terms of speed and effectiveness and also incorporated new functionalities like a real-time automated measurement tool for assessing the user improvement among sessions and several managing tools to maintain a user database. Lastly, final prototypes also included advanced mirroring algorithms for enhancing the user experience and completed the managing and measurement tools for obtaining and maintaining a detailed set of user statistics.

Discussion

The use of an iterative D-I model allowed obtaining the gradual refinement and enhancement of a system that was elaborated through constraints that were introduced incrementally since some of them were only possible to be discovered after testing and evaluating a functional prototype with the help of real users and clients. In this sense, user/client-driven meetings remained vital for early development of operating prototypes and for providing important feedback and insight for further substantial refinement and enhancement. Students obtained not only vital feedback from direct users and clients (i.e. patients and physiotherapists) but a clear picture on the real conditions in which the software will be run that allow refining much better the most relevant technical requirements. For all these reasons, the final prototype (Figure 3) was much more complex and robust than those of prior editions of the AP course, in which basically a waterfall model was employed.

CONCLUSIONS

A preliminary iterative model for design-implement experiences has been proposed and tested in the fourth-year Advanced Project taught at Telecom BCN for students to incrementally develop a mirroring software tool for facial paralysis rehabilitation. The spiral-based model has three cycles –inception, elaboration and construction– for refining and enhancing incrementally previous design-implement efforts and relies on user/client-driven meetings and tollgates with deliverables for subsequent improvement of prototypes until their deployment. The application of such model allowed students to obtain a final prototype technically richer and better suited to the needs of clients and users than those based on an almost linear PMLC model employed in previous editions.



a)

b)

Figure 3. The software aiding tool for facial paralysis rehabilitation: a) the automated measurement window and b) the statistics windows

REFERENCES

Binder, T., De Michelis, G., Ehn, P., Jacucci, G., Linde, P., & Wagner, I. (2011). *Design Things*. Cambridge: The MIT Press.

Boehm, B. W. (1988). A Spiral Model of Software Development and Enhancement. *IEEE Computer*, 21 (5), 61–72. doi: 10.1109/2.59.

Crawley, E. F., Malmqvist, J., Östlund, S., Brodeur, D. R., & Edström, K. (2014). *Rethinking Engineering Education: The CDIO approach, 2nd Edition*. New York: Springer:

Cross, N. (2008). *Engineering Design Methods: Strategies for Product Design, 4th Edition*. New York: Wiley.

Domingo, T., Ballarin, M., Rojano, J. M., Oliveras, A., García-Hernández, M., & Bragós, R. (2015). Cooperation Framework with a Hospital as a Design-Build Projects Source. *Proceedings of the 11th International CDIO Conference*, Chengdu University of Information Technology, Chengdu, Sichuan, P.R. China, June 8-11.

Gould, J.D., & Lewis, C. (1985). Designing for Usability: Key Principles and What Designers Think, *Communications of the ACM*, 28(3), 300-311. doi: 10.1145/3166.3170.

Kroll, P., & Kruchten, P. (2003). *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Boston: Addison-Wesley.

Koh, J.H.L., Chai, C. S. , Wong, B., & Hong, H.-Y. (2015). *Design Thinking for Education: Conceptions and Applications in Teaching and Learning*. Singapore: Springer.

Larman, C. (2003). Iterative and Incremental Development: A Brief History. *IEEE Computer*, 36 (6), 47–56. doi:10.1109/MC.2003.1204375.

Prince, S. (2012). *Computer Vision: Models, Learning, and Inference*. Cambridge: Cambridge University Press.

Project Management Institute. (2013). *A Guide to the Project Management Body of Knowledge (PMBOK Guide), Fifth Edition*. Newtown Square, Pennsylvania: Project Management Institute.

Schön, D. A. (1984). *The Reflective Practitioner: How Professionals Think in Action*. New York: Basic Books.

Takeuchi, H., & Nonaka, I. (1986). The New New Product Development Game. *Harvard Business Review*, 64 (1), p. 137-146.

VanSwearingen, J. (2008). Facial rehabilitation: a neuromuscular reeducation, patient-centered approach. *Facial Plastic Surgery*, 24(2), 250-9. doi: 10.1055/s-2008-1075841.

Wysocki, R. K. (2013). *Effective Project Management: Traditional, Agile, Extreme, Seventh Edition*. New York: Wiley.

BIOGRAPHICAL INFORMATION

Sergi Bermejo is an Associate Professor in the Department of Electronics Engineering (DEE) of the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, and teaches at the School of Telecommunications Engineering of Barcelona (Telecom BCN). His research interests are machine learning, with a special focus on large-margin classification, unsupervised learning, and their application to statistical signal processing, smart sensors, and autonomous robotics.

Miguel Ángel García-González is an Associate Professor at the DEE of UPC. He lectures at Telecom BCN. He does research on instrumentation methods and biomedical signal measurement and he is interested on time series signal processing by time-domain, frequency-domain, time-frequency spectra and non-linear dynamics techniques, and noninvasive measurement of physiological signals.

Ramon Bragós is an Associate Professor at the DEE of UPC. His research area is electrical impedance spectroscopy applied to biomedical systems. He lectures at Telecom BCN, where he is the Associate Dean of Academic Innovation.

Núria Montesinos is physiotherapist in the Rehabilitation Unit of the Bellvitge University Hospital, L'Hospitalet de Llobregat, Spain.

Montserrat Ballarín is the head of the Research Support & Technology Transfer Office (OTRI) of IDIBELL.

Corresponding author

Dr. Sergi Bermejo
Universitat Politècnica de Catalunya
Jordi Girona 1-3 (C4 building)
08034 Barcelona, SPAIN
sergio.bermejo@upc.edu