

Collaborative Design Tools: A Comparison between Free Software and PLM Solutions in Engineering Education

Nicolas Maranzana, Frédéric Segonds, Frédéric Lesage, Julien Nelson

► **To cite this version:**

Nicolas Maranzana, Frédéric Segonds, Frédéric Lesage, Julien Nelson. Collaborative Design Tools: A Comparison between Free Software and PLM Solutions in Engineering Education. Louis Rivest; Abdelaziz Bouras; Borhen Louhichi. 9th International Conference on Product Lifecycle Management (PLM), Jul 2012, Montreal, QC, Canada. Springer, IFIP Advances in Information and Communication Technology, AICT-388, pp.547-558, 2012, Product Lifecycle Management. Towards Knowledge-Rich Enterprises. <10.1007/978-3-642-35758-9_49>. <hal-01526112>

HAL Id: hal-01526112

<https://hal.inria.fr/hal-01526112>

Submitted on 22 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Collaborative design tools: a comparison between free software and PLM solutions in engineering education

Nicolas Maranzana¹, Frédéric Segonds¹, Frédéric Lesage¹, Julien Nelson²

¹ Arts et Métiers ParisTech, LCPI, 151 boulevard de l'Hôpital, 75013 Paris, France
{nicolas.maranzana, frederic.segonds, frederic.lesage}@ensam.eu

² CNAM, Ergonomics Laboratory, 41 rue Gay-Lussac, 75005 Paris, France
julien.nelson@cnam.fr

Abstract. The shift from sequential to concurrent engineering has led to changes in the way design projects are managed. In order to assist designers, many effective tools have been developed to support collaborative engineering, whose implementation is perceived as complex. Nowadays, industrial scenarios encourage companies to adopt PLM solutions, even if, sometimes, they can't understand the benefits. On the other hand, many free solutions with comparable functionalities are developed, which have been increasingly successful.

In this article, we test different associations of software to make a comparison between free software and market solutions. In this experiment, 24 students in a Master's Degree course aimed to design mechanical products by using software to assist collaborative distributed design, using two different configurations. This experiment allowed us to compare design functionalities between free and commercial solutions, in order to determine ways to improve efficiency in a collaborative distributed design situation. Finally, the feedback generated in this experiment allowed us to adapt training practices in engineering education.

Keywords: Engineering education, PLM, communication, free software.

1 Introduction

With the globalization of design and the massive development of Business Process Outsourcing in various professions [1], one of the major stakes in design today is to allow efficient collaboration between project stakeholders regardless of their geographical location. The development of Information and Communication Technologies (ITCs) has made it possible to provide tools intended to facilitate distributed collaborative design [2]. Regarding the management of technical data, Product Data Management (PDM) tools have evolved to take into account this new situation. However, some of these tools, originating from the industrial domain, are cumbersome, which reduces their accessibility to users, who often require a long training period. Furthermore, current industrial situations force some businesses to adopt Product Lifecycle Management (PLM) solutions, even though these businesses are often unable to understand the possible benefits of such tools [3]. Additionally, several free

solutions have been developed with comparable functions, which have met increasing success.

In the matter of engineering education, training programs have evolved to take these changes into account. Design projects carried out in schools of engineering by Master's degree students simulate real-world situations of distributed collaborative design. The software tools used include Catia v5 for Computer-Assisted Design (CAD), and SmarTeam for PDM. However, users' difficulties in mastering the software may hinder the progression of these projects. Based on these points, the motivations for this study are as follows:

- All companies, and even students, may need to commit to mastering distributed design tools, especially in the context of globalization and Business Process Outsourcing [1].
- In the case of education for distributed mechanical engineering, the available free tools are easy to master, and are widely broadcast on multimedia platforms. This is not the case for commercially available solutions, although these have other merits. Our motivation for this study is to test the usability of these free tools for engineering education.

2 State of the art review

2.1 Evolutions in design methodology over the past 20 years

Starting in the late 1980s and continuing to the present day, methodologies for product design have evolved greatly. Towards the end of the 1980s and the early 1990s, two forms of design organization emerged as distinct alternatives: sequential design, which involves carrying out design tasks one after the other, and concurrent engineering, also known as integrated design [5-7]. Two aspects of concurrent engineering that distinguish it from conventional approaches for product development are *cross-functional integration* and *concurrency*. By carrying out all these tasks in a parallel fashion, it becomes possible to reduce the time and costs associated with design, but also to improve the quality of products. With the development of Information Technology (IT), concurrent engineering methods evolved gradually towards collaborative engineering, which emerged in the 1990s. As it is the case for concurrent engineering, overlapping tasks are still present in collaborative engineering, but project stakeholders are requested to work together and interact in order to reach an agreement and make shared decisions [8]. In the early 2000s, PLM emerged as a solution to better adapt industrial design to the demands of globalization. With the development of PDM, PLM and associated workflows, software firms proposed solutions to the everyday problems of engineering design departments (versioning of documents, naming conventions, etc.). PLM aims to cover all the stages of product development by integrating the processes and the people taking part in the project [9]. This concept is generally used for industrial products. The PLM approach can be viewed as a trend towards complete integration of all the software tools involved in design and operational activities during the product lifecycle [10, 11].

These evolutions in design methodology have all been made possible by the development of specific software tools. These were initially developed as a response to the needs of the industry. But in recent years, tools with similar functionalities have also been developed for the consumer market.

2.2 Existing solutions and related functionalities in the professional and the consumer markets

Many pieces of software offer functionalities to assist a PLM approach [12]. Editors of PLM solutions mainly originate from the CAD sector. For example, Dassault Systèmes includes MatrixOne within its Enovia v6 software. Solidworks offers the Enterprise-PDM, Workgroup-PDM and n!Fuze products. Other editors such as PTC offer Windchill and Siemens with TeamCenter. One should also point out the existence of editors exclusively geared towards PLM, such as Audros Technology and Lascom, as well as of other editors more closely related to frameworks of standards, such as ProStep and OpenPDM.

Current PLM tools offer functionalities that can be found in most of these software solutions [12]. These can be classified into three main categories: PDM, configuration management, and distributed design tools.

The main functionalities found in PDM tools are as follows:

- Access rights management: depending on the user's clearances, he or she is given access to information contained within the PLM system. Depending on this clearance, the actions available to users may be restricted (regarding reading, writing, and modification of documents). Concepts of *roles* and *groups* are often present in such systems. Roles refer to predefined access rights that administrators may ascribe to users. Groups are sets of users with similar rights.
- Vaults: datasets and related documents are stored onto a server called a vault, as opposed to being stored locally on the user's computer. Data are stored in an object or a relational database. Hence, information is structured according to the data model implemented within the database. Documents are stored on the server. When a document is opened, it is replicated onto the user's workstation, for a duration that depends on the software considered.
- Document visualization. Users are able to visualize quickly documents in various formats, without owning the application that corresponds to a particular file format.
- Checkout and check-in. This functionality allows users to check out a document in order to ensure that no other user working on the document at the same time may alter it. Once the document has been edited, the user checks the document back in to make it accessible to other users once again.
- Document versioning. Several versions of the same document may be archived. Two levels are used for versioning. The terms used are "version" (the higher level, generally indicated with a letter such as A, B, etc.) and "revision" (the lower level, usually indicated with a number, 1, 2, etc.). This system is used to distinguish major alterations from minor alterations.

- States. Various states are associated with each document. These help define their level of maturity: creation, validation, obsolescence, etc. Changes in these states may be decided based on the workflow, *e.g.* “awaiting validation”: project members will await the project manager’s authorization to carry out subsequent operations.
- Workflows. These systems make it possible to model processes and to automate actions. These systems are mostly used in validation processes for documents and technical data.
- Configuration management consists in controlling information related to product structure, especially breaking it down into elementary parts, and adding information related to their functional and physical characteristics [13]. The standard [14] includes recommendations for using configuration management in the industry. It provides the detailed process, organization and procedures for management. According to this standard, configuration management is an integral part of PLM; it provides a clear vision of the configuration state associated with a product or project, as well as their evolutions by guaranteeing total traceability [14].
- Distributed design tools [1]. These allow users to share a screen, to remotely gain control over another user’s workstation, and to exchange instant messages. They also allow the use of a webcam to visualize a colleague, or of VoIP in order to talk with him/her. Usually, these collaborative functions are taken on by other software programs, which may or may not belong to the consumer market, such as Skype or IBM Lotus Sametime.

PLM is currently evolving towards PLM 2.0, which takes advantage of the intelligence that is collectively generated by online communities. In this view, all users may imagine, share, and experiment with 3D products.

Current software editors follow a holistic approach when designing information systems in companies. This poses the question of adapting their software to the company’s organizational context, as well as the question of the compatibility of information systems within the company. Implementing an integrated information system – or more simply, a shared information system – should never hinder the development of a company [15]. One possible solution to avert this risk is to integrate software solutions from the consumer market, which allow users to access some of the functionalities associated with PLM applications.

Figure 1 presents the software tools used in our study.

In addition, new approaches have been developed to unify design tools and facilitate software interoperability [16]. A federative approach allows exchanges between the various product models generated by different business tools, in an independent and progressive fashion [17]. Several distinct product models are dynamically linked following one (or more) correspondence maps, based on several concepts which are related at the semantic level through relationships of similarity or equivalence.

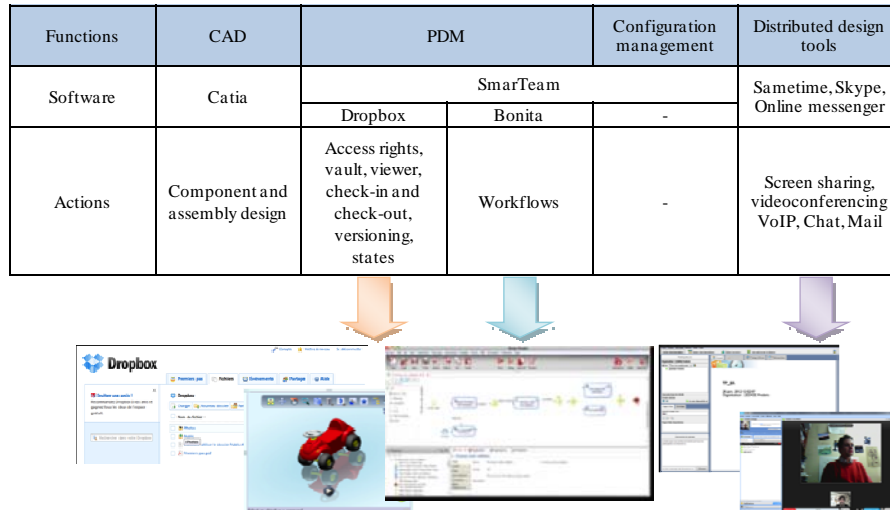


Fig. 1. Software tools and functionalities used in our study

2.3 A challenge for engineering education

In the field of mechanical engineering education, PLM is a means for students to structure their design methodology. From an educational point of view, a PLM method can be viewed as a sophisticated tool for analysis and visualization, enabling students to improve their problem-solving and design skills, but more importantly to improve their understanding of the behavior of engineering systems [1].

In our current, globalized world, products are typically designed and manufactured in several locations worldwide. It is therefore essential to train students to Computer Supported Collaborative Work (CSCW) [18]. In the field of engineering, companies and professional organizations expect students to have a basic understanding of engineering practices, and to be able to carry out tasks effectively, in a self-sufficient manner, as well as in a team environment [19]. Traditional design projects (*i.e.* involving co-located teams and synchronous work) were able to achieve these goals until a few decades ago, but they are no longer sufficient nowadays [8].

Furthermore, today's students have access to many tools for collaboration, which they often use outside of their studies. Tools such as Skype or Dropbox have become standards for remote collaboration. These tools are available on a wide range of interfaces (desktop computers, tablets, smart phones, laptops, etc. They are equipped with interesting functionalities and might, provided adequate support is available for education, be part of a program to train engineering students in the principles of distributed collaborative design. Over the past few years, our experience can be summarized in the following points:

- Little time is allocated to collaborative engineering design: in our case, 12 hours in a two-year syllabus;
- Over this time, students are unable to understand the operation of complex soft-

ware in any depth, because the typical user of such systems works on industrial tasks, involving millions of components to manage, several hundred users, etc. The student panel which we studied preferred free and easy-to-use software. For this reason, we propose in this paper a measurement of software usability [4] for the various software tools used, to compare their relevance in design work.

The experiment presented in the section below aims to compare the collaborative tools available in the Arts et Metiers ParisTech school of engineering with free solutions intended to carry out the same functions. This allows us to identify some pathways to educate our students in distributed collaborative design using the most appropriate tools, taking into account the scale of the design projects involved, and based on a concept of optimal support for user needs.

3 Experimentation

3.1 Educational approach and goals of the experiment

The work described here is an observational study based on scenarios [20, 21]. Our approach is to compare the use of commercial versus free software for the distributed design of mechanical products. In the design project we proposed to our participants, two distant teams collaborated and had to face problems which are partly related to some general aspects of distributed work, such as effective communication, building and sustaining a shared understanding, and conflict management.

3.2 Overview of the project

In this section, we first present the context of our study, followed by the products that were used as teaching materials in simulated design projects, and finally, the various team configurations.

3.2.1 Context and experimental protocol

Arts et Metiers ParisTech is a School of Engineering composed of eight centers located in France. The School has developed a collaborative engineering platform for the purposes of practical education in collaborative design, and to manage innovation design projects between its centers. Each center has computer workstations equipped with Catia v5 (CAD) and SmarTeam (PDM software).

The project studied here simulates the collaborative design of two simple mechanical components between two geographically separate design teams. The work session lasted four and a half hours. Twenty-four master's degree level students took part in the project. From an educational perspective, the goal was to allow students to apply principles of collaborative engineering. It was also to enable them to learn to store, structure, and secure technical data, to plot the evolutions of project documentation, and to use synchronous and asynchronous tools for collaborative work.

In the course of the project, the students were asked to fill in a survey. The survey relied on a series of questions focusing on the students knowledge of the various tools used, the usability of these tools, and their progress 1.5, 3, and 4.5 hours into the session. The terms related to usability and used in the questionnaire (“satisfaction”, “learnability”, “efficiency” and “errors”) were defined at the beginning of the experiment, to allow students to have a shared definition of these concepts. Five-point Likert type scales were used to assess the knowledge of participants and the usability of the chosen tools; for example, the questions asked included “How would you describe yourself in relation to <software x>, novice or expert?”, “How easy did you find it to learn how to use the software?”, “How many errors would you say you made during this session?”, etc.

3.2.2. The products to redesign

The brief given to students describes the 49 stages of a project aiming to redesign a mechanical product. It was published on a webpage accessible to students. The students were provided with two scenarios inspired from real-world situations of industrial collaboration.

Teams of students worked in one of two scenarios. The first scenario supporting our observational study aimed to simulate a redesign task for a set of gears (Fig. 2, left) following new specifications. The manager of the research department asks the project manager to redesign the piece, with help from a colleague.

The second scenario aimed to simulate the redesign of a tensioner (Fig. 2, right) to satisfy new conditions of use. Once again, the manager of the research department asks the project manager to redesign the element, with help from a colleague.

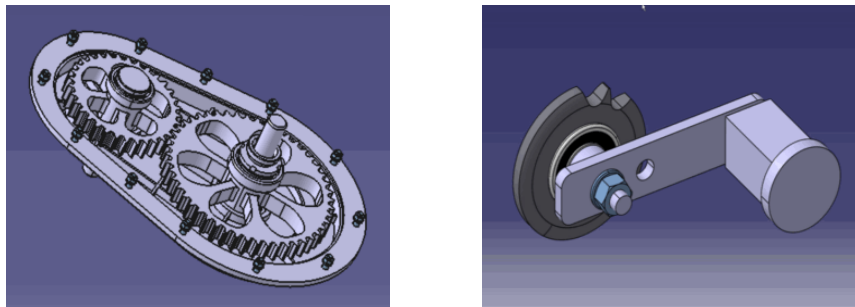


Fig. 2. The gears (left) and tensioner (right) in the redesign tasks

Team A took on the role of project manager for the tensioner and of designers for the gears. Conversely, team B, located on a remote site from team A, worked as project managers for the gears and as designers for the tensioner (Fig. 3).

	Team A	Team B
Project Manager	Gears	Tensioner
Engineer	Tensioner	Gears

Fig. 3. Summary of team roles in the design tasks

During a work session, the teams read the instructions to the 49 stages of the design task. These, in particular, provided the students with some guidance to create the project, a document structure, a new model for the set of gears or tensioner, as well as to organize reviews, to modify the product (e.g. changed bearings, altered measurements, etc.), to create the product's reference sheets, to draft technical drawings, to write a nomenclature, and to request validation of the parts produced in this way by the project management team.

3.2.3 Different software configurations

Our experiment aims to compare the effects of different material configurations to carry out the same tasks. Two software configurations, *i.e.* sets of software tools provided to the students, were proposed:

- The “blue” configuration includes the collaboration tools available within Arts et Metiers ParisTech, *i.e.* SmarTeam and IBM Lotus Sametime for chat, whiteboard, and screen-sharing capabilities;
- The “green” configuration includes free software programs that are thought to be equivalent: Dropbox, email client, Skype (for chat, audio/videocommunication and file sharing), IBM Lotus Sametime (used solely for its screen-sharing capabilities) and Bonita Studio (workflow management).

Both configurations also include Catia v5 software.

4 Results

In this part, we begin by describing the participants' profiles concerning their knowledge, prior to the project, of the proposed tools. We go on to describe the usability of these tools, using on Nielsen's four usual criteria for usability assessment [4], *i.e.* satisfaction, learnability, efficiency, and errors. We conclude at last by examining the progress made by participants in the design projects over time, in both the blue and green configurations.

4.1 Users' knowledge of the tools

Users ranked their knowledge of the various tools used between 1 (novice) and 5 (expert) at the beginning of the work session. Figure 4 shows the results obtained, depending on the material configuration.

Users' level of knowledge of the software tools in the blue and green configurations was broadly similar. This may be because the population of students was trained in the same school of engineering.

Catia v5, Dropbox and Skype were relatively well known, and students view themselves as relatively experienced in using them. The students had some basic notions in the use of SmarTeam. However, none of them had any experience of Sametime or Bonita Studio prior to the project. This suggests that the students have a fairly homogeneous level of expertise for the proposed tools.

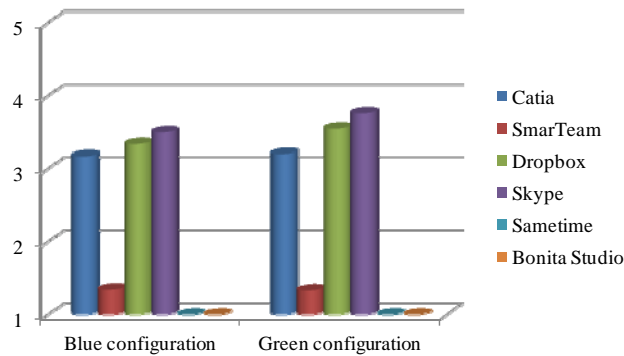


Fig.4. Users' knowledge of the tools (1=novice level, 5= expert level)

4.2 Usability of the software

To assess the usability of the various software programs deployed we applied Nielsen's approach [4]. The memorability criterion was not evaluated, since the project took place over a single, uninterrupted work session. Means for each of Nielsen's other criteria are presented on Figures 5 and 6, along with the corresponding standard errors. T tests were also performed for pairwise comparisons between the various software programs.

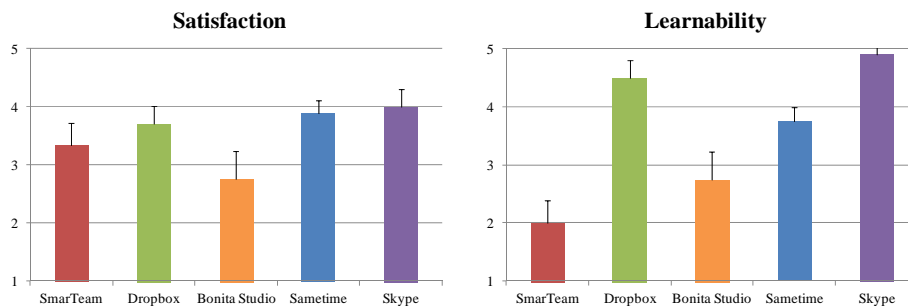


Fig. 5. User satisfaction (1=very low, 5=very high) and learnability (1=very difficult, 5=very easy) scores for each of the software programs used.

For the satisfaction criterion (Fig. 5, left), Bonita Studio was graded significantly lower than Sametime ($t(18)=2.1$, $p=0.05$) and Skype ($t(12)=2.23$, $p=0.045$). However, it should be noted that students had no prior knowledge of Bonita Studio prior to the study (see Fig. 4). Indeed, implementing a workflow is often considered a difficult task [22]. Other pairwise comparisons show no significant differences between means.

The learnability criterion (Fig. 5, right), however, clearly shows the students' difficulty in learning how to use SmarTeam, as opposed to Dropbox. Indeed, SmarTeam, was graded significantly lower than Dropbox, was ($t(14)=5.12$, $p<0.001$). Skype was viewed as very easy to learn, even considering the fact that all participants thought of themselves as expert users prior to the experiment. Sametime was viewed as fairly

easy to learn, considering none of the participants had used it before working on the project. However, it was graded significantly less than Skype in the learnability criterion ($t(24)=2.86$, $p=0.009$). In terms of learnability, the freeware programs used (Dropbox and Skype) clearly seem easier to learn than the tools proposed at the school, *i.e.* SmarTeam and Sametime. SmarTeam and Bonita Studio were viewed as complex products and probably require prior training.

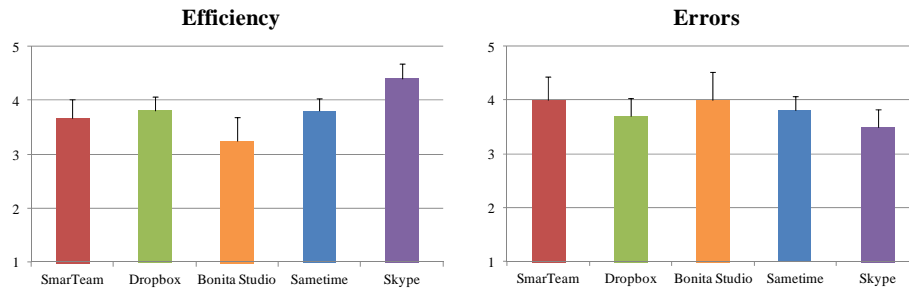


Fig. 6. Efficiency (1=very low, 5=very high) and error (1=many errors, 5=no errors) scores for each of the software programs used.

Efficiency grades (Fig. 6, left) were similar for all five programs, with a slight improvement when using Skype and a slight loss when using Bonita Studio: Bonita Studio was graded significantly lower than Skype for efficiency ($t(12)=2.97$, $p=0.012$). Other pairwise comparisons showed no further significant differences.

For the last criterion, *i.e.* error production (Fig. 6, right), pairwise comparisons revealed no significant differences between any of the software programs. Error rates were relatively low, probably because all the software solutions used were either commercial solutions or freely available solutions, but were never products in development.

4.3 Project completion over time

Let us now examine the final criterion addressed in our study, also measured in the survey: number of stages completed in the project over time depending on the configuration used (Fig. 7).

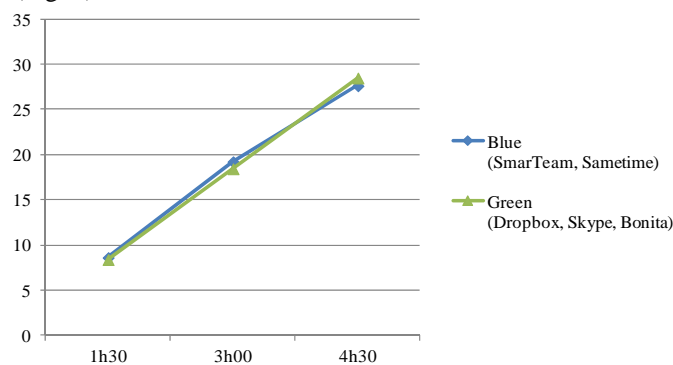


Fig. 7. Number of tasks completed over time

Examining the task completion rate 1.5 hours, 3 hours, and 4.5 hours into the work session, showed that the design teams' rate of progress was similar regardless of which toolset was used. In addition, none of the teams were able to complete all of the project's 49 subtasks. Therefore, in the context of two people located far from each other in a project to redesign mechanical components, the project completion rate will be similar regardless of which tools are used, for the configurations proposed in the paper.

5 Conclusions and research prospects

In this paper, we have tested the effects of different associations of design software, focusing on a comparison between freeware and commercial software solutions. An experiment was carried out as part of a short (4.5-hour) educational exercise, with pairs of students working in a synchronous manner.

Analyzing the results of the survey conducted during this experimentation allowed us to formulate two main contributions. First, the usability of the software used by the students was quantified following four prevalent criteria in the literature (user satisfaction, learnability, efficiency, and error production). Following this, we measured project task completion rate at three set times, and observed that the results seemed independent from the software configuration used.

These results must, however, be put into perspective. File access rights (*i.e.* the use of a vault with check-in/checkout functions), versioning, and file-naming conventions, all need to be taken into account when choosing a tool for work. Furthermore, data security must be taken into account, as this cannot be guaranteed today to companies who would consider using freeware solutions.

This study suggests some evolutions that could perhaps be implemented to freeware solutions, allowing users to access tools which complement (and compete with) existing market solutions. Following this approach, one might imagine lightweight tools for rapid implementation. This would allow designers to respond more efficiently to the requirements of short design projects and of companies based on small structures.

6 Acknowledgments

The authors would like to thank Stéphanie Buisine for her helpful suggestions.

7 References

1. Pezeshki, C., Frame, R. T., Humann, B., Preparing undergraduate mechanical engineering students for the global marketplace-new demands and requirements. in ASEE Annual Conference Proceedings. Salt Lake City, USA (2004)
2. Johansen, R., Groupware: Computer Support for Business Teams. Free Press, NY (1988)
3. Peruzzini, M., Mengoni, M., Germani, M., PLM benefits for networked SMEs. 8th International Conference on Product Lifecycle Management. Eindhoven, Netherlands (2011)

4. Nielsen, J., Usability Engineering. Morgan Kaufmann, San Francisco (1993)
5. Prasad, B., Concurrent engineering fundamentals: Integrated product and process organization (Vol. 1). Prentice-Hall, London (1996)
6. Sohlenius, G., Concurrent Engineering. Annals of CIRP, 1992. 41: p. 645-655.
7. Winner, R. I., Pennell, J. P., Bertrand, H. E., Slusarczuk, M. M., The Role of Concurrent Engineering in Weapons System Acquisition, I.R. R-338, Institute for Defense Analyses: Alexandria Va (1988)
8. Segonds, F., Maranzana, N., Veron, P., Aoussat, A., PLM and design education : a collaborative experiment on a mechanical device. 8th International Conference on Product Lifecycle Management. Eindhoven, Netherlands (2011)
9. Schuh, G., Rozenfeld, H., Assmus, D., Zancul, E., Process oriented framework to support PLM next term implementation. Computers in Industry, 59(2-3): p. 210-218 (2008)
10. Garetti, M., Terzi, S., Bertacci, N., Brianza, M., Organizational change and knowledge management in PLM implementation. International Journal of PLM, 1(1): p. 43 (2005)
11. Donati, T., Bricogne, M., Eynard, B., PLM platform: integrated support of the enterprise digital chain for Collaborative Product Development. 7th International Conference on Product Lifecycle Management. Bremen, Germany (2010)
12. Le Duigou, J., Bernard, A., Perry, N., Framework for PLM integration in SME networks, Computer-Aided design and Applications 8 (4), p 531-544 (2011)
13. Zina, S., Lombard, M., Lossent, L., Henriot, C., Generic Modeling and Configuration Management in Product Lifecycle Management, International Journal of Computers, Communications & Control, Vol. I, No. 4, pp. 126-138 (2006)
14. ISO 10007:2003. Quality management systems - Guidelines for configuration management (2003)
15. El Kadiri, S., Pernelle, P., Delattre, M., Bouras, A., Current situation of PLM systems in SME/SMI: Survey's results and analysis. 6th International Conference on Product Lifecycle Management, Bath, UK (2009)
16. Wegner, P., Interoperability. ACM Computing Surveys 28(1): 285-287 (1996)
17. Segonds, F., Iraqi-Houssaini, M., Roucoules, L., Véron, P., Aoussat, A., The use of early design tools in engineering processes: a comparative case studies. International Journal of Design and Innovation Research 5(3): 16 (2010)
18. Schmidt, K., Cooperative design : prospects for CSCW in design. Design Sciences and Technology, 6(2): p. 5-18 (1998)
19. Chen, Z., Siddique, Z., Web-based mechanical engineering design education environment simulating design firms. in Innovations in Engineering Education 2004: Mechanical Engineering Education, Mechanical Engineering Technology Department Heads. Anaheim, CA (2004)
20. Marin, P., Gerbaud, L., Mechekour, E., Masclat, C., An observational study of multi-disciplinary co-design – Application to an electromechanical device, Journal of Design Research, Vol. 6, N°3, pp. 311-332 (2007)
21. Martin, P., Veron, P., Training experience on Multi-site Collaborative Product Design', International CIRP Design Seminar, Grenoble, France (2003)
22. Bussler, C., Enterprise-wide workflow management, IEEE Concurrency, 7 (3), pp. 32-43 (1999)