Collaborative design tools in engineering education: Insight to choose the appropriate PLM software

Nicolas Maranzana[1] (iD),
Frédéric Segonds[1] and
Stéphanie Buisine[1,2]

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. Nowadays, industrial scenarios encourage companies to adopt product lifecycle management solutions, even if they may not be able to understand their benefits. Indeed, product lifecycle management roadmap is quite difficult to implement and return on invest can take time. Moreover, many free solutions with comparable functionalities are developed, which have been increasingly successful. In this article, we test different configurations of software to make a comparison between free software and market solutions. In this experiment, 72 students in a Master's degree course aimed to design mechanical products by using dedicated software to assist collaborative distributed design, using two different configurations: free and commercial solutions. The research question to be solved is: as engineering educators, what is the most efficient way to train our students to collaborative distributed design? This experiment allowed us to compare design functionalities between the two configurations, 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.

[1]Arts et Métiers ParisTech, LCPI, Paris, France
[2]Ei.Cesi, Irise, Nanterre, France
Corresponding author:
Nicolas Maranzana, Arts et Métiers ParisTech, LCPI, 151 Boulevard de l'Hôpital, 75013 Paris, France.
Email: nicolas.maranzana@ensam.eu

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 facilitate efficient collaboration between project stakeholders regardless of their geographical location. The development of Information and Communication Technologies 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. These evolutions have created the need for new skilled professionals, and universities should adapt their curricula in response.[3] There is an increased need for academia to work with industry and, therefore, an increased need to teach our students how to design a product in a collaborative way. However, some of these tools, originating from the industrial domain, are difficult to implement and to handle: this 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.[4] 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.[5–9] 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-Aided Design and SmarTeam™ for PDM. However, users' difficulties in mastering the software may hinder the progression of these projects. PLM is a key factor of success for industrial companies: it is therefore crucial for us to train our students in PDM/PLM manipulation and to make them understand the working principles of these systems. It brings to them skills in collaborative design in a globalized world. Thus, Gandhi[10] notices that PLM in education is the key to innovation and success in organizations in the engineering and technology sector. 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] Based on these points, the research question of this paper is: As engineering educators, what is the most efficient way to train our students to collaborative distributed design?. In the case of education for distributed mechanical engineering, the available free tools are most of the time easy to master and are widely broadcast on multimedia platforms. This is not the case for commercially available

solutions, although these have other merits. Thus, our contribution in this study is to test the usability[11] of these free tools for engineering education and to compare it to commercial solutions. To do so, Nielsen define four indicators for the usability: user satisfaction (i.e. the system should be pleasant to use, so that users are subjectively satisfied when using it; they like it), software learnability (i.e. the system should be easy to learn so that the user can rapidly start getting some work done with the system), efficiency (i.e. the system should be efficient to use, so that once the user has learned the system, a high level of productivity is possible), and lack of errors (i.e. the system should have a low error rate, so that users make few errors during the use of the system, and so that if they do make errors they can easily recover from them).

Literature review

Evolutions in design methodology over the past 25 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,[12–14] also known as integrated design.[15] Two aspects of concurrent engineering that distinguish it from conventional approaches for product development are *cross-functional integration* (degree of overlapping of the tasks) and *concurrency* (multiple stakeholders integration during the design process). 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, 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.[16] In the early 2000s, PLM emerged as a solution to better adapt industrial design to the demands of globalization (current period). 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.[17] 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.[18,19]

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. However, in recent years, tools with similar functionalities have also been developed for the general public.

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

Editors of PLM solutions mainly originate from the CAD sector. For example, Dassault Systems 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.[20] These can be classified into three main categories: PDM, configuration management, and distributed design tools.

The main functionalities found in PLM tools are as follows:

- PDM-related functionalities
- *Access rights management*: Depending on the user's clearance level, 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. It 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.[21] The standard (ISO 10007:2003)[22] 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.

- *Distributed design tools*: 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.[23] 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.[24] A federative approach allows exchanges between the various product models generated by different business tools, in an independent and progressive fashion.[25] 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.

Functions	CAD	PDM		Configuration management	Distributed design tools
Software	Catia	SmarTeam			Sametime, Skype, Online messenger
		Dropbox	Bonita	-	
Actions	Component and assembly design	Access rights, vault, viewer, check-in and check-out, versioning, states	Workflows	-	Screen sharing, videoconferencing VoIP, Chat, Mail

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

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.

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.[26] 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.[27] 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.[16,28]

Furthermore, today's students have access to many tools for collaboration, which they often use outside of their studies. Tools such as Skype™, Dropbox™, Trello™ or Slack™ have become standards for remote collaboration. These tools are available on a wide range of interfaces (desktop computers, tablets, smartphones, laptops, etc.) and data are increasingly stored in the cloud for an access from anywhere. 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.[29,30]

Over the past few years, our teaching experience can be summarized in the following points:

- Little time is allocated to collaborative engineering design: in our case, 12 h in a two-year syllabus;
- Over this time, students are unable to understand the operation of complex software 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[11] 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 Métiers 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.

Method

The work described here is an experimental study based on scenarios.[31,32] The experiment was carried out as part of a training program in collaborative engineering for Master's degree students at Arts et Métiers ParisTech. Our goal was to assess the usability of two different kinds of software solutions: (a) solutions that were readily available at Arts et Métiers ParisTech and (b) free software programs that are thought to be functionally equivalent.

Participants

Seventy-two students (22–24 years, average: 22.75 years, 20 women, 52 men) took part in this project. Because the use of CE tools involves collaboration between groups of people, participants were randomly distributed into 22 teams of two to four members. Each team had to collaborate remotely from two fictitious sites, named A and B (for e.g. Amsterdam and Basel). Any Participant belonged to one team only and one site only.

Materials

Teams of students used the available workstations. Each team was assigned one of the following software configurations:

- The *blue* configuration included the collaboration tools available within Arts et Métiers ParisTech, i.e. SmarTeam™ for PDM and IBM® Lotus Sametime™ for chat, whiteboard, and screen-sharing capabilities;
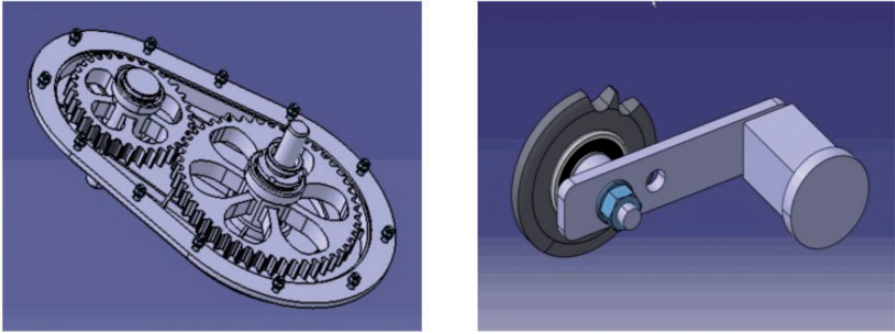
Figure 2. The reduction gear (left) and tensioner (right) in the redesign tasks.

- The *green* configuration included free software programs that are thought to be equivalent: Dropbox™, Skype™ (for chat, A/V communication, file sharing and screen-sharing capabilities), and Bonita Studio (workflow management).

Both configurations also included Catia™ v5 software.

Each team had to achieve successively two scenarios aiming to simulate product redesign projects through collaborative engineering: redesigning a reduction gear following new specifications (Figure 2, left) and redesigning a tensioner to satisfy new conditions of use (Figure 2, right).

Procedure

Upon arrival, each team was given a text describing a linear sequence of 49 tasks to be carried out within the duration of the work session (4.5 h). The overall task sequence is described in Figure 3. In order to stimulate collaboration, Project management responsibilities and Design responsibilities were always distributed between A and B sites. All in all, *project*, *team role*, and *software configuration* were fully counterbalanced: half of teams worked in blue configuration, and the other half in green configuration. The A site was responsible for project management in the reduction gear project and for design in the tensioner project, whereas the B site was responsible for project management for the tensioner and design for the reduction gear. Therefore, during the experiment, each participant worked on different products (reduction gear and tensioner) and took on different roles (project managers and designers).

Over the course of the experiment, participants were asked to fill in a Google® Form Survey meant to assess three aspects of their work: (a) their knowledge of the software tools used in the work session; (b) the usability of these tools; and (c) their progress—number of tasks completed—after 1.5, 3, and 4.5 h of work. Software usability was assessed based on four of the criteria proposed by Nielsen:[11] user satisfaction (i.e. the system should be pleasant to use, so that users are subjectively
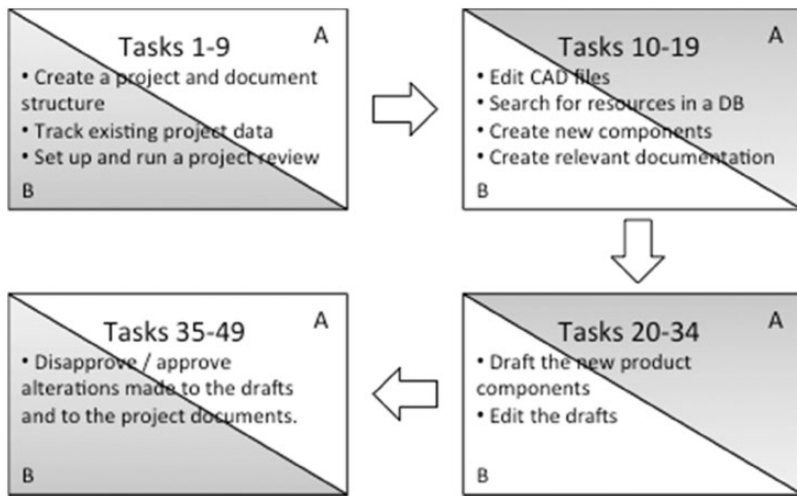
Figure 3. Overall structure of the work session. Gray: teams worked as project managers. White: teams worked as designers.

satisfied when using it; they like it), software learnability (i.e. the system should be easy to learn so that the user can rapidly start getting some work done with the system), efficiency (i.e. the system should be efficient to use, so that once the user has learned the system, a high level of productivity is possible), and lack of errors (i.e. the system should have a low error rate, so that users make few errors during the use of the system, and so that if they do make errors they can easily recover from them). Participants assessed each of these criteria, as well as their knowledge of the software programs used, using five-point Likert scales so that high scores always represent high usability (i.e. high satisfaction, high learnability, high efficiency, and low error rate).

Results

In this section, we present the participants' knowledge of the tools, we analyze their usability, and finally, we report on project completion over time.

Users' knowledge of the tools

Figure 4 shows the results obtained, depending on the software configuration. The student panel was 72 people: 36 in each configuration (blue and green).

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 the students viewed themselves as relatively experienced in using them. The students had
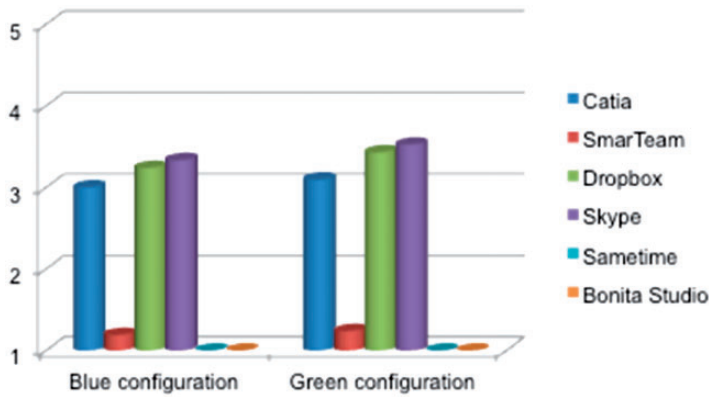
Figure 4. Participants' average level of knowledge of the software tools (1 = novice users; 5 = expert users).

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.
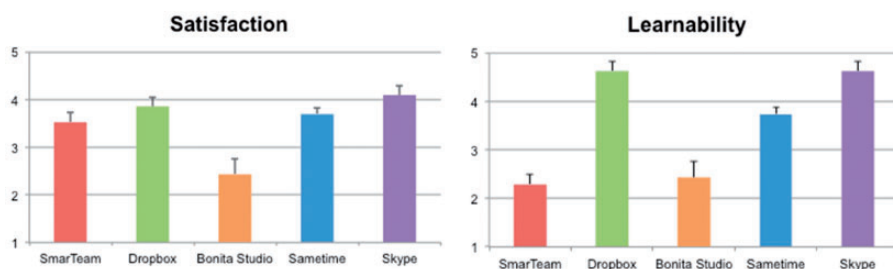
Usability of the software

In this section, we report on the analysis of usability of the five software tools (SmarTeam and Sametime for blue configuration, Dropbox, Skype and Bonita Studio for green configuration). The software tool was processed as a between-subject independent variable and each usability criteria (satisfaction, learnability, efficiency, errors) as dependent variables. Each team provided 2 ratings of the software used (1 rating by members in site A, and 1 by members in site B), which resulted in 22 ratings for each one of the five software tools (110 ratings in total). Univariate analysis of variance was performed on each usability criterion with H_0 stating the absence of difference between the tools (i.e. the five tools are similarly satisfying, learnable, efficient, and error-free) and H_1 hypothesizing that the level of usability (satisfaction, learnability, efficiency, and errors) depends on the tool. Previous experience with the tool was inserted in the analysis as a covariate. Finally, Fisher's LSD post-hoc test was used for pairwise comparisons between the five software programs. The analysis was run on SPSS PASW v21.

Table 1 displays all means and standard deviations for usability criteria.

Means for Nielsen's criteria of satisfaction and learnability are presented in Figure 5 along with the corresponding standard errors. Participants' level of satisfaction (Figure 5, left) differed significantly according to which tool was used ($F(4/104) = 6,43$; $p = 0.001$). Skype™ had the highest satisfaction score and was significantly better rated than Smarteam™ ($p = 0.039$). Bonita was rated as significantly less satisfying than all other tools ($p < 0.006$). Other pairwise comparisons showed no significant differences.

Table 1. Mean and standard deviation of each usability criterion for the five software tools analyzed, as rated by participants.

	Satisfaction		Learnability		Efficiency		Error-free	
	M	SD	M	SD	M	SD	M	SD
SmarTeam	3.62	0.81	2.67	1.07	3.81	0.93	3/86	0.86
Dropbox	3.86	0.83	4.64	0.73	3.5	0.86	3.45	1.22
Bonita Studio	2.11	1.27	2.11	1.17	2.33	1.41	3	1.66
Sametime	3.72	0.85	3.78	1.12	3.56	0.84	3.97	1.06
Skype	4.09	0.92	4.64	0.58	4.23	0.69	3.68	1.04

Figure 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.

Learnability ratings were also significantly influenced by the tools used ($F(4/104) = 10,16$; $p < 0.001$). Smarteam[TM] and Bonita were rated as significantly less learnable than the three other tools ($p < 0.001$), while Skype[TM] and Dropbox[TM] were rated as significantly more learnable than the others ($p < 0.001$). Figure 5 (right) clearly shows the students' difficulty in learning how to use SmarTeam[TM], as opposed to Dropbox[TM]. Indeed, SmarTeam[TM] was graded significantly lower than Dropbox[TM]. Skype[TM] 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[TM] was viewed as fairly easy to learn, considering none of the participants had used it before working on the project. In terms of learnability, the freeware programs used (Dropbox[TM] and Skype[TM]) were clearly viewed as easier to learn than the tools proposed at the school, i.e. SmarTeam[TM] and Sametime[TM]. SmarTeam[TM] and Bonita Studio were viewed as complex products and probably require prior training.

Efficiency ratings (Figure 6, left) also produced significant differences ($F(4/104) = 6.24$; $p = 0.002$), with Bonita obtaining lower results than the four other tools ($p < 0.041$) and Dropbox[TM] lower results than Skype[TM] ($p = 0.011$). Other pairwise comparisons showed no further significant differences.
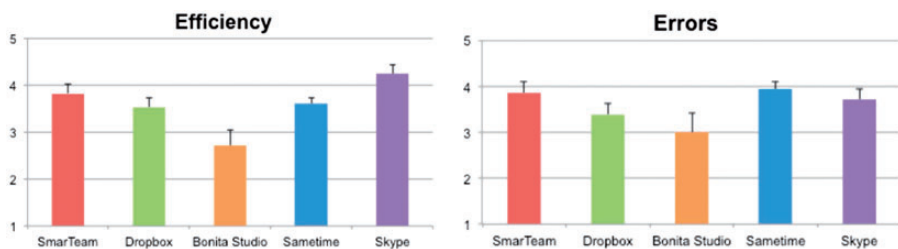
Figure 6. Efficiency (1 = very low, 5 = very high) and error (1 = many errors, 5 = no errors) scores for each of the software programs used.

For the last criterion, i.e. error prevention (Figure 6, right), pairwise comparisons revealed no significant differences between any of the software programs ($F(4/104) = 1.55$; NS). 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.

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 (Figure 7).

We performed a repeated measurement analysis of variance on the number of tasks achieved, with the three project Steps (1.5, 3, and 4.5 h into the work session) as a within-subject factor and the Configuration (blue, green) as well as the group (A, B) as a between-subject factors. Each team provided two answers for each of these steps (1 answer from members in site A, 1 from members in site B), resulting in 44 answers for each one of the three steps in project achievement. We tested two hypotheses: Regarding the effect of project Step, H_0 stated a stagnation of the number of tasks achieved through time, while H_1 hypothesized an increase of the number of tasks with time. We also tested the Step×Configuration interaction with H_0 stating the same progress over time in the two configurations and H_1 a differential progress as a function of the software configuration (green vs. blue).

We observed a main effect of the project Step ($F(2/80) = 255.5$; $p < 0.001$), which showed a significant progress of the students during time. Besides, a main effect of the configuration ($F(1/40) = 12.4$; $p = 0.001$) showed that students in the green configuration completed on average more tasks per step ($M = 19.5$; $SE = 0.85$) than students in the blue configuration ($M = 15.3$; $SE = 0.85$). Moreover, our results showed a Step×Configuration interaction ($F(2/80) = 7.67$; $p = 0.001$): students in the green configuration progressed faster than students in the blue configuration did (Figure 7). On average, these students carried out eight additional steps by the time they reached the 4.5-h mark, corresponding to a 25% increase in
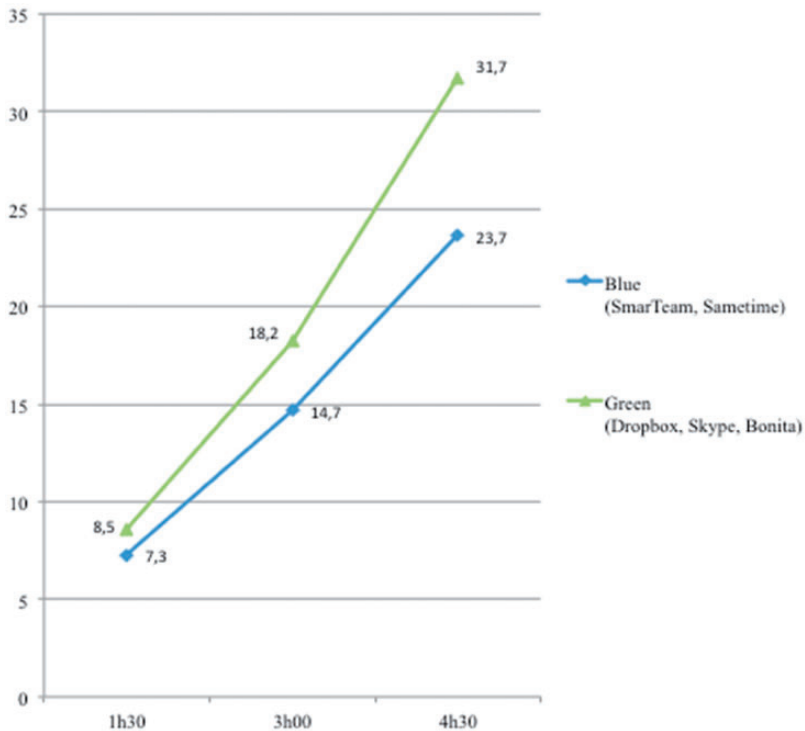
Figure 7. Number of tasks completed over time.

completion rate. The other results showed no significant main or interaction effects. In particular, the main effect of the Group was not significant ($F(1/40)$ $= 0.47$; NS), confirming that there was no bias introduced by the scenario or the artifact to redesign (reduction gear vs. tensioner) in terms of complexity.

Therefore, in the context of two remote sites in a project to redesign mechanical components, the project completion rate would be improved by using freeware programs. However, these conclusions apply to a simple redesign project, involving less than 20 components and less than 20 alterations. Further experiments would be necessary to know whether our results also apply to larger scale projects, involving more components and/or longer durations.

These results must also 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.

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 in a particular context. Indeed, the academic constraints (software solutions purchase depends on strategic choices at the university or school level; the choice is not necessarily possible at the teacher level), the educational context (with students and not under the industry pressure), and the short time allowed for the study (the time of the course) represent the limits of this study. The experiment was carried out as part of a 4.5-h educational exercise, with teams of students working in a synchronous manner.

Our research question was: as engineering educators, what is the most efficient way to train our students to collaborative distributed design? Analyzing the results of the survey conducted during this experimentation allowed us to study two main aspects. 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 prevention). Second, we measured project task completion rate at three set times. Finally, in this particular context, we observed that the results seemed to be improved by using a configuration with freeware programs.

The difference between usability levels of commercial and free tools may have been influenced by prior knowledge participants had of these software: for example, after only 4.5 h working with Bonita, they may have underestimated its efficiency and rated it accordingly low. For this reason, usability ratings, and particularly satisfaction and efficiency components, may evolve over time and further advantages of tools specific for mechanical engineering may arise with advanced use.

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.

As a short-term perspective for our engineering education program, we plan to modify the structure of the work session proposed in Figure 3, in order to allow our students to master workflow management skills (stages 35–49). Indeed, this notion seems very important in a distributed collaborative design environment.

Another perspective would be to carry out a longer project (few months) aimed at designing a new product in partnership with a company in a synchronous or asynchronous manner. Other software could also be tested as 3DExperience, Trello, Slack, etc. between students and engineers in order to get even closer to real design conditions.

Declaration of conflicting interests

ORCID iD

Nicolas Maranzana ⓘD http://orcid.org/0000-0002-1982-9041

References

1. Pezeshki C, Frame RT and Humann B. Preparing undergraduate mechanical engineering students for the global marketplace-new demands and requirements. In *ASEE Annual Conference Proceedings*, Salt Lake City, USA, June 2004.
2. Johansen R. *Groupware: computer support for business teams*. New York: The Free Press, 1988.
3. Fielding E, McCardle JR, Eynard B, et al. Product lifecycle management in design and engineering education: international perspectives. *Concurrent Eng* 2014; 22: 123–134.
4. Peruzzini M, Mengoni M and Germani M. PLM benefits for networked SMEs. In: *8th international conference on product lifecycle management*, Eindhoven, Netherlands, July 2011.
5. Mamo J, Farrugia P, Borg J, et al. Using engineering design tools in multidisciplinary distributed student teams. In: *17th International Conference on Engineering Design and Product Design Education (E&PDE'15)*, Loughborough, UK, September 2015.
6. Rosca D. Multidisciplinary and active/collaborative approaches in teaching requirements engineering. *Eur J Eng Educ* 2005; 30: 121–128.
7. Spinks N, Nicholas L, Silburn J, et al. Making it all work: the engineering graduate of the future, a UK perspective. *Eur J Eng Educ* 2007; 32: 325–335.
8. Rossouw A, Hacker M and De Vries MJ. Concepts and contexts in engineering and technology education: an international and interdisciplinary Delphi study. *Int J Technol Design Educ* 2011; 21: 409–424.
9. Tsai JP, Lee RS and Wang YZ. University/College Cooperation in course development: synchronous collaborative teaching/learning in advanced engineering in Taiwan. *Int J Mech Eng Educ* 2006; 32: 273–290.
10. Gandhi P. Product lifecycle management in education: key to innovation in engineering and technology. In: Fukuda S, Bernard A, Gurumoorthy B, et al. (eds) *Product lifecycle management for a global market*. Berlin Heidelberg: Springer, 2014, pp.121–128.
11. Nielsen J. *Usability engineering*. San Francisco: Morgan Kaufmann, 1993.
12. Prasad B. *Concurrent engineering fundamentals: integrated product and process organization* (Vol. 1). London: Prentice-Hall, 1996.
13. Sohlenius G. Concurrent engineering. *Ann CIRP* 1992; 41: 645–655.
14. Winner RI, Pennell JP, Bertrand HE, et al. *The role of concurrent engineering in weapons system acquisition, I.R. R-338*. Alexandria, VA: Institute for Defense Analyses, 1988.
15. Tichkiewitch S and Tollenaere M. *Advances in integrated design and manufacturing in mechanical engineering II*, Springer, 2007.
16. Segonds F, Maranzana N, Véron P, et al. Collaborative reverse engineering design experiment using PLM solutions. *Int J Eng Educ* 2011; 27: 1037–1045.
17. Schuh G, Rozenfeld H, Assmus D, et al. Process oriented framework to support PLM next term implementation. *Comput Ind* 2008; 59: 210–218.

18. Garetti M, Terzi S, Bertacci N, et al. Organisational change and knowledge management in PLM implementation. *Int J Prod Lifecycle Manage* 2005; 1: 43–51.
19. Donati T, Bricogne M and Eynard B. PLM platform: integrated support of the enterprise digital chain for Collaborative Product Development. In: *7th international conference on product lifecycle management*, Bremen, Germany, July 2010.
20. Le Duigou J, Bernard A and Perry N. Framework for Product Lifecycle Management integration in small and medium enterprises networks. *Comput-Aided Des Appl* 2011; 8: 531–544.
21. Zina S, Lombard M, Lossent L, et al. Generic modeling and configuration management in product lifecycle management. *Int J Comput Commun* 2006; 1: 126–138.
22. ISO 10007:2003. Quality management systems – guidelines for configuration management, 2003.
23. El Kadiri S, Pernelle P, Delattre M, et al. Current situation of PLM systems in SME/SMI: Survey's results and analysis. In: *6th international conference on product lifecycle management*, Bath, UK, July 2009.
24. Wegner P. Interoperability. *ACM Comput Surv* 1996; 28: 285–287.
25. Segonds F, Iraqi-Houssaini M, Roucoules L, et al. The use of early design tools in engineering processes: a comparative case studies. *Int J Des Innov Res* 2010; 5: 61–76.
26. Schmidt K. Cooperative design: prospects for CSCW in design. *Des Sci Technol* 1998; 6: 5–18.
27. Chen Z and 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, https://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1653003
28. Pavlova M. Teaching and learning for sustainable development: ESD research in technology education. *Int J Technol Des Educ* 2012; 23: 733–748.
29. Feldhusen J, Löwer M, Brezing A, et al. PLM-enhanced engineering education. In: *International conference on engineering and product design education (E&PDE'10)*, Trondheim, Norway, September 2010.
30. Sauza Bedolla J, Ricci F, Martinez G, J, et al. A tool to support PLM teaching in Universities. In: *10th international conference on product lifecycle management*. Nantes, France, July 2013.
31. Marin P, Gerbaud L, Mechekour E, et al. An observational study of multi-disciplinary co-design – application to an electromechanical device. *J Des Res* 2007; 6: 311–332.
32. Martin P and Veron P. Training experience on multi-site collaborative product design'. In: *International CIRP design seminar*, Grenoble, France, May 2003.