When users manage the cooperation between innovative firms : the case of the software industry.

I. Introduction :

Our research concerns the customer/supplier relationship during the development of radical software innovation. Taking the customers' needs into account has been identified as a key factor in the innovation process since the work carried out by Freeman (1982). Von Hippel and Katz (2002) even consider that the customers play a decisive role in the emergence and direction that the innovation takes. More specifically, in software innovation , it is taking the customers' needs into consideration that seems critical (Cusumano 2004). Fichman and Kemerer (1997) suggest involving the users in the phases prior to the conception and development of the innovations so as to take their needs into consideration. However, the first customers are likely to direct the innovation process towards their own needs which can be very specific. To what extent should the innovation designer meet these needs given that he is looking to develop a product that could be of interest to a large number of customers ? This is the question that our research covers. In order to offer some answers to the question we decided to study the case of a company in the business software field. We identify the methods behind the integration in the innovation process of the first customers. We show the implications of this integration on the organisation of the innovation project and on the architecture of the innovation itself.

II. Review of the literature and choice of the theoretical framework :

We'll refer on the one hand to the work of Von Hippel who, having emphasised the users' role in the innovation process, studied different configurations making it possible to distinguish them more systematically. We'll then present work on the modularity that underlines how this architecture contributes to encouraging innovation and meeting a variety of needs while at the same time benefiting from economies of scale. These two common aspects of work are not unique to the software field. Finally we'll position the analyses of the software development processes that

demand flexibility that is essential given the difficulty of explaining and anticipating customers' needs.

II-1. **The notions of *lead user, toolkit* and *user community* developed by von Hippel:**

Von Hippel (1986) suggests that firms should generate innovative concepts in partnership with *lead users*. The notion of *lead user* is defined as follows : it is about users (individuals or organisations) who are aware of needs which will subsequently become those of a large number of users, and who hope that by satisfying these needs they will make significant profits. As it is generally difficult for a user to define needs in relation to new products, a possible approach consists in selecting *lead user* customers and then getting them to test the prototypes in order to acquire information which can subsequently be exploited But, faced with the acceleration of the pace that innovations appear in each industry, it can be beneficial to leave certain users to develop their innovation themselves, as is the case for 80% of products in the scientific instrumentation sector (von Hippel, 1994). Hence, some firms go even further by choosing to equip their customers with tools which enable them to design and develop themselves the products that suit them . Von Hippel (2001) suggests an approach whereby the user receives the tools ("a user toolkit" or user toolbox) which allows the user to completely take in hand the design of the product, any possible industrialisation remaining the producers responsibility. The objective then is to incorporate certain solutions developed by these lead users in standard products which will subsequently interest a maximum number of users (Thomke and von Hippel, 2002). *Toolkits* also allow customers to develop a solution exactly adapted to their needs if they so wish. The *toolkit* approach applied in creating a new, innovative product involves the overall innovation process being broken down into sub-tasks assigned to either the user or the producer (von Hippel and Katz, 2002). Such task distribution can entail radical changes to the architecture of a product, and generally leads to the development of a modular architecture, as we can observe in free software (von Hippel and von Krogh, 2003).

For von Hippel (2001), the existence of a *user community* for innovation is fundamental; this community shares freely the knowledge acquired relating to the innovation as well as the improvements it contributes to it (example of free software or *open source*). A *user community* can only function under certain conditions. The first is that there are a significant number of sufficiently competent users who are motivated by the innovation. The second condition is that the multiple sources of innovation can be activated, so that each member of the community can benefit (if not, each user would have to entirely develop or fine tune the solution himself). Sharing the innovation results from incentives such as the growth of the reputation of the innovator and the creation of obligations towards the innovator. In addition, knowledge sharing leading to the growth of the diffusion of the solution can push manufacturers to incorporate it in the future concept of the product (Harhoff *et* al., 2000). In the case of innovations which should be incorporated in physical products (which have to be manufactured and distributed physically) the lead users can ensure significant development work and prototype tests, but production and distribution continue to be ensured by the manufacturer. On the other hand, in the case of intangible products, the community can, according to von Hippel (2001), ensure the complete range of functions of innovation processes and user innovation should be in competition with the innovation of commercial firms.

The question of the skills needed to take part in creating an innovation and therefore to be a lead user in a total sense is not covered very much in this work. Is it not determined by the role that the users can play in various technological contexts and in radical technological innovation ?

II-2. *Modularity* at the heart of literature about software architecture:

Modularity is at the centre of research on creating innovation in the computer field. “Different companies can independently design and produce components, such as disk drives or operating software, and those modules will fit together into a complex and smoothly functioning product because the module makers obey a given set of design rules” (Baldwin & Clark, 1997). *Modularisation* is the breakdown of a complex system into near-autonomous sub-systems which can be conceived independently (Baldwin and Clark, 1997 ; Aoki, 2002). The problem is then to

elaborate a complex system integrating these sub-systems. *Modularisation* can therefore be seen as a strategy of specialisation and division of tasks enabling the complexity to be mastered. For Aoki, the interest for this strategy is reinforced because (Aoki, 2002) the systems have become so complex that the *modularisation* has spread to the modules themselves. *Modularisation* can also be considered as a way of managing innovation because each person in charge of a specific module is in a position to innovate while respecting the rules laid down by the architect. However, the growing trend towards the *modularisation* of technologies and the disintegration of systems implies the production of new types of knowledge (Steinmueller, 2002), such as norms, standards and integration awareness (Shapiro and Varian, 1999). These new types of knowledge are necessary in the coordination, i.e. the integration of weakly linked groupings. The knowledge is then produced by two types of activities, research and coordination. The latter is in full expansion, determining the process of original innovation (Pavitt, 2002). How is this knowledge emerging ? Who is creating it ? How are the design rules being developed ? These questions remain open.

II-3 Flexibility of development processes :

Different studies have explored flexible models to develop a new product, characterised by the recovery of development stages (Krishnan et alii 1997). These models are based on a process which puts the accent on the capacity to be generated and to take account of new information as long as possible during the innovation development (McCormack, Verganti & Iansiti, 2001). Rather than a sequential process like “stage-gate”, the development becomes an “evolutionary” learning and adaptation process (Callon & Latour, 1991 ; Tushman & O’Reilly, 1997). The activities proceed in an iterative way with the return of information or “feedback” obtained during an experimental cycle used to direct the activities in the next cycle (Eisenhardt & Tabrizi 1995). It’s in the software field that the flexible development models have been the most widely recommended (eg: the “incremental” model, Wong 1984 ; the “spiral” model, Boehm 1988 ; or the “rapid prototyping” model, Connell & Shafer 1989). Models like these have been developed to solve the problems encountered with the traditional development model, “waterfall”. The “stage-gate” type model is the

result of efforts made by companies looking to control the management of large software development projects (Royce 1970). Flexible models for software development rely on an iterative process based on the construction of a series of prototypes that have to make it possible to obtain *feedback* on the balance between design and customer needs. Literature points out that a number of companies use this approach in different ways (Cusumano & Selby 1995, Iansiti & MacCormack 1997). McCormack, Verganti & Iansiti (2001) worked on the correlation between the flexibility of a software development model and the project performance level. In this way, by studying the development of Internet software, these authors show that the *flexibility level of the development process* is indeed correlated to its performance level. This flexible process is characterised by the capacity to generate and reply to new information over a longer proportion of the development cycle. Such a process is supported notably by larger investment in the software architecture design phase and earlier *feedback* from the market on the performance of the product. This type of conception process analysis has been highlighted in other sectors like the car industry or chemicals (Midler, 1993 ; Charue-Duboc & Midler, 2002).

Is this flexibility criteria also relevant for radical innovation ?

III Methodology:

With the view of bringing answers to these questions, we have chosen to study the case of an innovative company in the software field. We have concentrated on a series of projects aiming to use applications at different customers that draw on the same innovative software brick. We compare these projects' organisation and innovative offer. This comparative approach is aimed at identifying the learning and irreversibility effects created by the first innovative products designed for the first customers compared with products from the same family designed for the next customers.

III-1. Research setting:

SoftCo, where we carried out our study, produces innovative software packages: especially *text mining* software which allows the extraction, categorisation and cartography of information

contained in any body of text (newspaper articles, legal texts, patents…). SoftCo's *text mining* software is built on strong technological foundations : SoftCo, in addition to their own patents, draws on the patents developed at Xerox's European research centre for which it has acquired the licensable rights that represent ten man years of research. SoftCo has a software range which principally comprises ESoft (terminological extraction), KSoft (document categorisation), CSoft (*clustering* ,grouping of documents presenting similarities) and MServer (application able to join together the different SoftCo software programs). For each customer, a personalised development which completes ESoft has to be carried out in order to define exactly the terms and concepts that the customer wishes to extract: SoftCo calls this development a *skill cartridge*. Each sale assumes, in addition, personalised developments so as to integrate this "standard" software into the customer's systems which themselves are developing. This family of innovations can be considered as a radical innovation. On the one hand it is based on a combination of innovative algorithms of semantic analysis and statistical analysis. On the other hand it introduces a break in practices. In fact, the offer proposed by SoftCo to their customers modifies the working habits of the firms that adopt it, by enabling the automation of text analysis: automatic (or semi-automatic) indexation and categorisation, which up until now had been carried out manually. This subsequently allows the electronic storage of documents (enriched by metadata such as the date of publication), and of the knowledge they contain (such as the principal theme of an article).

III-2. Data Collection:

We have followed longitudinally and in real time two projects for two big, different companies (a newspaper group and a publishing house).The objective for each one was to develop a solution making it possible to design and manage a knowledge base (using the *text mining* technology). This solution is based on software which extracts, categorizes and archives the information. These different software solutions were mainly developed by SoftCo (that offers *text mining* software as such) and by another medium size French company, KnowCo (that develops a knowledge-based

management system). The aim of these two projects is to develop an application for this software for newspaper groups and publishing houses.

The first project concerns the creation of a knowledge base for the newspaper group PressPro. It was to facilitate the creation of theme based news files for journalists. It was carried out between 2003 and March 2006. The second project was based on the creation of a knowledge base for the publishing house EditPro. Its aim was to compile legal summaries sold to firms of corporate lawyers. It began in 2005 and should be completed with the presentation of the solution at the end of 2006. Throughout the projects we interviewed on a regular basis, on average once every two months, the people working for SoftCo in contact with the customer (project managers, technical managers, account manager) in the form of semi-directive interviews, as well as the project managers at PressPro and EditPro. We were also present at internal project meetings , and in all took part in thirty one meetings and interviews between June 2004 and June 2006:

- 15 interviews with the people involved at SoftCo in the PressPro project,
- 12 interviews with the people involved at SoftCo in the Edit Pro project,
- 2 in-house meetings at SoftCo, one on each project,
- 2 interviews with the customer's project managers.

The longitudinal monitoring aimed at limiting the bias linked to a posteriori rationalisation (Girin,1990 ; Yin, 1994 ; Dumez, 2004) and to set up material in companies where there are not many formal documents.

We were looking to collect three types of information on in-house project management at SoftCo , on the customer/supplier relationship and on the role of the customer in the innovation process. We asked questions on organisation choices, initial hypotheses, progress during the project, the customer's influence on developments ; the nature of the interactions set up between the customer and the software designer, and the difficulties encountered during the three phases of these projects : design, development and presentation of the software.

IV Case analysis:

We are now going to present the projects that we have been following and we'll then come back to the participation of the first customers in defining the offer and its design. After this we'll describe the flexibility of the development process and finally we'll deal with the personalisation of the product with the software module called "knowledge cartridge"

IV-1. Presentation of the projects:

The two projects studied involved different types of actors:

- Several software producers (including SoftCo and KnowCo in the two case studies),
- The customer,
- And in the EditCo project, a *service provider*, i.e. an IT service firm in charge of ensuring that the software solution is integrated in the customer's information system.

- **The first project, PressCo:**

Start / Finish	October 2003 / March 2006 (follow-up project of a "version 2" of the solution).
Integrator of the	None (role ensured by the software producers themselves)

software solution (service provider)			
Software project manager	XLMCo		
Software producers participating in the project	SoftCo	KnowCo (knowledge-based management system)	XLMCo (XML format document archiving system)
Standard software used	1. ESoft (terminological extraction), 2. KSoft (categorisation of documents)	ASL	XLMCo Server,
Specific software developed for the project	1. "people" *cartridge* developed specifically for this project, 2. bridge between KnowCo ESoft and ASL.	1. Application joining together the software programs of SoftCo, KnowCo and XLMCo 2. Thesaurus, and "biography" classification plan 3. coupling with SoftCo ESoft software.	1. filter enabling the recovery of PressCo archives, in the XLMCo server, 2. bridge between ESoft and XLMCo Server.

- **The second project, EditCo:**

Start / Finish	May 2005 / mid-2006
Integrator of the software solution	ConceptPro
Software project leader	SoftCo

Software producers participating in the project	SoftCo	KnowCo
Standard software used	ESoft, KSoft	ASL
Specific software developed for the project	1. "Legal" *cartridge* 2. bridge between KnowCo ASL and ESoft (adaptation of the "PressCo" bridge)	1. "Legal" thesaurus 2. coupling with SoftCo ESoft software (adaptation of "PressCo" coupling)

The longitudinal monitoring that we have carried out over two years, has enabled us to see the emergence of a unique innovative offer: the linking of an extraction tool (ESoft) and an information categorisation tool (KSoft), with a knowledge base management system (ASL), designed for firms supplying electronic-support contents (texts).

IV-2. The role of the first customers in defining the offer and its design :

In the first PressPro project there were three stages that could be identified when the customer played a constructive role in the innovation process.

PressPro's role was first to set up a partnership between software producers. In fact, at the origin of the PressCo project, we find the new head of the documentation department, who requests SoftCo and KnowCo to associate their terminology extraction software and knowledge-based management system. Then, it is yet again PressCo who suggests associating XLMCo, a specialist in XML format document archiving, at the beginning of the project. The customer was therefore to have a significant

contribution in defining the modular architecture of the solution. The solution developed for PressCo is broken down into two software bricks from SoftCo, one from KnowCo and one from XLMCo to which has to be added the *documentary application* which joins them together, and the *specific skill cartridge*: in all there are six distinct software modules plus the bridges between these modules.
Secondly, during the development phase of the solution itself, meetings between groups of users and software project managers made it possible to "transfer" to the software producers the information needed on the profession of a researcher and the context in which the future solution was to be used . The meetings held nearly every week and over a period of 12 months between the software project managers and the *user groups* show the effort made to provide information in order to get there.

Finally, during the "pre-production" phase , i.e. during the phase prior to the definitive deployment, the PressPro researchers put forward several explicit requests on operational aspects (relating to the constitution of files and information searches) that the three producers had not thought about.

The second project was launched once the partnership between SoftCo and KnowCo was up and running. The second customer contributed to the structure of the offer and to the completion of the project though in a more active way than the first customer. Here, as well, we'll concentrate on three stages : on the planning of the project, during the development phase and on the end of the project when it is unveiled. EditPro, a customer, suggested a small scale service provider, ConceptCo, a specialist in knowledge management who had already done some work for them. As a result, ConceptCo signed a partnership agreement with SoftCo (as a value added reseller). The presence of ConceptCo did not prevent SoftCo and KnowCo interacting strongly at the same time with EditPro on the project. The question of a service provider participation was also raised in the first project but the customer had not agreed to the software producers' request , the latter assuming the integration task themselves. EditPro also suggested a "supervisory" project integrator but the latter quickly disappeared from the project because his role turned out to be non-existant.

Then EditPro themselves developed a fictitious knowledge base to carry out software tests which led them to request an improvement in some of the system's performances such as the access time into the database. According to the SoftCo and KnowCo project managers this turned out to be very useful because it also created an improvement in the software interface. (These two developments were even able to be integrated into the PressPro project which had not yet finished).

Finally EditPro, the client, agreed and even requested SoftCo to train its programmers on its personalised *skill cartridge* so that in the longer term it could create and maintain its own solutions.

On the first project , the start of the process was the subject of a number of uncertainties.

First of all, the customer could not test the solution prototype. In fact , putting a prototype together quickly was not conceivable because the interface between the SoftCo, KnowCo and XMLCo software did not exist. In the absence of a prototype, defining the specifications for the solution was more difficult. The specifications remained vague in terms of the performance level expected (no quantification of the extraction quality) and the type of documents that the system should be able to handle. Drafting precise specifications for the terminology extraction was not easy because the customer didn't know exactly what he wanted at this point, just that it should be "the best possible solution". *In fine,* PressPro's implicit expectations proved to be too high when compared with the performances that the developed solution could achieve. In fact, the newspaper articles that PressPro wanted to analyse were written in a more literary way and were less pure fact than legal texts or patents, for example, and this makes terminology extraction more complex. The PressPro project ended in a mixed way since the solution that was set up did not give complete satisfaction.

However, without the work carried out on the PressPro project, the EditPro project could not have been opened out in the same way. The main changes in project management that were taken into account for the EditPro project were as follows :

- A quick development of a solution prototype which could be tested, enabling a better comprehension by the customer of the possibilities of the future software.
- Drawing up the requirements and specifications much more precisely which then enabled a better understanding and appropriation of the project by the customer, as well as better work guidelines for the SoftCo, KnowCo and EditPro project teams.

One can also observe major differences in the skills of the service provider and the customer when comparing the two projects which, according to the members of the SoftCo project team, contributed to the smooth running of the said project.

- ConceptCo, the service provider, knew how to develop the application used to join up the different software bricks by relying both on its skills in software integration and in knowledge-based management. This enabled SoftCo and KnowCo to concentrate on their software bricks, in interaction with the customer.
- The EditPro project team were not only skilled in terms of their profession but also had skills in *text mining* terminology extraction technology and in knowledge-based management systems. Thus, the EditPro project manager knew both *text mining* and knowledge-based management while two future users of the system, members of the EditPro project team, had language skills (useful when tackling terminology extraction) in addition to the skills of their profession. It was this, according to the SoftCo project team members, that enabled them to have a better understanding of what they could expect from the future system.

In summary, the work carried out in the first project enabled SoftCo and KnowCo to learn , mainly thanks to the customer PressPro, how to create an innovative software offer for the *Publishing* sector. The comparison of the two successive projects leads us to point out what was learned from the first project and then used in the second project :

- SoftCo and KnowCo defined a joint standard offer during the first project

bringing together a number of their software applications with software bridges needed for their interface,

- The SoftCo and KnowCo project managers defined a methodology and ways of Cooperating for their future joint projects,
- The difficulties encountered on the PressCo project resulted in SoftCo and KnowCo modifying their approach to the EditPro project so as to achieve a better understanding and assimilation of the project by the customer (understanding and assimilation which are, moreover, made easier by the nature and extent of the customer's skills).

IV-3. A complex but flexible process :

Additionally, we can observe a great flexibility in the organisation of the PressCo project, without leading to major deadline problems (*only* several months behind the initial schedule). This flexibility particularly affects the tasks to be carried out and the responsibilities attached to these tasks : thus the number of software producers involved in the software solution development changes at different times (one then two, then three software producers in contact with the customer) ; we can also observe the responsibility of the development of the *documentary application*, which has to link up the different software bricks, and which is transferred after a few months from PressCo to XLMCo, which moreover is not their job ; in the same way, recovering the press articles already stored by PressCo, which should have been done by XMLCo, is finally done by EditCo ; in addition the users groups end up refusing to take part in project meetings from the beginning of 2005, but this does not block the process which continues with just one group of users. Finally, we can add that the "theoretical" project manager of the global solution, who is also the project manager of the "XLMCo" part, only exerts very slight control which in turn contributes to the great adaptability of the whole project while generating a lack of clarity which irritates the SoftCo and KnowCo project teams.

We should also mention the flexibility shown by the software producers. This flexibility is due to their small size while their motivation is a decisive factor in bringing projects to a successful conclusion, which in turn can become benchmarks and therefore they feel that they constitute a unique means of learning and of completing their innovative offer. Thus, for example, when XLMCo are slow to develop the filter making it possible to retrieve archives stored in PressCo's former knowledge-base, it is SoftCo's developers who carry out this task so as to not slow down their part of the project, and this without any opposition from XLMCo. Similarly, when there is a lack of manpower, SoftCo does not hesitate redeploying developers responsible for SoftCo's standard software or *core products*.

IV.4 Skill cartridges and offer personalisation

To be able to exploit SoftCo' software, it is necessary to develop what SoftCo calls a *skill cartridge*, which must contain the specific terminology of the sector studied (for example economic intelligence in the petroleum sector, initial handling of CVs received by the HRD of a bank). Initially, SoftCo thought that each customer could develop their own *skill cartridge*, after having defined the concepts and the terms that they wanted to extract automatically. SoftCo placed a lot of hope on its *toolkit* called *STDK* , a development environment and collection of tools aimed at helping the programmers in their work and at facilitating the development of the personalised part of the solution to be carried out by the customer. This tool was therefore supplied to the clients.

However, in the PressCo project, SoftCo had to assume the development of the cartridges themselves. Rapidly, SoftCo had to conclude that their customers were not using this *toolkit* which was not particularly user-friendly. The SSII contacted did not seem interested in the development of the cartridges because in their view the *text mining* market was still in its infancy and they balked at the idea of investing in it. SoftCo made up their minds to continue with these developments in the short term and to only use the *toolkit* in house for the time being.

Nevertheless, the *STDK* was not abandoned and an engineer continues to work regularly on this project alongside his other projects: the strategy of SoftCo being that the var and service provider

partners develop a *skill cartridge* for their customer. For the final customer to be able to develop the skill cartridges themselves in the medium term, the man-machine interface and the navigation interface would need to be improved, documentation would need to exist and the SoftCo developers would need to take stock of its use. In the spirit of the people in charge at SoftCo, the *STDK* should be the last stage in the development of their offer, which should enable this producer to limit their service activities to concentrate on sales of *off the shelf* software which is conceived as being more profitable. Moreover, the personalised development being carried out by the service provider or the final customer, would resolve a critical problem for SoftCo which is the follow-up and maintenance of the personalised software that has been developed in this way.

V Discussion

V.1 Lead-user ?

We can observe therefore that the customers show *lead user* characteristics.

First of all, the customers were looking for solutions to their specific problems. By meeting their needs, innovation should bring them advantages : productivity gains and task enrichment for the PressPro project users, an improved offer for the EditPro project bringing added value to the company's customers. They went as far as defining the outlines of the offer that suited them and engaged several software producers requesting that they work together for the occasion. After the software producer selection period , a close relationship is formed between the customer, involving both his users and computer department, and the software producers. The latter put an organisation in place enabling them to work in an organised and consistent way with the groups of future users as well as with the management of the customer's information systems. For these reasons that we can consider that these clients have taken part in the design process for this innovative solution.

In addition, we should point out that the users needs are precise as well as being at the avant-garde of a market that can be predicted in terms of the automated analysis of free text. In fact, the increasing volume of information that individuals and organisations can have access to leads us to predict a

growing need for automated analysis of free text, the growth of which could reflect that of the *data mining* market which covers the analysis and treatment of formatted information stored in data bases.

However, we remark that the customer does not have the expertise of the technology brought by the innovation (statistical and linguistic analysis combined), nor the capacity to integrate a project including several software producers who should supply a common offer. *Text mining* is a new technology, and even a new concept. Therefore it is not well known by firms so it seems normal that the first customers cannot relate to its potential, its limits and how it's implemented. This naturally contributes to limiting their ability to assimilate. The direct interaction between innovative software producers and users has, moreover, the aim of "evangelising" the latter, i.e. of convincing them of the relevance of the concept, while simultaneously trying to understand how they perceive it. It is true that PressCo, the first customer, had the idea of associating the software to different producers but they could not really control the project nor perceive its limits. PressCo does not follow through the innovation development process to the end and is not able to construct a complete solution which satisfies their own needs, their action does however contribute to the elaboration of a finalised offer. However, EditPro participates actively in the project is being trained to be able to develop its own *skill cartridges* as quickly as possible.

User communities represent for von Hippel (2001), a unique means of sharing information and of creating an innovation. They are particularly highlighted in the software industry, both open source and ownership.

In the cases studied, we cannot talk about a *user community* because there are naturally few users, and because the latter do not yet possess the expertise needed for software adaptation. Nevertheless, as the number of users and the capacity to carry out their own developments are increasing, this situation could well evolve in the future. As we have said, *text mining* is a new concept and as a result is difficult for companies to grasp. Launching a *text mining* offer entails the diffusion of this new concept, bringing different skills together and creating a new market.

V.2 Modularity and creation of the offer:

Companies developing innovative software are often quite small and this goes hand in hand with a specialisation process (Horn, 1999). It leads to the offer being split up and renders the setting up of global solutions; that the customers are supposedly seeking, more complex. SoftCo, KnowCo and XLMCo know each other, but before the specific demand of the customer had not thought about working together to propose a joint offer. They even felt that to some extent they could do without the skills of the other two producers, judging that they were themselves sufficiently expert in their specialist fields. This project led to a formal association and led to SoftCo and KnowCo proposing a joint software offer for the creation and management of a knowledge-based system using *text mining* technologies.

This natural modularity allows users to test and to give feedback on each module: thus, in the case of PressCo, the researchers could test a solution containing only the KnowCo and XLMCo modules, before the SoftCo modules had been finalised.

The cases studied seem to show that the specialisation of innovative software producers leads to the direct intervention of several actors with customers at the same time. This complicates the task of the customer, swept up in a process of interaction with several actors in parallel and also complicates the task of the software producers who have to coordinate their work and their collaboration with their customer. At the beginning, in the PressCo project, neither the customer nor the software producers involved wanted to work in this way. It was impossible because KnowCo's skills in terms of terminology extraction were inadequate. The two producers were driven to working directly with their client and even asked the third, XMLCo, for help with an archiving solution for newspaper articles. The intervention of a service company, responsible solely for the integration of the solution as was the case in the second project did not get rid of these numerous interactions.

This multi-player modular structure of the offer brings about a multiplication of interactions between the customer and the software producers, and among the software producers themselves. These permanent interactions create a condition which favours the development of the software bricks and

their associations, so as to achieve the solution which is sought after. They also make it possible, for each software producer, to “sell” the new concept to users, after it has been tested and reformulated dialectically.

All in all, it seems that the natural *modularity* of the innovative offer favours its assimilation by the customer in that it permits multiple interactions between the customer and each software module producer and facilitates the tests and feedback on the sub-parts of the solution. On the other hand it complicates the global vision of the project and the integration of the solution.

This modularity enables work on different tasks to be done in parallel, even though some of them (definition of interfaces of course but not solely) require several producers to work together. This last point is one of the main factors which makes the innovation process a flexible one and therefore improves performances (McCormack, Verganti and Iansiti, 2001).

The naturally modular structure of an offer made up of software developed by a variety of independent companies enables it to evolve, without questioning its architecture or any of the modules it is comprised of (Baldwin and Clark, 1997).

V-3. Discussion : The *toolkits,* skills required for the customers and the obstacles

Thus, the *toolkit* concept which enables the customer to develop his own solution, though possibly relevant in the longer term, does not realistically seem operational during the launch phase of the offer. This may seem to contradict the results of some work (von Hippel, 2002). We suggest therefore that this point is studied in more detail in the follow-up to our work.

In addition, SoftCo, capitalising on previous projects, can offer generic *skill cartridges* in its range : for example, the development of a *skill cartridge* of economic intelligence in the oil industry for a particular customer has made it possible to add to SoftCo’s range of standard software, a generic *skill cartridge* of economic intelligence that can be adapted to each business sector. Thus we can see the

elaboration of an innovative software offer enabling knowledge bases for suppliers of texts with numeric content to be created.

VI Conclusion :

Our research brings to the fore a lead-user whose skill is primarily one of usage and whose contribution to the design of the innovative software is in the architecture of the complete offer. This result in the field of proprietary software is quite different from work on free software which underlines the lead-users' role in the writing of the code for innovation software. In our case, the customer chooses the software "bricks" from different companies to be brought together so as to meet his own needs and sets up contracts between the companies in order to achieve an integrated application. This leads each company to modify its "brick" and to create bridges enabling the whole application to function. The companies also write the code needed to integrate the new application in the customer's computer system. The architecture is designed by the first customer for his specific needs and it is then taken up by the software companies in their offer to the second customer. Thus we show how the software companies studied transfer the contribution from the first client to the software design. It is difficult in our case to show such a user community but the learning effect that Von Hippel emphasises can be found. It is in the sequencing of the projects and relies on the fact that the same software companies are encouraged to cooperate in a recurrent way for similar offers.

The second area that our work throws light on, relates to the elaboration of design rules that make possible a modularisation of systems and their design. For computers, these design rules have already been established by a dominant player in the sector, by Intel for example. Our research reveals another construction process for design rules. The modular structure results in the existence of a number of specialist software companies that have developed specific bricks. The design rules gradually emerge as these companies are encouraged to cooperate and construct a global offer meeting the client's needs and requiring the bricks to be linked.

However the policy in the computer industry has been towards vertical disintegration together with modularisation. We are analysing a move contrary to combining specialist offers in an integrated global offer. We are thus emphasising the work needed to make the modules interconnecting and software design that can only be achieved in a situation of cooperation.

We are demonstrating that the elaboration of design rules is split between two players : the lead users who request a number of companies to work together and these same companies who adapt their modules and design the necessary interfaces to make them interconnecting.

The third point that we'll come back to concerns the toolkits. The software companies designed their offer with customization "cartridges" enabling the software to be adapted to the customers' specific needs. These toolkits seemed of little use to involve customers at this advanced stage of the innovation where the priority is the construction of a first prototype. The involvement requested of the customer was both too demanding and irrelevant compared with his concern for the construction of a global offer to meet his documentary research needs. At this point design with a customized cartridge seemed quite interesting to facilitate the software company's inter-project learning effects and to limit the adaptation costs of each customer and not to involve the customer in the software design.

The advances noted between the first and second projects, encourage us to make the assumption of a gradual maturity in the offer by the enrichment of the different projects, and at the same time a maturity in the interaction between the innovating software producer and the first users. The first projects are going through a phase where the customers, motivated by the innovation but not in control of it, contribute to the offer emerging in cooperation with the software producers who have to build a modular offer and put in into context so that it can be operational. From this point, once the offer is stabilised, everything is in place for a *toolkit* that can be used by the customers or by the service providers. This assumption and its general nature in the event of radical innovation would be interesting to test in future research.

Bibliography :

- Aoki M. (2002), “Modularity : Its Relevance to Industrial Architecture”, Le Centre de recherche économique de Saint-Gobain.

- Baldwin, C. Y. and K. Clark (1997), “Managing in the Age of Modularity”, *Harvard Business Review*.

- Boehm, B. (1988), “A Spiral Model of Software Development and Enhancement”, *IEEE Comput.*, vol 21, pp 61-72.

- Callon M. et Latour B. (1991), « La Science telle qu’elle se fait », La Découverte Paris.

- Charue-Duboc F.et Midler C. (2002), "L'activité d'ingénierie et le modèle de projet concourant", Sociologie du travail, vol 44, pp 401-417

- Connell J.L. & Schafer L. (1989), *Structured Rapid Prototyping: An Evolutionary Approach to Software Development*, Yourdon Press, Englewood Cliffs, NJ.

- Cusumano, M. (2004), *The Business of Software*, Free Press, New York.

- Cusumano, M. & Selby, R.W. (1995), *Microsoft secrets*,- Free Press New York

- Dubois Anna & Lars-Erik Gadde (2001); “ The construction industry as a loosely coupled system – Implications for productivity and innovativity”, Paper for the 17th IMP Conference, 9-11 september 2001, Oslo, Norway.

- Dumez H. (2004), « Elaborer la théorie à partir des données », Revue Sciences de Gestion, N°44, pp. 139-155.

- Eisenhardt K. & Tabrizi B. N. (1995), “Accelerating Adaptive Processes: Product Innovation in the Global Computer Industry”, *Administrative Science Quarterly*, Vol. 40.

- Fichman, R.G. and Kemerer, C.F. (1997) “The Assimilation of Software Process Innovations: An Organizational Learning Perspective”, *Management Science*, vol. 43(1), pp. 1345-1363.

- Freeman, C.(1982), *The Economics of Industrial Innovation, Second Edition*, Frances Printer, London.

- Girin J. (1990). L'analyse empirique des situations de gestion : éléments de théorie et de méthode. In (s/d) Martinet, *Épistémologies et sciences de gestion*. Paris: Éditions Economica.

- Harhoff D., Henkel J. & von Hippel E. (2000), "Profiting from voluntary information spillovers: How users benefit by freely revealing their innovations", MIT Sloan Scholl of Management.

- Horn F.(1999), "Diversité des informations traitées par des moyens informatiques, standardisation optimale et acteurs du processus de standardisation", Communications et stratégies n°33, pp. 85-117.

- Iansiti, M. & MacCormack, A. (1997), « Developing Products on Internet Time », *Harvard Business Review,* vol. 75, pp 108-117.

- Krishnan V., Eppinger S. & Whitney D. (1997), " A Model-Based Framework to Overlap Product Development Activities" *Management Science,* vol. 43, N°4, pp. 437-451.

- MacCormack A., Verganti R. & Iansiti M. (2001), "Developing Products on "Internet Time" : The Anatomy of a Flexible Development Process", *Management Science,* vol. 47, N°1, pp. 133-150.

- Midler C., *Twingo, l'auto qui n'existait pas*, InterEdition, Paris, 1993.

- Pavitt, K. (2002) "System integrators as "post-industrial" firms", DRUID Conference, Copenhagen Business School.

- Raymond E. (2001), *The Cathedral and the Bazaar (revised edition)*, O'Reilly, 2001.

- Royce, W.W. (1970), "Managing the development of large software systems: Concepts an techniques", *Procedures* WESCON, Western Electric Show and Convention, Los Angeles. Repirnted 1989 in *Proc. 11th Int. Conf. of Software Engnrg.* Pittsburgh USA.

- Shapiro, C. & Varian, H. R. (1999) "The art of standards war", *California management review*, 41 (2): pp. 8-32.

- Steinmueller, W.E. (2002) “Collaborative innovation: rationale, indicators and significance.” In: J. de la Mothe and A.N. Link (eds), *Networks, Alliances and Partnerships in the Innovation Process*; Boston: Kluwer, pp. 29-43.

- Thomke S. & von Hippel E., (2002). « Customers as Innovators , a New Way to Create Value », *Harvard Business Review*, April, pp. 74-81.

- Tushman M., O'Reilly C.A. (1997), *Winning through innovation*, Harvard Business Press.

- von Hippel, E. (1986). “Lead Users: A Source of Novel Product Concepts”, *Management Science*, vol. 32, no. 7: pp. 791–805.

- von Hippel, E. (1994) “Sticky Information and the Locus of Problem Solving : Implications for Innovation”, *Management Science*, vol. 40,4 : pp. 429-439.

- von Hippel, E. (2001). “Perspective: User Toolkits for Innovation”, *Journal of Product Innovation Management*, vol.18, pp. 247–257.

- von Hippel, E., & R. Katz. (2002). “Shifting Innovation to Users via Toolkits.” *Management Science*, vol. 48, no. 7: pp. 821–833.

- von Krogh, G. & von Hippel, E. (2003). « Special Issue on Open Source Software development», Editorial in Research Policy 32/7 : pp. 1146 - 1157.

- Yin, R. (1994). *Case study research: Design and methods* (2nd ed.). Beverly Hills, CA: Sage Publishing.