INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

PILOT PROJECTS: THE ROAD FROM SHOULD-BE TO TO-BE

Marco Bertoni, Marco Ugolotti
Politecnico di Milano, Italy

ABSTRACT

This paper presents a methodology to guide the definition of Pilot specifications when dealing with the design and implementation of a Knowledge Management System (KMS) aiming to introduce a collaborative working paradigm in the Extended Enterprise. Main scope of this work is to support the KMS development team in configuring a Pilot testing environment able to provide valuable feedbacks for final system tuning, reducing meanwhile time and cost needed for its design and implementation. On one side, the methodology aims to help process owners in better expliciting and formalizing their expectations about the new system and to enhance their level of involvement in Pilot design. An integrated use of Mock-up representations and Enterprise Modelling techniques, such as IDEF and UML, is, therefore, proposed for the Pilot requirements definition stage, in order to enhance interoperability among all process stakeholders.
On the other side, the methodology supports consultants and process owners to speed up the decisional process related to the selection of the specific simulation context in which the Pilot solution will be tested. In order to do so, main implementation parameters (in form of testing environment, people involved, time needed for the trial, etc.) have been analyzed and grouped to derive a set of standard Pilot configurations which can be applied in a semi-automatic way to test the new system depending on the purpose of the trials. One of the main advantages of this approach is the possibility to derive the Pilot implementation context automatically from the metrics chosen for its evaluation. The use of predetermined standard Pilot configurations, moreover, can be helpful in order to better interpret the reliability and the quality of data obtained by the simulation.

Key words: Knowledge Management, Process Modelling; Pilot Implementation; Should-Be process; Mock-Up

1. INTRODUCTION

The necessity to succeed or, at least, to survive, in the global market forces companies to improve the quality, capability, and timeliness of their products, at reduced costs. In order to reach these ambitious objectives, they are asked to reduce the duration and to improve the robustness of the product development process, which means doing things better and taking better decisions.
In such a context, industries try to satisfy their increasing need for knowledge and expertise by establishing highly complex and dynamic relationships between a vast number of specialist companies, which are asked to communicate and collaborate in spite of different skills, organization and ways of work. The introduction of such a collaborative working paradigm can result in enormous reductions in costs and cycle time and it can also strongly improve the quality of the design process. However, many factors can slow down or impede initial objectives' achievements. Very often the design and implementation of a new Knowledge Management Solution (KMS) in the Extended Enterprise have shown, in fact, a limited success [1]-[3].
Since the implementation of a Knowledge Management system deeply impacts on the design team's way of work and, in general, on product development process performances, it can be useful to verify "a priori" the viability and effectiveness of the proposed approach [4]. Before embarking on a full-scale implementation, it could be helpful to know if the KMS would really provide, once implemented, those functionalities required by the users [5].

A physical prototype can so be built to make sure the KMS will meet both business and technical needs and to anticipate future benefits and drawbacks [6]. These trials take the concepts out of the realm of theory and provide empirical knowledge of what can reasonably be expected by the new technology/ methodology. The way in which this *Pilot* is conceived and configured is critical for the KMS success, since empirical data obtained by the trials are at the basis of the final system tuning. Taking the wrong direction at this step, can cause resistance to the introduction of the new working paradigm and, consequently, big losses in terms of time and money.
Main aim of this paper is to propose a methodology (Figure 1) able to support the entire Pilot design and implementation process, starting from user requirements identification to the definition of the specific implementation context for the trial.

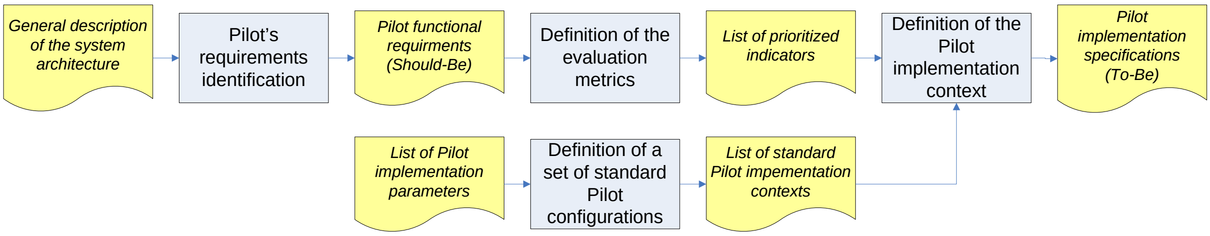

Figure 1: Methodology for Pilot implementation specifications definition

The methodology here presented tries to address some of the problems emerged during the development and implementation of a Knowledge Enabled Engineering (KEE) system in the frame of the European project VIVACE (Value Improvement through a Virtual Aeronautical Collaborative Enterprise) [7] aiming at supporting a collaborative working approach in the Extended Enterprise.

2. METHODOLOGIES FOR PILOT IMPLEMENTATION: SOA AND RELATED CHALLENGES

A deep literature review in the Knowledge Management domain [8]-[16] showed that very little attention has been given to the investigation of methodologies aiming to support the Pilot design and implementation task. On the mind of the authors, the Knowledge Management area still lacks of clear and commonly agreed guidelines to be used to guide the Pilot specification definition process. Due to the peculiar nature of the "element" to be tested, a Pilot project in the knowledge domain is more than just a proof-of-concept. It involves gathering requirements for the requested functionalities, setting the infrastructure and landscape, and technically configuring the solution. The results of the testing activity, moreover, are strictly dependant by the peculiar environment chosen for the trials, since human aspects play a big role in the successful implementation of such a working paradigm [1][17]. Configuring the best Pilot environment to run the trial is not an easy task. On one side, it's important to clearly understand what to measure, that means what functionalities of the KMS need to be tested. Since it's difficult both for users and process owners to communicate their expectations about the new system, specific tools and techniques need to be applied to formalize this "tacit knowledge". Although the use of a graphical modelling language can be helpful in order to record and share this information, the lack of a commonly accepted standard in this area can generate confusion and misunderstanding strong enough to slow down the implementation effort [18][19].
Moreover, once the functionalities to be replicated in the final solution have been determined, the Pilot design team is asked to translate these high level indications into a "physical" system. A lot of different Pilot implementation parameters have to be fixed during Pilot set up. Choosing the best value for each of them, in order to build the best trade off between reliability of data obtained and cost of implementation, can be a labor intensive and time consuming task. A general framework, giving indications about the way system performances are influenced by the particular testing environment chosen could be helpful, therefore, to support process stakeholders in correctly interpret information collected during the analysis.

3. WORKING APPROACH

The methodology presented in this paper tries to answer some of the questions underlined in the previous section, with the final aim to improve the quality of the Pilot implementation specification

process. From a “business” point of view, main scope is to help Pilot development team in saving time and money during the design and implementation of a KMS inside company’s walls. On one side, it allows clearly identifying most important functionalities to be implemented and validated in the prototype, reducing both the time needed to design the solution at the conceptual step and the risk to replicate and test non-crucial capabilities. On the other side it aims to reduce time needed to physically build the testing environment, providing a structured framework for the selection of the best implementation context on the basis of the specific purposes of the trials.
First, in order to improve knowledge elicitation and formalization during the conceptual design step, the authors analyzed in detail features and characteristics of most used process modelling techniques and tools, with the scope to identify new approaches to enhance requirements communication and sharing across the different teams participating to the Pilot development.
Second, in order to reduce time needed to configure the Pilot solution, the attention has been oriented towards the identification of standard Pilot implementation contexts to be automatically associated to the metrics used for the evaluation. Main assumption at the basis of this approach is that, among all possible combinations of implementation parameters, just a few of them really represent the best trade-off between quality of data obtained by the simulation and cost of the trials.

4. IMPROVING PILOT’S REQUIREMENTS IDENTIFICATION

Main aim of the prototype testing activity is to provide empirical data for final system tuning. This decision making process has to be based on solid assumptions and on reliable information in order to be successful. Since the Pilot will replicate just a sub-part of the final system, is very important to select and test only those “core” functionalities crucial for KMS success, which are the ones addressing most important customer requirements. The capability to clearly identify “real” user needs is at the basis of a well-targeted Pilot implementation. Main aim of the methodology at this step is, therefore, to propose new tools and techniques to be used to improve knowledge elicitation and sharing across the different teams participating to the Pilot development.
Traditionally, the Pilot specification definition activity takes the move from a high level representation of the final system architecture linked to a generic description of Pilot requested features. In order to better specify what functionalities need to be replicated in the Pilot solution, the development team is asked to elaborate and formalize a list of functional requirements describing more in detail the prototype from an IT perspective (Should-Be model). Since users and process owners are the real containers of the company’s knowledge [20], they are asked, therefore, to communicate their expectations and impressions regarding the new solution, mainly through interviews and questionnaires. However, their active participation in this phase is very often limited by the difficulties encountered both when communicating their vision about the new system and when analyzing IT expert proposals. Most important system requirements, especially during the first iteration step, can remain unexpressed, and there is a high risk that the entire system would be developed more according to consultant previous experiences and skills than on the basis of user expectations.

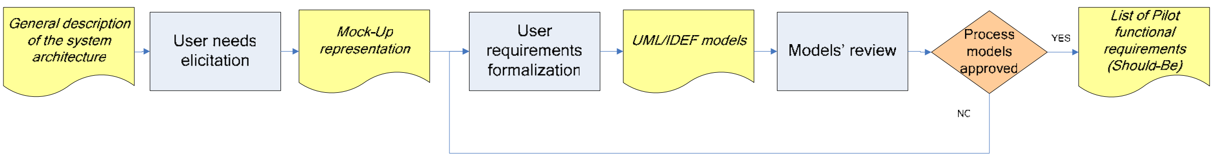

Figure 2: Methodology for Pilot functional requirements identification

The methodology (Figure 2) proposes the use of Mock-up representations to solve the lack of interoperability between users and managers and IT experts in the first stages of the Pilot requirements definition activity. Mock-ups represents a first attempt to build a prototype of the final Pilot solution; they consist of a sequence of slides, representing Pilot’s interface screenshots, showing the way users can interact with the KMS to solve their specific knowledge problems. The choice to rely on this kind of description is motivated by the need to give an intuitive and easily understandable description of how the Pilot solution would look like once implemented. The use of prototypes, in fact, addresses both the inability of many users to specify their information needs and the difficulty of systems analysts to understand the user environment, by providing the user a tentative system for experimental purpose at the earliest possible time [21]. The adoption of Mock-up representations in the early design

phases facilitates, therefore, the requirements elicitation process, since this method doesn't require particular skills in the process modelling domain. People who are not modelling experts, in fact, can be disoriented, in the early stages of the Pilot development, by the intrinsic complexity of traditional Enterprise Modelling languages [20].
Later on, moving towards the definition of the Pilot solution at a more technical level, Mock-ups need to be linked to traditional process models in order to go deeper in detail with the description of Pilot functionalities. Modelling tools to be used at this step require, in fact, a more rigorous structure and formalization, since they represent the system very close to the implementation level. The methodology proposes to support the functional requirements identification activity making an integrated use of UML [22] and IDEF [23] representations. The choice to use UML is motivated mostly by the fact that it is very popular among IT domain experts because it developed a software design oriented semantic and syntax [18], which means that graphical symbols and relations in the models are concisely defined, and their meaning can be unambiguously interpreted. This kind of description provides, therefore, to people with right requisite, knowledge and skills, a very complete view of the company processes, particularly focalized on the IT aspects [24]. Translating user expectations, however, it's not an easy task. Sometimes technological barriers are too difficult or costly to overcome and previously identified user requirements need to be revisited. Several numbers of iteration loops are usually needed in order to re-align user expectations and technical constraints. For this reason, since Pilot stakeholders are asked to continuously interact for the system development, the intrinsic formality of UML can be disorienting for people who are not technicians or modelling experts [24]. They would prefer to share knowledge using a tool they already know, to get more familiar with the KMS architecture. An integrated use of UML and IDEF models, the first mainly IT oriented and the second well known also by people without a technical background, is therefore proposed by the methodology to improve knowledge formalization and sharing during the requirements identification activity. The integrated use of Mock/up and UML/IDEF diagrams allow, moreover, introducing a sort of Concurrent Engineering on new process definition, which helps users and process owners to be much more involved in system development and consultants to build a solution really able to address users' needs.

5. DEFINITION OF EVALUATION METRICS

Once Pilot requirements have been consolidated and approved, an evaluation metrics need to be defined in order to understand what parameters will be measured during the Pilot testing phase.

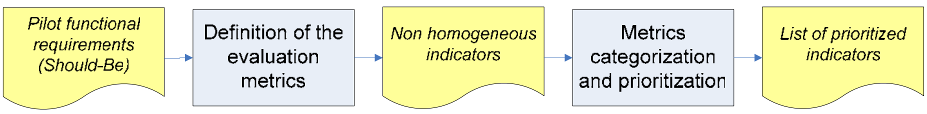

Figure 3: Definition of evaluation metrics

Evaluation metrics is initially built on the basis of the specific functional requirements expressed in the previous step. First output of the metrics specification activity is a list of non homogenous indicators, expression of the needs and of the expectations of those process stakeholders internal to the company. The term "non homogenous" refers to the fact that, due to the intangible nature of the knowledge resource [25], these indicators differ in terms of relevancy and granularity of the feedback provided. Several approaches to measure the impact of KM initiatives have been developed in the past and they include, for instance, the House of Quality (Quality Function Deployment or QFD), Balanced Scorecard, the American Productivity Center (APQC), Skandia Navigator, IC index, and Intangible Assets Monitor [26][27]. These methods focalize on different aspects of the evaluation process and propose their own way to translate high level strategies into real targets. In order to provide a general purpose methodology and to solve this lack of standards, the authors rely on the classification proposed by the OMG group [28], based on three different abstraction layers (CIM-PIM-PSM) to classify indicators aiming to measure Pilot performances at *Business, Process, Technological* level.

Business. Measuring the business value of KM initiatives has become imperative to ascertain if the expectations are realized [25][27]. Costs indicators can be used, therefore, in order to evaluate the impact of the new system on business outcomes [29]. The implementation and usage of the new KMS have a deep impact, in fact, on overheads and financial costs, as well as on the usage of resources in the development process (material and labour costs). Indicators populating the business layer, in synthesis, are conceived in order to provide long term feedbacks about company competitiveness and business results [30].

Process. Considering the intangible nature of the knowledge resource, it is extremely difficult to show an absolute one-to-one correlation between a knowledge-related action and a monetary flow. In order to gain a clear understanding of the Knowledge Management approach adopted, it could be worth, instead, to focus on product development process success factors such as innovation, quality or cycle time [27][31]. Performance indicators at this level allow measuring directly the impact of the new solution on the design process, which means how the new collaborative working paradigm would impact on everyday activity flows. KMS effectiveness can be measured through, for instance, the effective use of the system along the design process, its frequency of use, the degree of usability of information obtained, or, for instance, the way in which "knowledge elements" have been formalized and updated, and the way in which different "knowledge sources" interoperate [29].

Technological. At technological level, new process advantages are strictly linked to the particular architecture and configuration chosen for the Pilot solution. These advantages refer mainly to two categories. They can be obtained increasing the quality of the results of each single design activity, and, at the same time, reducing the time needed to obtain these results. At technical level the focus is shifted from the design process to a single activity or sub-activity. At this level, measurements are much les influenced by the social and behavioural aspects of Knowledge Management [32] and for this reason quantitative data are much more meaningful than qualitative feedbacks.

Since the metrics definition activity usually brings to the identification of a huge number of candidate indicators to be measured during the trial, they need to be refined and ranked in order to setup only the metrics that would be most valuable. The Pilot solution should be built taking into consideration what effectively it's needed to be known from the testing activity, which means to make use of a few focused measures, aligned to strategic objectives. It means that not all the indicators have the same important and weight in the contest of the new system implementation. Some indicators are more critical for company's success, some are not. Too many metrics simply mean that most would be ignored in practice [33].

6. CONFIGURING STANDARD PILOT ENVIRONMENTS

Having the right information (which in this case means "knowing what to measure") is not enough to take right decisions. Information, in fact, needs to be processed in the mind of individuals to become knowledge and to guide the decision-making process. Since the Pilot replicates just a part of the KMS, it is far to be a perfect mirror of final system performances, data obtained during the trials need to be correctly interpreted to guide the final tuning process and their reliability has to be carefully judged. Numerical values can be easily misinterpreted and the risk to take the wrong direction is very high. Measurements are strongly dependant, in fact, by the particular implementation environment chosen. More the Pilot would be close to the final implementation, more it would provide reliable feedbacks. However, very often such an approach is not feasible, due to time and cost constraints. What the Pilot development team is asked to do, therefore, is to determine the best trade-off between data reliability and cost of implementation. Setting up the Pilot environment, for this reason, can be a complex and labour intensive task. The design team is asked, in fact, to fix a huge number of different implementation parameters in order to configure the testing environment correctly. All their possible combinations can generate, in fact, a never-ending likelihood of possible Pilot's configurations. Main idea of the methodology is, therefore, to define most meaningful combination of these implementation parameters. It proposes a set of guidelines to guide this specifications' definition process, in order to reduce the time needed to design and implement a Pilot system in the Knowledge Management area by providing a framework to select, the most reliable and "cheapest" implementation environment on the basis of the peculiar objectives of the trials.
Driving hypothesis of the work here presented is that the definition of Pilot implementation contexts is fundamentally metrics driven [34]. When the IT expert team, in fact, has to choose in what way the Pilot will be physically implemented, it has to keep well in mind what to obtain from the evaluation

process. It means that the Pilot solution has to be built in order to make feasible the measuring and interpretation of those interesting parameters selected in the metric's definition task.

6.1. Grouping Pilot implementation parameters

The work moves from the assumption that, even if Pilot performances can be influenced by a lot of factors, just a few of them really impact on the reliability and quality of the data measured during the trials. As stated by many authors, in fact, in any series of elements to be controlled, a selected small fraction, in terms of number and elements, always accounts for a large fraction in terms of effect. This *Pareto* principle [35] (or, more correctly, *Lorenz* principle [36]) is commonly used in marketing and sales contexts, customer complaints, in quality control and manufacturing deficits and in other business settings. On the mind of the author it can be applied also in the knowledge domain, and more specifically to support and guide the definition of Pilot implementation specifications. Depending on the scope of the KMS implementation and on its impact on the company's environment, a few "optimal" standard Pilot configurations can so be identified.
The definition of these classes of implementation parameters has been based both on a deep literature review [37]-[40] and on the previous experience matured by the authors in the frame of VIVACE. This activity brought to the identification of 7 different families representing main dimensions on which the Pilot development team is asked to reason about before starting with the physical Pilot implementation. These implementation variables and their values are summarized in Table 1.

Table 1. Pilot implementation variables

Environment	*Real*	Pilot is tested in an everyday working environment
	Laboratory	Pilot is tested in a protected environment
People	*Final users*	System is tested directly by end users
	External testers	System is tested by people external to the organization
Conditions	*Local*	Small group of people in the same place/company
	Global	Large group of people in different locations/companies
Instruments and tools	*Ad Hoc*	Most of HD and SW tools are designed and implemented ad hoc
	Available	Most of HD and SW tools are already available inside the company
Test duration	*Long*	Several months
	Medium	2-3 months
	Short	less than 30 days
Data managed	*Real*	The Pilot is filled with real design data
	Fuzzy	Unrealistic data are used for security reasons
Performance controlling methods	*Periodic*	Pilot perfomances are evaluated periodically
	Step by step	Pilot perfomances are evaluated at the end of each simulation stage
	Continous	Pilot perfomances are evaluated in a continous way, for each sub-activity

Initially, the Pilot development team needs to determine if the prototype would run in an everyday working environment or in parallel with the normal design activity flows (Environment). Moreover, it is asked to assess if it would be tested by end users themselves or by people external to the organization (People). A decision must be taken about the possibility to perform the simulation "locally", in a single company with a small group of people, or "globally", involving several users in different locations. Moreover, it's very important to decide if HD and SW tools need to be strongly customized and implemented ad hoc or if already available in-house technologies can be configured to run the simulation (Instruments and tools). It is also necessary to establish the duration of the testing period (Test duration) and, for security reasons, Pilot stakeholders are asked to decide if run the simulation using fuzzy data instead of real ones (Data managed) Finally, it is very important to determine the Performance Controlling Methods (PCM) to be applied. Pilot performances can be monitored in a continuous way, step-by-step, or periodically.

6.2. Definition of a set of standard Pilot configurations

Starting from the CIM-PIM-PSM framework, the authors propose three different layers to categorize and group the implementation parameters described in the previous section. Pilot projects can be developed, in fact, to evaluate the KMS effort from a *Business*, *Process* and *Technological* point of view. If the new collaborative working approach is thought to have a great impact all over the company's departments and not only on the design process, the implementation effort would refer to the *Business* layer. If the KMS, instead, is conceived to bridge the gap between different design teams, without involving other company's functions, the *Process* level must be taken in consideration. If the improvement effort is focalised on a specific design process phase, it mainly addresses the *Technological* layer.

Business. Business outcomes are often the hardest measures to evaluate, particularly because of the intangible nature of knowledge assets [29]. Determining the impact on the organization is not easy. Since the social and behavioural aspects of Knowledge Management are the most important drivers for a successful KMS implementation in the long-term, it requires a long maturation process to assess the benefits of the proposed approach, and a short test usually doesn't provide high quality indications about expected impacts [6]. Reliable feedbacks regarding the benefits and improvements of the new system at a business level can be obtained only if the Pilot can be tested under everyday working conditions, final users and real data. Configuring such a testing environment allows measuring the long wave effect of the new working paradigm in all the departments involved. It requires a careful planning and development of the ad-hoc technologies to be tested, and the results of the simulation must be carefully monitored along all the testing period. Higher costs associated to the Pilot development and implementation are mitigated by the possibility to avoid unexpected costs after the real implementation. Company wide testing of KMS, however, can be risky. If problem arise in the piloting phase, the performance of the engineering departments may suffer severely

Process. If the main aim of the testing activity is just to obtain mainly process-related measurements, implementing the Pilot too close to the final KMS can be too costly and time consuming for the purpose of the trials. Process performances, in this case, are less influenced by the peculiar environment's conditions in which the testing solution will be implemented. Reliable performances measurements can be obtained also running the Pilot in parallel to the design process without interfering with design team's work. However, only final users have the capabilities to understand if the system would be able to improve design process performances in the "real world", so it would be preferable to test it with internal employees and real data. The possibility to run the simulation in an isolated environment it's, moreover, very helpful in order to check system performances in a more rigorous way (step-by-step) and to reduce Pilot testing time.

Technological. When the aim of the KMS is just to provide local improvements (typically integrating already existing tools) it's preferable to run the prototype in an isolated environment by external testers to reduce the impact of the trials in everyday activity flows. Indicators to be measured are mainly quantitative and strictly linked to software and hardware performances, poorly dependant by the particular testing environment chosen. In this case, the testing activity's impact on the design process can be strongly mitigated. In order to reduce costs and testing time, the choice to run the trial in an isolated environment seems to be the more cost effective. This configuration would allow, moreover, performing easier and continuous performances' checks. Moreover numerical values obtained at this level are independent also by the nature data used. For security reasons, fuzzy data should be used during the test.

These three standard contexts represent, on the mind of the author, the best trade-off between data reliability and Pilot cost when dealing with a KMS implementation mainly focused on one of the three layers identified. However, the level of implementation of the KMS cannot be always assessed in these terms. Since many different process stakeholders are interested in evaluating the KMS from their own point of view and feedbacks at different level of abstraction may be required, it's not always so easy to determine what kind of Pilot configuration need to be selected. For this reason, two standard Pilot implementation contexts have been developed and added to the initial classification, in order to provide useful guidelines also in the case in which the KMS implementation effort couldn't be easily classified (Figure 4).

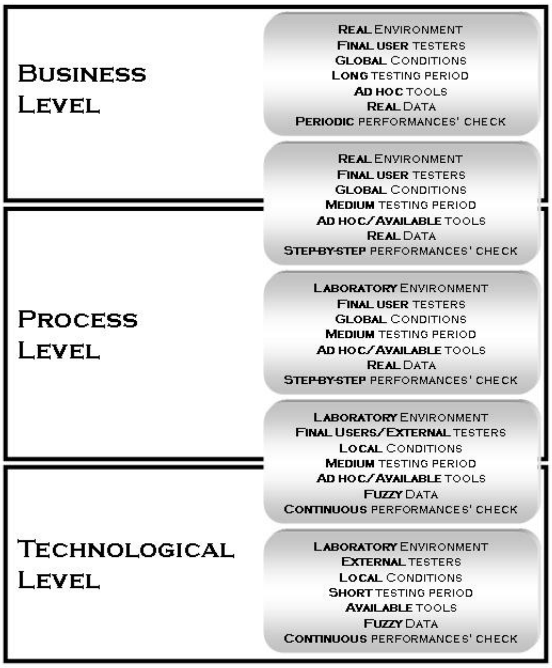

Figure 4: Standard Pilot implementation contexts

7. DEFINITION OF THE PILOT IMPLEMENTATION CONTEXT

Once metrics indicators have been ranked and prioritized, the "optimal" Pilot environment to be set up for the trial can be obtained in a semi-automatic way as the one closest to the mean abstraction level of the prioritized indicators. In order to make clear this mapping process, an example of metrics to be used for the Pilot validation activity is shown in Table 2.

Table 2: Example of metrics to be applied for Pilot evaluation

Labour cost for product development	[€/hour]	Business
Frequency of use of the collaboration tool	[n°/month]	Process
Number of knowledge elements created per month	[n°/month]	Process
Number of different design options investigated	[n°]	Process
Number of iterations in a repetitive process	[n°]	Technical
Search time for a specific design decision rationale	[hour]	Technical
Time needed to perform a specific simulation	[hour]	Technical
Time needed to perform a simulation by a newcomer	[hour]	Technical

These indicators represent only those most important success factors for the implementation under analysis. Since it is not feasible, in fact, to measure all the parameters resulting form the metrics definition activity, consultants and process owners are asked to generate and synthesize a list of "prioritized" indicators to be measured during the trials. In this example, it can be seen as half of these indicators is related to a technical dimension, since they are weekly dependant by the way the product development process is organized. Some other measurements, like the frequency of use of the collaboration tool, are strongly related to the behavioural and social aspects of Knowledge Management and to the way in which the design process is configured. A lot of business indicators have been also proposed, but just the Labour cost has been considered relevant to assess product development process' improvement in the long term. The classification proposed by the methodology orient process stakeholder's choice towards the correct implementation environment to chose. From Table 2 it emerges clearly that the focus is much more oriented towards the evaluation of *Technological* and *Process* performances, than on *Business* ones. Since measurement at technical level are poorly influenced by the particular boundary conditions of the testing environment, the methodology suggest therefore to deal with an isolated and controlled trial, run in parallel with the design process by external testers in order not to slow down normal activity flows. However, in order to evaluate the long term effect of the new working approach, Pilot duration time needs to be extended and final users must be involved in the experimentation to obtain reliable feedbacks also at a process dimension. Figure 5 provides a graphical example of the way the methodology is conceived and applied to determine the best Pilot implementation configuration.

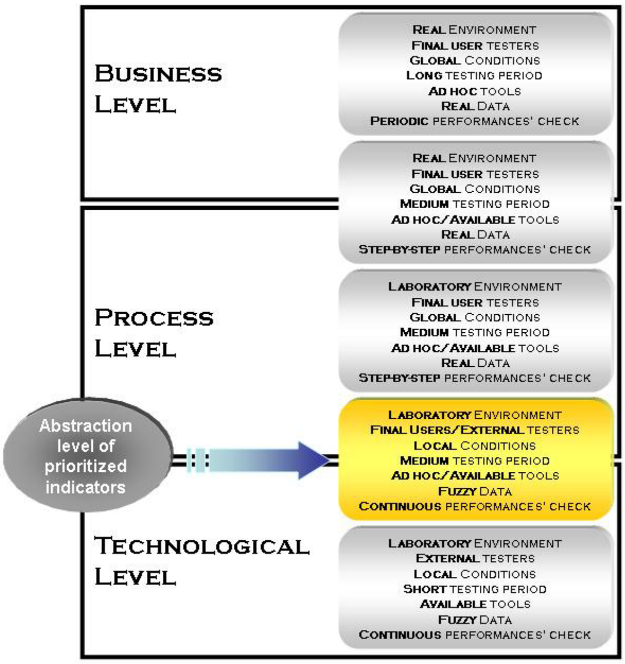

Figure 5. Definition of the Pilot implementation context

8. RESULTS AND OPEN ISSUES

The methodology here presented originated from the work done in the frame of the European project VIVACE for the implementation of a KEE solution in the aeronautical domain. The proposed approach aims to support consultants and process owner in the definition of Pilot implementation specifications when dealing with the design and implementation of a new Knowledge Management System. The proposed approach is based in two main assumptions. On one side, to take "good decisions" for final system tuning, a clear understanding of the problem to be addressed is needed. It means that those functionalities to be replicated and tested, making use of the prototype, have to be correctly and unambiguously identified. On the other side, to reduce time for Pilot implementation, a general framework is proposed to suggest how to design the Pilot implementation environment to obtain the best trade-off between data reliability and cost of the prototype.

The methodology focused, on one side, on process modelling languages, providing a method to improve communication and knowledge sharing among the participants of the Pilot development activity, increasing users' and process owners' involvement in Pilot design. Since the KMS development involves people from different background, the use of Mock-up representations together with traditional modelling techniques showed to be successful in order to enhance common understanding and agreement on Pilot specifications and to link high level user requirements to the technicalities of the Pilot solution. The integration of different process modelling languages and the development of specific tools for a more effective and interactive prototyping are interesting areas to be explored to improve and refine the methodology in such a domain.

On the other side, the methodology proposes a general framework to speed up the decisional process when physically configuring the Pilot solution. Five different standard Pilot implementation contexts have been elaborated and proposed as optimal trade-off between data reliability and cost of implementation. Depending on the purpose of the implementation effort, defined mainly by the mean abstraction level of the metrics to be used during system evaluation, these standard contexts provide a basis to support the decision making activity regarding the way the trial will be physically configured and run. One of the advantages of this approach is also related to the fact that the Pilot development team can know in advance what they can expect from the particular Pilot configuration chosen, in terms of reliability and quality of the numerical values produced and time needed to obtain them. The five standard Pilot contexts provide also a means for correctly interpret the results of the simulation process. Further research is needed to verify deeply in detail the effectiveness and reliability of the proposed approach. It could be worth to apply it in several study cases in order to develop this "conceptual framework" into a sort of "technical manual" or guide able to predict how the physical system would behave depending on the particular standard context chosen.

REFERENCES

[1] Alavi, M. and Leidner, D.E., Knowledge Management Systems: Issues, Challenges, and Benefits, In *Communications of the AIS*, 1999

[2] Malhotra, Y., Why Knowledge Management Systems Fail? Enablers and Constraints of Knowledge Management in Human Enterprises, in *Handbook on Knowledge Management*, 2002

[3] Akhavan, P., Jafari, M. and Fathian, M., Exploring Failure-Factors Of Implementing Knowledge Management Systems in Organizations, In *Journal of Knowledge Management Practice*, 2005

[4] Akhavan, P., Jafari, M. and Fathian, M., Critical Success Factors of Knowledge Management Systems: A Multi-case Analysis, in *European Business Review*, 2006

[5] Gallupe, B., Knowledge Management Systems: Surveying the Landscape, In *International Journal of Management Reviews*, Vol. 3, 2001

[6] Presley, A. and Liles, D., RD Validation Planning: a Methodology to Link Technical Validations to Benefits Measurement, in *R&D Management*, 2000

[7] *VIVACE Project web site*, http://www.vivaceproject.com.

[8] Levett, G.P. and Guenov, M.D., A Methodology for Knowledge Management Implementation, In *Journal of Knowledge Management*, 2000

[9] Davenport, T.H., De Long, D.W. and Beers, M.C., Building Successful Knowledge Management Projects, In *Managing the Knowledge of the Organization*, 1997

[10] Wiig, K., de Hoog, R. and van der Spek, R., Supporting Knowledge Management: a Selection of Methods and Techniques, In *Expert Systems With Applications*, 1997

[11] Gersting, A., Gordon, C. and Ives, B., Implementing Knowledge Management: Navigating the Organizational Journey, in *Knowledge Management*, 1999

[12] McLure, W.M., A Framework for Successful Knowledge Management Implementation, In *Proc. of the Fourth Americas Conference on Information Systems, AIS'98*, 1998, pp 635–637

[13] Sunassee, N. and Sewry, D., A Theoretical Framework for Knowledge Management Implementation, In *Proc. of the Annual Research Conference of the South African institute of computer scientists and information technologists on Enablement Through Technology*, SAICSIT'02, Port Elizabeth , 2002, pp 235-245

[14] Grimàn, A., Roias, T. and Perez, M., *Methodological approach for developing a KMS: A Case Study*, 2002 (Departamento de Procesos y Sistemas, Universidad Simon Bolivar, Venezuela)

[15] Hahn, J. and Subramani, M.R., A Framework of Knowledge Management Systems: Issues and Challenges for Theory and Practice, In *Proc. of the 21st International Conference on Information Systems, Brisbane*, 2000, pp302-312

[16] Wong, K.Y., Critical Success Factors for Implementing Knowledge Management in Small and Medium Enterprises, In *Industrial Management & Data Systems*, 2005

[17] Stoddard, D.B.and Jarvenpaa,. S.L., Business Process Redesign: Tactics for Managing Radical Change., in *Journal of Management Information Systems*,1995

[18] Kalnis, A., Business Modelling. Languages and Tools, *Progress in Industrial Mathematics at*

ECM, 2004, pp41-52
[19] Vernadat, F.B., *Enterprise Modeling and Integration: Principles and Applications*, 1996 (Chapman & Hall, London).
[20] Liou, Y.I., Knowledge Acquisition: Issues, Techniques, and Methodologies, in *Proc. of the ACM Conference on Trends and Directions in Expert Systems*, Orlando, 1990, pp212-236
[21] Janson, M.A. and Smith, L.D., Prototyping for Systems Development: A Critical Appraisal, in *MIS Quarterly*, 1985
[22] *UML web site*, http://www.omg.org/uml.
[23] *IDEF web site*, http://www.idef.com
[24] Kim, C.H., Weston, R.H., Hodgson, A. and Lee, K.H., The Complementary Use of IDEF and UML Modelling Approaches, in *Computers in Industry*, 2003
[25] Glazer, R., Measuring the Knower: Towards a Theory of Knowledge Equity, in *California Management Review* , 1998, pp175-194
[26] Kankanhalli, A. and Tan, B.C.Y., A Review of Metrics for Knowledge Management Systems and Knowledge Management Initiatives, in *Proc. of the 37th Hawaii International Conference on System Sciences*, HICSS'04, Hawaii, 2004
[27] Vestal, W., Measuring Knowledge Management, *an American Productivity and Quality Center white pape*r, 2002
[28] The Object Management Group Website, http://www.omg.org
[29] *Measuring knowledge value? Evaluating the impact of knowledge projects*, A Knowledge and Innovation Network white paper, 2002
[30] Metrics Guide for Knowledge Management Initiatives, 2001 (Department of the Navy Chief Information Officer, Washington)
[31] Schultze, U. and Boland, R.J., Knowledge Management Technology and the Reproduction of Knowledge Work Practices, In *Journal of Strategic Information Systems*, 2000
[32] Ellis, J., *The role of knowledge management in enabling competitive advantage for Australian organisations in the global economy*, 1997 (Royal Melbourne Institute of Technology, Melbourne)
[33] *Metrics for Knowledge Management and Content Management*, 2003 (Steptwo, Australia)
[34] Fui-Hoon Nah, F., Janet Lee, J., Nah, F.F.H. and Lau, J.L.S. Critical Factors for Successful Implementation of Enterprise Systems, In B*usiness Process Management Journal*, 2001
[35] Craft, R.C. and Leake, C., The Pareto Principle in Organizational Decision Making, In *Management Decision*, 2002
[36] Juran, J.M., *The Non-Pareto Principle*; Mea Culpa, 2001 (Juran Institute, Southbury)
[37] Stoddard, D.B. and Jarvenpaa, S.L., Business process redesign: tactics for managing radical change, in J*ournal of Management Information Systems*, 1995
[38] Pikosz, P., Malmström, J. and Malmqvist, J., Strategies for Introducing PDM Systems in Engineering Companies, In *Proc. of the 4th International Conference on Concurrent Engineering*, ISPE'97, Rochester, USA, 1997
[39] Liebowitz, J., Knowledge Management and its Link to Artificial Intelligence, In *Expert Systems with Applications*, 2001
[40] Roth, N., Lettice, F. and Evans, S., Measurement of Knowledge Reuse and Invention in New Product Development. In *Proc. of 8th International Conference on Concurrent Engineering,* ICE'02, Rome, June 2002

Contact: Marco Bertoni
Politecnico di Milano
Department of Mechanical Engineering
Via La Masa 34, 20156, Milano
Italy
Tel: +39 02 2399 8094
Fax: +39 02 2399 8202
E-mail: marco.bertoni@polimi.it
URL: http://www.kaemart.it