Experience Reuse to Improve Agility in Knowledge-Driven Industrial Processes

V. M. Llamas[1,2], T. Coudert[1], L. Geneste[1], J. C. Romero Bejarano[2], A. de Valroger[2]
[1]Laboratoire Génie de Production / ENIT – University of Toulouse, Tarbes, France
[2]Axsens bte. Toulouse, France
({vllamas, thierry.coudert, laurent.geneste}@enit.fr; {juan.romero, aymeric.devalroger}@axsens.com)

Abstract - **Companies need to become more agile to survive to the unstable and highly changing market-place. This can be achieved through the adaptation and control of their business processes. A process sufficiently structured but not over constrained by standards and based on experience feedback principles is necessary. This article describes a proposition of agile process driven by the reuse of experiences and knowledge. For this purpose, based on Case-Based Reasoning (CBR) principles, the complete lifecycle of an agile process is introduced, from requirements definition, retrieval, reuse, adaptation, and storage steps. Finally, an example applied to the domain of industrial problem solving is presented to illustrate the methodology.**

Keywords – **Industrial Processes, Agility, Knowledge Management, Experience Feedback.**

I. INTRODUCTION

One of the major challenges for today's organizations is to be able to continuously and rapidly adapt, in order to face significant changes in the market-place over the world. In most companies, this is achieved through re-structuring and controlling their processes.

The evolution of the concept of *process* until these days is described in [1]. Their study includes influencing people that modified the vision of processes and consequent methods that brought processes' progress. Process management has progressed from Lean, to BPR (Business Process Reengineering), to BPM (Business Process Management) in the last 25 years. An approach for agile enterprises, operating in a dynamic and complex environment, to adopt the high-level architecture modelling standard ArchiMate combined with different low detailed level modelling standards (e.g. BPM, UML) is described in [2].

Regardless the continuous adaptation and renewal of methods and techniques to improve business processes in today's organizations, some lacks of agility emerge. Then, the question that arises is: How to achieve process agility? Enterprises continuously perform several types of processes (e.g. product design, production, maintenance, problem solving, supplier selection), whose performance could be improved over time by sharing and reusing lessons learned and experiences [3]. Therefore, this article intends to complete the agile approach introduced in [4] to improve business processes by setting the basis for agile knowledge-driven processes.

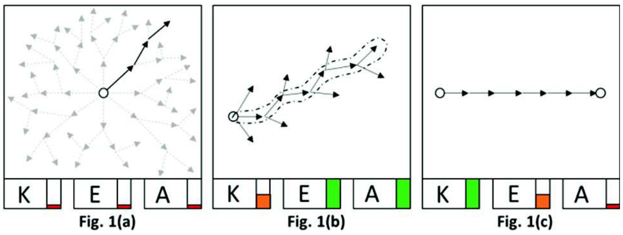

Fig. 1. Generic processes structuration.

This paper focuses on defining the capitalization and reuse of general knowledge and experiences as drivers to improve agility. General knowledge is composed of best practices, procedures, rules, etc. [5] defined through a set of situations that establishes a standard, or a formalized manner to perform a process. An experience describes a task performed in a particular context. Once capitalized, an experience is a piece of contextualized knowledge.

In order to illustrate and to introduce the agile approach, three types of generic processes are shown in Fig. 1. In Fig. 1(a), a totally non-structured process, and in Fig. 1(c), a standard fixed enterprise process (structured) is represented. The third case, Fig. 1(b), corresponds to an agile process. The level of knowledge (K) and experiences (E) capitalized and reused in each one of the three processes, and the level of agility (A) are represented in the lower part of Fig.1.

Fig. 1(a) represents a non-structured process. This type of process presents a high level of flexibility that allows readjustments through different alternatives. When facing a disturbance, the process can be reconfigured to reach the objectives, for instance, new activities can be added in real time to overcome problems. However, this high level of flexibility involves a low level of formalization, it is quite difficult to define standards for its systematic reuse. Then, formalization and reuse of general knowledge and experiences is difficult to achieve, their capabilities are very low, and consequently also is agility.

At the opposite, the situation in Fig. 1(c) describes a structured process. The process is formalized as a set of pre-defined activities, allowing its systematic reuse. The advantages are both, that activities can be performed without ambiguities, and that knowledge can be formalized and reused to help decision makers. However, in such a type of process, when unexpected events or disturbances occur, it is very difficult to change and to react. Furthermore, when overcoming non-formalized scenarios (i.e. new activities performed out of the standard), the standardized available knowledge can be inadequate. A high level of knowledge is capitalized due to standardization. Also, experiences capitalization and reuse only concern activities because decision-making is already standardized through general knowledge. Hence, there is no agility in the process to react to problems.

Therefore, both non-structured and structured processes present drawbacks. Consequently, a third approach is proposed in Fig. 1(b): an agile process. An agile process is a flexible approach driven by the continuous reuse of general knowledge and experiences, through the combination of both previously described extreme situations. Thus, drawbacks from both structured and non-structured processes are taken into account to define an agile process that:

- is sufficiently structured to ensure objectives satisfaction and process efficiency but not over constrained by standards,
- can be reconfigured and adapted to unexpected situations,
- is based on experience feedback principles (i.e. the process is driven by knowledge and experiences reuse and permits to learn new knowledge and experiences during its execution).

In order to apply both agile thinking and knowledge and experiences capitalization and reuse to business processes, related works concerning agility and knowledge/ experiences reuse concepts are presented in the next section. It leads to the definition of the contributions addressed in this paper. In section III, the complete lifecycle of an agile process is presented. In section IV an illustrative example of a problem-solving agile process is presented. Finally, the conclusion and the perspectives of this work are presented in the last section.

II. RELATED WORKS

A. Agility concepts

The concept of agility has been studied in different application domains: business agility, enterprise agility [6], agile organization, agile workforce, IT agility, agile manufacturing, agile supply chains [7] and agile software development [8]. However, there is no general consensus on a definition of agility [9].

Two of the most discussed agility's subjects in literature are agile software development methods and agile manufacturing, briefly introduced in this section.

In 2001, the Agile Alliance was created introducing formally agility through the Agile Manifesto. It outlines values and principles common to all agile software development methods [10]: individuals and interactions should be valued over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan [11]. The general principles these methods introduce are the flexibility and adaptability face to changes in requirements through the project. It means that, using agile practices, the developer can easily modify the code to respond to changes of the requirements without major losses for the project [12]. Two of the most spread methods are Extreme Programing and SCRUM [10], [13].

From another perspective, the concept *Agile Manufacturing* was introduced by a group of scholars of the Iacocca Institute of Lehigh University in 1991 to provide clarifications on the causes of new conditions in business at the time [14]. There is no unified definition of Agile Manufacturing nor of its core concepts. Agility is defined as "*the ability to cope with unexpected changes, to survive unprecedented threats of business environment, and to take advantage of changes as opportunities*" in [14].

B. Knowledge and experiences capitalization and reuse

The concepts of knowledge and experiences capitalization and reuse have been studied and applied to several domains, such as workflow adaptation [15], project memory reuse [16], and continuous improvement [3].

The continuous capitalization and reuse of experiences and knowledge all along a process is a major challenge to achieve agility in processes.

A distinction between knowledge and experiences is used in this article. According to [17] knowledge is a *more subjective way of knowing and is typically based on experiential or individual values, perceptions and experience.* In [5] the hierarchy data, information, experience, knowledge is explained: information corresponds to an event along with its context; an experience permits to formalize analysis and solution. Finally, when lessons learned, procedures, rules etc. are implied from past experiences, knowledge is obtained.

Then, for the purpose of this article, experience is every piece of understanding from a previous situation that can be capitalized in order to be reused in the future. Knowledge is also a piece of understanding, but its level of generalization is higher. Knowledge has already been validated by knowledge experts.

Experience feedback is a structured process of capitalization and exploitation of information extracted from the analysis of positive and/or negative events [3]. According to [18], six steps define the experience feedback process: Collecting experience, modeling experience, storing experience, reusing experience, evaluating experience and, maintaining experience.

Case-based reasoning (CBR) is a method that reuses previous experiences to solve problems, through the following cycle: *retrieval* of most similar cases to the current problem, *reuse* of information from the retrieved case to solve the current problem, *revise* the proposed solution and *retain* the parts of the experience for future problem solving [19].

C. Contribution

Existing agile approaches do not propose a unified and complete experience and knowledge-driven process. Then, this article intends to include CBR principles in order to propose an *agile process based on the capitalization and reuse of experiences and knowledge.*

III. DEFINITION OF AN AGILE PROCESS LIFECYCLE

This section describes the complete cycle of an agile process, based on five steps.

Fig. 2 shows the complete lifecycle of an agile process: 1/ requirements definition, 2/ search of past cases in the Experience base (EB) and/or general knowledge in the Knowledge base (KB), 3/ definition of the process' first version, 4/ process on run-time/adaptation, and 5/ storage in the Experience or Knowledge base. All five steps are described in the next subsections.

A. Requirements definition

First, requirements are defined. All stakeholders submit their objectives and constraints for the process. Stakeholders include: process architect (see section III.B), customer, internal customer (quality manager, accounts, etc.), project/process manager, etc.

Each stakeholder has different requirements regarding the process, such as the cost, delay, involved resources, the structure of the process itself, etc. Thus, all requirements (including objectives and constraints) have to be clearly defined, along with their owner (the person that sets the requirement), during this first step.

B. Search of past cases in the EB and/or general knowledge in the KB

Once constraints and objectives are defined, the process architect searches into the EB/KB for similar cases and suitable knowledge. The process architect is in charge of the definition of the first version of the agile process and of managing its lifecycle.

The knowledge base is a structured collection of pieces of knowledge. For the purpose of this paper, KB contains a set of different types of models of processes (e.g. problem-solving-9S process, maintenance process). Knowledge is defined from standards (e.g. problem-solving processes such as, 9S process, Six-Sigma /DMAIC). Knowledge can be obtained by generalizing experiences [18], but it is important to notice that its formalization is not described in this article.

For each type of process model stored in KB, several instances (i.e. experiences) may exist in the EB, i.e. for a standard process, each time that it is performed (even, it is modified), it is stored in the EB along with its context, indicators and parameters (e.g. process $9S_1$, process $9S_2$). All experiences corresponding to the same process model are gathered into a specific set of experiences (SoE) in the EB (see section III. E).

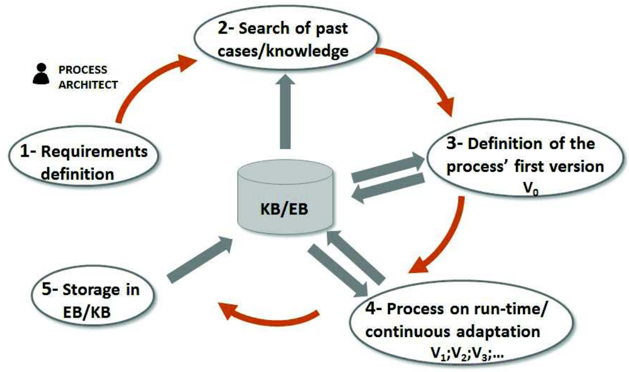

Fig. 2. Lifecycle of an agile process.

EB contains all the agile processes that have been executed, either from scratch, or by modifying a standardized process.

In order to search knowledge in the KB, and similar past cases in the EB, a simplified system of tagging is used, using taxonomical similarity [20]. All process models capitalized in the KB are given with a set of tags concerning the type of process. Agile processes stored in the EB are tagged according to: the type of process, process objectives, process context and stakeholders' constraints. Therefore, once the first step of requirements definition is over, for each new agile process its own characteristic keywords are defined. Thus, when a search needs to be done, first, current process' tags are compared with existing ones in the KB to find a suitable process model. Second, current process' tags are compared with existing ones from the corresponding SoE in the EB to find similar experiences. Consequently, a set of options (including knowledge and experiences if both are available) is proposed to the process architect so that s/he can define the first version of the agile process.

C. Definition of the process' first version

The third step is the definition of the first version of the agile process (V0) before its launch. The process' architect takes into account the information obtained in steps 1/ and 2/ to propose a first version of the process including different options, as illustrated in the left side of Fig. 3. Each option corresponds to a possible choice defined either from past experiences, or by the process architect.

V0 is composed of activities, processes and decision-making points, defined a priori.

Along with the outlining of V0, the process' architect defines and estimates the indicators for the agile process (e.g. performance, risk, cost, delay).

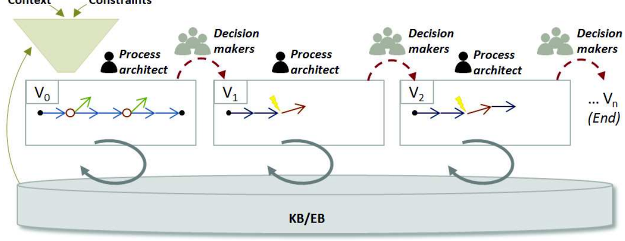

Fig. 3. Evolution of an agile process' versions.

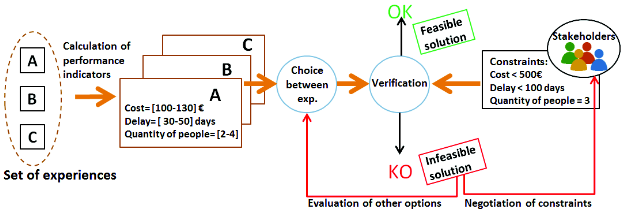

Fig. 4. Indicators and constraint negotiation.

The concept of constraints negotiation is introduced at this stage. If an option of a process is going to violate one of the constraints, the architect (or the user in run-mode) can contact the stakeholder who is owner of the constraint and negotiate its release, as shown in Fig. 4.

D. *Process on run-time and continuous adaptation*

During the process run-mode, at each decision-making point, the way forward is decided. Users perform the process using the first version V0 as a guideline. When a decision is reached, inputs (EB/KB, context and constraints) are considered in order to decide which option to take next. Decision makers can also propose and implement options not provided in V0. Moreover, modifications can be inserted in real time to face potential unexpected events that can occur.

Every time a decision involves a change to the previous version, a new version (V_{n+1}) is created, as illustrated in Fig. 3. Once the way forward is decided, each decision result is stored in the EB.

All along the agile process, but especially at decision points, EB and KB are accessed, searching for similar previous situations and knowledge. Also, once the decision has been made, its result is stored into the EB for future reuse.

Constraints negotiation can be performed during decision-making, as illustrated in Fig. 4. For each option of a selected set of experiences, performance indicators are calculated and the chosen experience is compared to constraints. If constraints are satisfied, then the option becomes a feasible solution. If the constraints are violated, either another option is considered, or a negotiation with the stakeholders takes place in order to release the constraint.

E. *Storage in the Experience / Knowledge base*

Finally, when the agile process is finished, the complete process is capitalized in the EB along with process' global indicators.

All the experiences corresponding to a same process model are gathered in a set of experiences (SoE) as shown in Fig. 5. Each SoE contains information regarding all the performed experiences, and gets them together in a graphic representation including all the options for a same process.

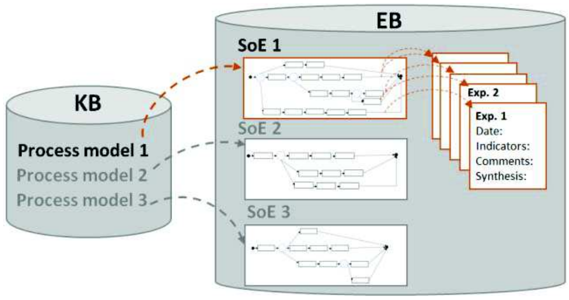

Fig. 5. Illustration of the content of KB and EB.

Moreover, the information concerning each experience is also capitalized (date; cost, delay, performance, and risk indicators; comments; synthesis; etc.). Then, when the architect performs the search step, for a SoE only the experiences satisfying current process constraints are proposed.

IV. APPLICATION EXAMPLE: PROBLEM-SOLVING PROCESS

In order to clarify the previous described agile process lifecycle, a problem-solving process is illustrated in this section.

The problem that needs to be solved refers to a cutting machine that is broken in the workshop, then the product cannot be finished, and consequently it cannot be delivered to the customer.

Following the agile process lifecycle, the first step is to define stakeholders' requirements. For the purpose of this example, stakeholders are: the client, the quality manager, the maintenance manager and the process architect (see section III.B). Requirements are: the global objective is to repair the machine and to prevent that it breaks again (owner: all); the process cannot exceed 4000€ (owner: quality manager); it needs to be completed in less than 5 days (owner: client); and, no more than four people can be involved in the process (owner: maintenance and quality managers).

Taking into consideration the type of process and the requirements as inputs, the process architect searches first into the KB and then in the EB. The standard process model 9S is selected from the KB. It permits to select the SoE of Fig. 6 where all the past experiences of the standard's execution have been gathered (with agile modifications). Experiences 1 and 3 are automatically rejected because they exceed the constraints defined for this process. Two experiences are proposed to the architect even if they are near to exceed limits (5), or they exceed them (4).

The architect chooses experience 4 after negotiation with the quality manager because of the low risk this option involves. After adapting the process to this specific problem, the architect proposes a first version of the process (V0).

During run-time, the process is performed following the proposed first version.

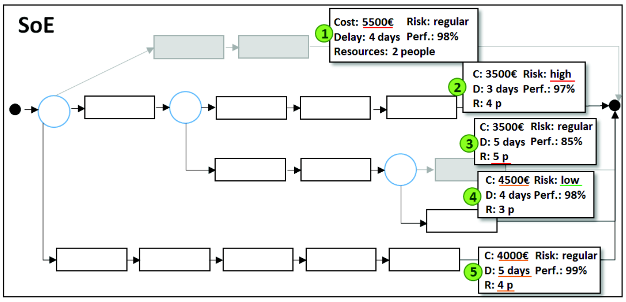

Fig. 6. Retrieved set of experiences for the standard 9S.

Users decide the way forward during each decision point, taking into consideration previous experiences from similar processes. After each decision point, decision results are stored in the EB.

Once the process is completed, and the problem is solved, the process' architect gathers all the information concerning the process and stores it in the EB. The process is given with a global note on performance and risk, as well with comments and synthesis information.

V. CONCLUSION AND PERSPECTIVES

Principles for an agile knowledge-based process have been presented in this article.

Experience feedback and case-based reasoning principles were used as basis to achieve the capitalization and later reuse of experiences and general knowledge in this article.

The importance of experiences and knowledge capitalization and reuse, in order to improve and to facilitate agile business processes, has been outlined. The lifecycle of an agile process has been defined, including its definition, adaptation and storage in the experience base for its future reuse. The main benefit for enterprises, to improve business processes through the balance between fixed-standardized and non-structured processes, has been described.

More works needs to be done on the experiences and knowledge capitalization and reuse mechanisms. First, an easy and accurate system of tagging to ensure capitalization needs to be defined. Second, the retrieval of similar past cases through exploration has to be outlined. Also, a method allowing to calculate indicators and to ensure constraints satisfaction needs to be implemented. Finally, the application of the proposed approach to a real case of an enterprise in order to allow experimentation and feedback needs to be concretized.

REFERENCES

[1] M. von Rosing, J. von Scheel, and A. Q. Gill, "Applying agile principles to BPM" in The Complete Business Process Handbook, M. von Rosing, A. Scheer, and H. von Scheel, Morgan Kaufmann - El Sevier, Waltham, 2014, pp. 553-577.

[2] A. Q. Gill "Agile Enterprise Architecture Modelling: Evaluating the Applicability and Integration of Six Modelling Standards," Information and Software Technology vol. 67, pp. 196–206. Nov. 2015

[3] H. Jabrouni, B. Kamsu-Foguem, L. Geneste, and C. Vaysse. "Continuous Improvement through Knowledge-Guided Analysis in Experience Feedback," Engineering Applications of Artificial Intelligence, Semantic-based Information and Engineering Systems, vol. 24, no. 8, pp. 1419–31, Dec 2011.

[4] V. Llamas, T. Coudert, L. Geneste, J.C. Romero-Bejarano, and A. de Valroger, "Proposition of an agile knowledge-based process model" 8th IFAC conference on Manufacturing Modelling, Management and Control, Troyes, France, 2016, in press.

[5] B. Kamsu Foguem, T. Coudert, C. Béler, and L. Geneste, "Knowledge Formalization in Experience Feedback Processes: An Ontology-Based Approach," Computers in Industry, vol. 59, no.7, pp. 694–710, Sep. 2008.

[6] R. Dove, "Tools for Analyzing and Constructing Agility" Agility forum, Bethlehem, PA, 1994.

[7] M. Christopher, "The agile supply chain: competing in volatile markets" Industrial Marketing Management, vol. 29, no. 1, pp. 37-44, Jan. 2000.

[8] P. Kettunen, "Adopting key lessons from agile manufacturing to agile software product development—a comparative study," Technovation, vol. 29, no. 6–7, pp.408-422, June 2009.

[9] S. Izza, R. Imache, L. Vincent, and Y. Lounis., "An Approach for the Evaluation of the Agility in the Context of Enterprise Interoperability" in Enterprise Interoperability III: New Challenges and Industrial Approaches, K. Mertins, R. Ruggaber, K. Popplewell, and X. F. Xu. Springer, 2008.

[10] L. Lindstrom and R. Jeffries, "Extremme programing and agile software development methodologies," Information Systems Management, 21:3, 41-52, 2004.

[11] M. Fowler and J. Highsmith, "The Agile Manifesto", in Software Development, pp. 28-32, Aug. 2001.

[12] R. McCauley, "Agile Development Methods Poised to Upset Status Quo", SIGCSE Bulletin, vol. 33, 2001.

[13] K. Schwaber and M. Beedle, "Agile software development with Scrum" Prentice Hall, 2002.

[14] H. Sharifi and Z. Zhang, "A methodology for achieving agility in manufacturing organizations: an introduction" International Journal of Production Economics, vol. 62, no. 1–2, pp. 7–22, May 1999.

[15] R. Bergmann and Y. Gil, "Similarity Assessment and Efficient Retrieval of Semantic Workflows," Information Systems vol. 40, pp. 115–27, Mar. 2014.

[16] S. Bekhti, N. Matta, and C. Djaiz, "Knowledge Representation for an Efficient Re-Use of Project Memory," Applied Computing and Informatics, vol. 9, no. 2, pp. 119–35, July 2011.

[17] K. Dalkir, "Knowledge management in theory and practice" Elsevier/Butterworth Heinemann, 2005.

[18] R. Bergmann, "Experience Management: Foundations, Development Methodology, and Internet-Based Applications", Lecture Notes in Artificial Intelligence, vol. 2432, Springer, 2002.

[19] A. Aamodt and E. Plaza, "Case-based reasoning - foundational issues, methodological variations, and system approaches." Ai Communications, vol. 7, no. 1, pp. 39–59, Mar. 1994.

[20] H. Jabrouni, B. Kamsu-Foguem, L. Geneste, and C. Vaysse, "Analysis Reuse Exploiting Taxonomical Information and Belief Assignment in Industrial Problem Solving", Computers in Industry, vol. 64, Oct. 2013.