Deciphering Extreme Programming Practices for Innovation Process Management

C. A. Siebra, Mozart S. A. Filho, Fabio Q. B. Silva and Andre L. M. Santos
CIn/Samsung Laboratory of Research and Development, Federal University of Pernambuco, Recife, Brazil

Abstract - **Innovation can be summarized as the successful exploitation of new ideas. However, there is a long way from the conception of an idea to its final evolution as a product. Several approaches intend to lead the process of innovation into organizations, so that this process can be carried out in a more efficient way. Our research, in particular, aims to investigate how we could apply concepts of *eXtreme Programming* (XP) into the management of innovation process. XP is a software engineering methodology that ensures a more flexible and adaptable software development. We argue that the XP strengths can be also applied to improve innovation processes. Experimental data is also explored as a way to support this hypothesis.[1]**

Keywords - **Extreme Programming, process management, innovation**

I. INTRODUCTION

Innovation processes involve the exploration and exploitation of opportunities for a new or improved product, process or service, based either on an advance in technical practice ("know-how"), or a change in market demand, or a combination of the two [1]. Several works and researches [2,3,4] classify innovation as one of the main sources of economic growth and competitive advantage. In this way, they highlight the importance of the innovation study under the conditions of increased global competition, technological change, fast-changing market situations and continuous customer/client demand for quality services.

In the same way that the development of a product has a cost in terms of time (*DT – Development Time*), the evolution of an idea to a "ready to be developed" stage also has a cost (*ET – Evolution Time*) and this cost is usually uncertain. The concept of uncertainty is so strong in innovation that an idea may not succeed during its evolution when some new knowledge (e.g., technical difficult, existence of similar product in market, long time required for implementation) is raised on. Hence, the degree of innovativeness can be understood as a measure of task uncertainty, as it describes the difference between a status *quo ante* and the actual innovation outcome on several dimensions of a new product [5].

We are considering, in our research, that the final phase of an innovation is a prototyping stage, whose function is to carry out important *proof of concepts* that confirm the implementation viability of an idea. After this confirmation, the product is inserted into the production line and different methods of production optimization [6] can be used to decrease the *DT* of such a product. Differently of the production line, which has several works (in several areas) about the optimization of the production engineering, researches about innovation remain insufficient to provide a simple or elegant theoretical framework that encompasses the strengths of the findings on innovation processes and its optimization [7].

Our research aims to contribute with the current state of the art in innovation process management, focusing on the *ET* reduction and, consequently, decreasing the total time to deliver a new product. For that end, we are using concepts of *eXtreme Programming* (XP) [8], a software engineering methodology to create computational systems in a more flexible and adaptable way. Using such concepts, we are redefining our innovation process management, so that we can improve our innovation indicators. Note that indicators are not only related to the amount of generated ideas, but to several other concepts as detailed in the next sections.

The remainder of this paper is organized as follows: Section II presents some innovation concepts that were important to an initial definition of our process. Section III introduces the eXtreme Programming methodology and its set of practices. Section IV details how the XP practices are being applied to improve our process of innovation. Section V details each of the iterations that compose our process, also on the perspective of the XP practices. Section VI relates the strengths of this process to innovation management indicators. Section VII concludes this work, stressing its main contributions and indicating future research directions.

II. INNOVATION CONCEPTS

Researches about innovation have found useful to divide innovation into three partially overlapping processes:

1) *Production of scientific and technological knowledge*: any process of innovation must be started

[1] The results presented in this paper have been developed as part of a collaborative project between Samsung Institute for Development of Informatics (Samsung/SIDI) and the Centre of Informatics at the Federal University of Pernambuco (CIn/UFPE), financed by Samsung Eletronica da Amazonia Ltda., under the auspices of the Brazilian Federal Law of Informatics no. 8248/91.

with generation of knowledge, or at least being based on some previous knowledge that is part of the organization know-how. Note that, since the industrial revolution, the production of scientific and technological knowledge has been increasingly specialized, by discipline, by function and by institution. This means that the know-how of an organization tends to be very scope limited and this fact leads organizations, which are keen to innovation, to invest in research in new areas;

2) *Transformation of knowledge into working artifacts*: an idea is an embryonic stage of a product. Thus, every idea must evolve, generally via the acquisition of more knowledge, so that it can reach a mature stage and we can decide for its real benefits. This process can also eliminate ideas and, according to our experience, the most common conclusion to several ideas is their elimination rather than their use.

3) *Responding to and creating market demand*: this issue involves a continual process of matching working artifacts with users' requirements. The nature and extent of opportunities to transform technological knowledge into useful artifacts vary amongst fields, over time and we can say that this is a very dynamic issue. One of the most important aims in this context is to attend the time to market constraint. If this constraint is not respected, a product (and consequently work time) may be lost due to first releases of competitors, or because a market window is already overdue.

All these three processes can be associated with the *level of uncertainty* of each new idea. Considering respectively each of the processes above, an idea may not have a correct scientific or technological support, its transformation into a working artifact may be impossible or very hard, and the implementation of the idea may not be well accepted by the marked. All these possible facts are points of uncertainty. According to [9], innovation processes can be defined as a process for reducing uncertainty or, alternatively, as processes for collecting and processing information to cover such uncertainty. The more uncertain the task is, the greater the quantity and quality of information processing required to generate the knowledge necessary to successfully evolve this idea. Thus we can conclude that uncertainty is the difference between the amount of information required to perform particular tasks and the amount of information already possessed by the organization. Based on this discussion, we can summarize that the search for knowledge, to reduce the level of uncertainty, is the core part of any process of innovation, as illustrated in the figure in follow (Fig. 1).

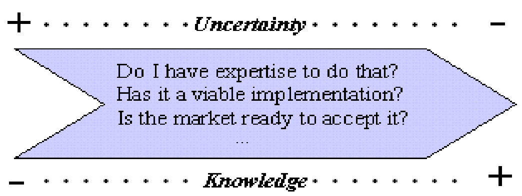

Fig. 1. Relation between uncertainty and knowledge in innovation.

III. EXTREME PROGRAMMING AND ITS PRACTICES

Extreme Programming (XP) is a software engineering methodology and the most prominent of several *Agile methodologies* [10]. Like other Agile methodologies, XP relies on and encourages particular values such as short quick development steps, feedback, communication and adaptation to clarify the requirements and design. As discussed in the previous section, uncertainty is one of the main features of innovation so that it is hard to have a clear idea about the final product. XP is suitable for this situation because such a methodology provides a set of 12 practices that place a higher value on adaptability of an artifact, in evolution, to new scenarios composed by acquired knowledge.

XP was specified to software development and all the main concepts and literature are related to the software engineering context. The intention of this section is to break up this boundary and to present the XP practices in a more general way, so that they can be applied to other engineering contexts, such as to innovation processes. Considering this perspective, the XP practices can se summarized as:

1) *Small releases*: a XP project is composed by a series of iterations that gradually evolves an artifact. Each iteration must spend short intervals, start with an activity that determines the tasks for the current iteration and end with a 'finished' product, which is as functional as possible. Small releases also help the technical members of a team in giving accurate estimates. In software engineering, for example, any time estimate beyond 2 to 4 weeks tends to be imprecise;

2) *Refactoring*: it is the process of improving the design of an artifact without changing its functionality. Refactoring should be done on an ongoing basis throughout development of the artifact. Better arrangements for parts of an artifact can provide, for example, support to other ideas. On the other hand, allowing poorly structured ideas to exist in a project is a risk that accumulates over weeks of development;

3) *Continuous integration*: mixing the latest part of an artifact from each developer together can be a difficult process, especially if this task is not done often. Thus, after creating some part, developers must then integrate their changes with the latest core artifact, as soon as possible, and ensure the validation of this new artifact;

4) *Planning game*: each iteration has requirements. By requirements we understand what is expected from each iteration. Then, this practice accounts for defining the format of these iterations and setting their agenda, so that a minimal plan is available to lead the process;

5) *Test-first design*: XP succeeds by making a project resilient. Resilience means accurate and frequent feedback. This practice suggests that, before the performance of an iteration, we must specify methods to evaluate the results of this iteration. Each iteration must be completely validated before the start of the next one. This avoids the propagation of errors along the project;

6) *On-site customer*: every XP project has one or more individuals to fulfill the customer role on the team. Customers' job is to write and prioritize requirements, assist with acceptance testing and be on hand to answer questions from the development team as they arise. Having the customer as an active member of the team provides frequent and cheap communication and smaller need for formal documentation;

7) *Simple design*: radical requirements changes can appear at any time during a XP project. Thus, artifacts must be able to be updated at any moment and XP teams place enormous value on the extensibility of such artifacts. Note that extensibility seems to be inversely proportional to design complexity;

8) *Pair specification*: all development on an XP team is done in pairs. One thing pairing provides is constant artifacts' review because any decision is taken without dual agreement. This can reduce or eliminate the need for review meetings which can often be boring and wasteful;

9) *Collective ownership*: this practice allows anyone on the team at any time to know what is going on in the organization. Therefore, if some issue is raised up about an artifact and someone in the team has a better expertise in such an issue, this member can immediately contribute with the evolution of an artifact;

10) *Development standards*: having a developing standard for a project is a commonly accepted practice in most projects regardless of methodology. This practice is equally important within an XP team, especially in light of the Collective Ownership practice;

11) *Metaphor*: means an informal architecture of the artifact. The metaphor describes the artifact in simple concepts. The point is to pick something common enough that each member of the team can understand. Since a XP project has few to no formal documentation, the metaphor can be an useful tool to aid in communication among team members. A good metaphor can sometimes inspire improvements to the application itself. But it is important to remember the metaphor is simply a tool of communication and should be changed as the needs of the project change;

12) *Appropriate-work-hours week*: XP promotes a well rested team mainly because tired workers make mistakes rather than creating solutions.

We can trace an informal classification for these practices. The first three practices (1, 2 and 3) are related to the creation of an *iterative* and gradual process of development. The next four practices (4, 5, 6 and 7) allow for cheap, easy and frequent *feedback* to add resilience to the artifact. The faster requirements are discovered and changes are identified, the faster the artifact can proceed in the right direction. The next four practices (8, 9, 10 and 11) are more aimed at support the *communication* and *collaboration* among team members. Finally, Practice 12 is associated with the *human welfare* of being relaxed to better produce.

IV. A XP INNOVATION PROCESS

According to XP (Practice 1), any process of development must be composed of small iterations that evolve a product in functional steps. By functional we mean that the outcome of each iteration must be self-contained and complete in accordance with the level of granularity previously specified into the process (Practice 4). Based on these practices, we have defined a *cycle of innovation* as the set of 5 different iterations (Fig. 2): *investigation*, *identification*, *aggregation*, *consolidation* and *post-validation* (Business Case (BC)+Prototype).

We have defined a total time of 20 work days for each cycle. The first four iterations represent 10 of these work days, whereas the last cycle represents the remainder 10 work days. Thus, we approximately spend one moth in each new innovation if this innovation goes along all the cycle. However, note that an innovation can be eliminated during any of the iterations. This is an acceptable time to create a new and valid product concept, which was already certified about technical restrictions.

The outcome of each iteration must actually provide what is expected from this iteration. Therefore, we have

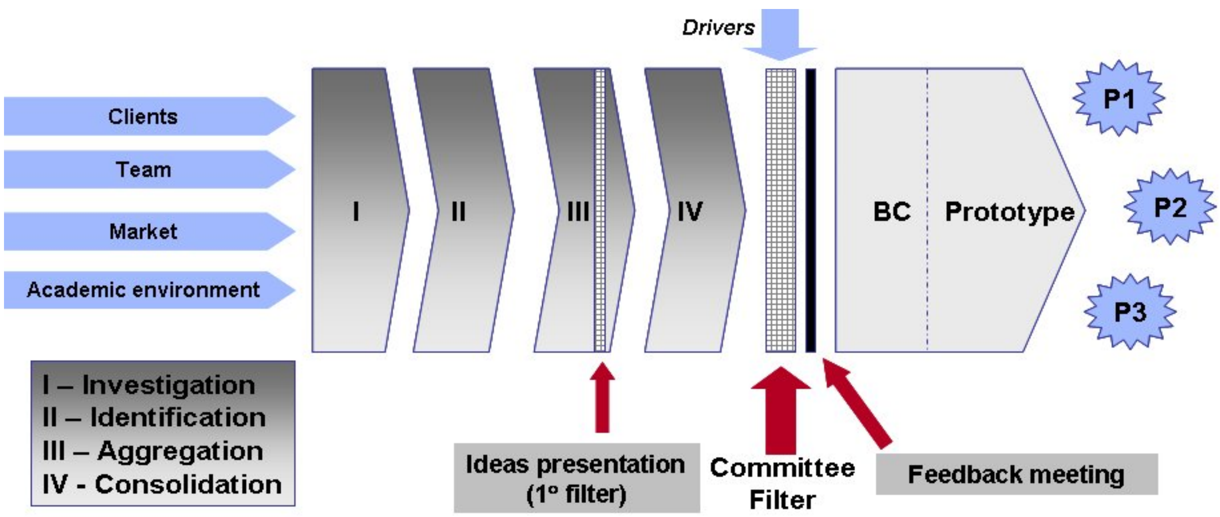

Fig. 2. The innovation process and their stages.

specified the function of each iteration, as detailed in Section V. This specification also works as a kind of evaluation, based on XP Practice 5, which ensures the outcome quality of each iteration. We must also ensure the correctness of the idea itself in each iteration because the idea evolution must be constructed on a correct previous basis.

Considering now the XP Practice 9, we must have a collective ownership of every product in development, so that everyone can collaborate in this process. In this way, when an innovation is started, a simple metaphor (Practice 11) of this innovation is exposed to the team in a blackboard, for example. In fact, innovation is a collaborative skill involving actively scouting the future, generating new ideas, choosing the best, rapidly and effectively implementing them, and then learning the lessons from successes and failures in order to begin again. XP supports this concept of collaboration. However collaboration also means that parts of the evolution of an idea may come from different member. In this case and also following the XP practices (Practice 3) we need always integrate all related ideas as soon as possible.

At last, it is important to understand the use of refactoring (Practice 2) in an innovation process. This practice is very associated with the optimization of code design in software engineering and it means to organize the stuff already implemented in a better way. Imagine that during the evolution of an idea, such an idea is continuously changing its initial main function or concept. When this happens, it is time to take some decision similar to a refactoring. A solution is to start the cycle again, considering the new directions that the idea is going through.

V. ITERATIONS OF THE PROCESS

As discussed in the previous section, we have defined a XP process in 5 iterations: *investigation*, *identification*, *aggregation*, *consolidation* and *post-validation* (Business Case + Prototype). Each of these iterations is detailed in follow.

A. Iteration One – Investigation

According to Linus Pauling, Nobel Price of Chemistry (1954) and Peace (1962), "the best way to have a good idea is to have several ideas and after that, throw away the bad ones". We are using this idea to produce several simple sketches of something that could work and the XP metaphor practice (Practice 11) is very suitable here. We do not want any formal definition about what this idea is. Rather, we are looking for informal concepts that have the potential to evolve in a product.

We have identified four main sources of insights to the iteration of investigation. First, *Clients* can suggest several directions that can be important, for example, for their current main product line. Second, the own *Team* can create threads via individual insights or internal discussions. Third, the trends of the *Market* are important indications about products that must be explored. Fourth, the *Academic environment* is potentially a rich source of inspiration and it is very important to define ways to transform the ideas produced in this environment in real products/services to organizations.

The idea is to allocate pairs of team members (Practice 8) into all these fronts, so that we can stimulate discussions. Sometimes a pair can go along all the cycle together, but this is not compulsory. Depending on the expertise and personal interest, the pair can be changed and ever a different pair, from the original one, can finish the process.

It is important to consider the difference among the creation ability of team members. While some members have more facilities to prospect about ideas, others can find out such process very hard. To balance the ability of the team, there are some techniques of research [11] that can improve the investigation quality. For example: *Naturalistic Observation*, *Correlation Research*, *Participatory*, *Exploratory* and *Experimental techniques*. The experimental technique, for example, is one in which a researcher manipulates a variable (anything that can vary) under highly controlled conditions to see if this produces (causes) any changes in a second variable. Considering the creation of a new product to handsets, if we have an application integrating handsets and GPS, we should think about additional features, such as tracking of people and things and how they could affect a feature such as usability. Then we can prospect if any of these configurations are interesting as a new product. At the end of several iterations we can conclude which features are appropriate or not to be considered.

B. Iteration Two – Identification

After having an idea, it is important to identify if this idea actually has the potential to generate some benefit in terms of product and/or service. For that end, new ideas must be specified in a more formal way. We are carrying out this formalization via the use of a pre-defined template (Fig 3).

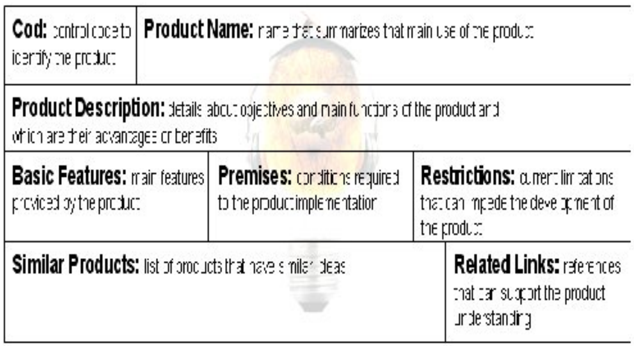

Fig. 3. Draft of our template for identification of new ideas.

This template acts as a standard of development (Practice 10) to ensure the quality of the product generated in this stage because it guides the researcher into the process of thinking about its idea as something to the market. Furthermore, the definition of a template is also important as a way to unify the description of ideas so that they can be easily compared during the next stage.

C. Iteration Three – Aggregation

The process of aggregation intends to formally open the product to other persons from the team so that they can suggest, criticize and indicate research directions. Thus, this iteration is the moment where we more strongly carry out the idea of collective ownership (Practice 9). Furthermore, when someone is working in a research, he/she generally tends to fix his/her line of reasoning in a specific direction so that important points may be forgotten.

This problem is attenuated with the use of pair development, however it could persist. In this way, it is appropriate to have opinions from collaborators that are out of the process because they are able to see issues from different perspectives. Furthermore, as discussed before, other members of the team could have a better expertise about the area in research so that they can play an important role during the evolution of an innovation.

The first official filter of ideas also happens during the presentation of ideas. When an idea is presented, the majority of the participants may find out that such an idea has matters that were not identified by its creator. These matters could be, for example, a technical difficult for implementation, the existence of the same product already in the market, a long time necessary for its implementation and so on. Then, we can see that this collaboration of partners is fundamental to identify problems in a potential product.

Note that managers have an important role in this scenario. Managers who are aggressive about eliciting the ideas of their staff find that getting everyone involved in the effort of improving the operation has an incredible multiplier effect on the rapidity of the improving product and the commitment of employees to those improvements. To do that, managers need to foster a climate of openness that gets employees engaged in the process of aggregation and in the whole innovation process.

The final product of this aggregation process is the consolidation and implementation of all suggestions given by the team during the presentation meetings. Note that several issues can be still open after the meetings and one of the aims of proponents is to sort out these issues.

D. Iteration Four – Consolidation

The consolidation stage is the last stage before the committee filter. Thus, this stage must provide all necessary information that can support the understanding of the product. From our experience, we have found that a deeper technical description could be very useful to better characterize a product. Thus, this stage must focus on:

- Product abstract architecture: specification of the components that compose the product and the interaction between them. Furthermore, the alternative techniques available to implement these components;
- Product dynamic: specification of the main fluxes of execution. In other words, how the product is going to be used.

Both descriptions must consider an appropriate level of details. The intention of this stage is not to give so many details (Practice 7) so that the generated documentation can be used in future cycles of development. Rather, specifications must only provide a level of information that supports the understanding of products benefits and viability to a no-technical committee.

E. Iteration Five – Post-validation

This stage starts with the *committee filter*, which decides for the most appropriate ideas according to a set of drivers that are parameters to qualify/classify an idea (e.g., amount of traffic that it intends to generate, business/personal utility, etc.). Drivers are generally defined by top members of the organization hierarchy and they are the main parameters to ensure the product quality in this level. Observe that the idea of a committee, which in fact represents the client, acting only in this moment, is contradictory to XP (Practice 4). However the presence of clients could be a natural factor to restrict the creative process of the team.

Another important component of this process is the *feedback meeting*. This meeting accounts for presenting the reasons of acceptance or not to all team members. This kind of feedback is important because team members may become unmotivated when their ideas are rejected. Thus, an explanation about motives of acceptance and rejection are important to maintain the team closed to the whole innovation methodology. At the end, a Business Case (BC) and prototype of the chosen products are generated, closing the innovation cycle. Each new cycle must go through all these processes and two or more cycles can be carried out in parallel.

VI. EFFECTS ON THE INNOVATION INDICATORS

Within the literature about innovation management, measures of aspects of innovation are frequently proposed, responding to the needs of both firms and academics to understand the effectiveness of innovation methodologies [12]. In fact, if we intend to propose improvements to some process, it is important to understand how such improvements affect the innovation indicators. We consider the following indicators as important for a process of innovation:

- *Amount of initial ideas*: total number of ideas generated in the investigation iteration;

- *Amount of final ideas*: total number of ideas that reaches the post-validation iteration;
- *Average life time of non-selected ideas*: it is important to identify as soon as possible ideas that are not going to be useful because their evolution usually consumes work time of the team. Thus, a good process tends to decrease this indicator;
- *Quality of final ideas*: we are relating quality with number of patents generated by our team. If a product has a concept that is worth to be protected, we can say that such a product has quality in terms of innovation.

Using such indicators, we have monitored four cycles of innovation. The first cycle (Fig. 4), which has not used the XP concepts, we had the generation of 32 ideas (*amount of initial ideas*), whereas 3 of these ideas have reached the last stage (*amount of final ideas*). The *average life time of non-selected ideas* was 6 days and we do not have any patent as results of this cycle. In the other three cycles, which has used the XP concepts, we had an average of about 11 initial ideas, 3 final ideas and an average life time of 4 days. Furthermore, 2 patents were generated as part of the innovation process.

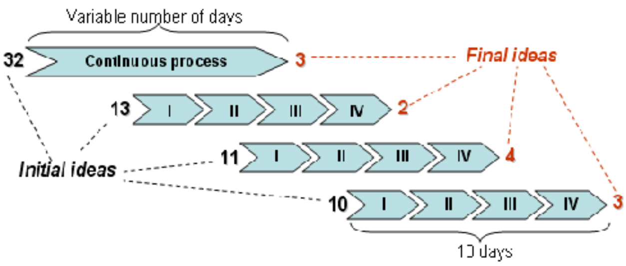

Fig. 4. Four cycles of innovation.

Note that the initial number of ideas has decreased using XP practices. However the final number is still the same. Thus we have a better gain and less effort because the team is not working on ideas that are not going to be deployed. A shorter average life time (4 against 6 days) also corroborate with this idea. Furthermore, we have started to generate patents, what could indicate innovation with more quality.

VII. CONCLUSION AND DIRECTIONS

This paper discusses how XP practices can be used to improve the innovation process management via the continuous use of concepts such as communication, cooperation and early feedbacks. We have employed the XP practices and some cycles of innovation and we could observe that the process becomes simpler and more efficient according to some indicators for innovation.

Innovation is inherently uncertain, given the impossibility of predicting accurately the cost and performance of a new artifact, and the reaction of users to it. It therefore inevitably involves processes of learning through either experimentation (trial and error) or improved understanding (theory). Note that this is also valid to innovation process management, so that, only via experiments and understanding of related facts, it is possible to better adjust the process according to the features of a particular organization.

ACKNOWLEDGMENT

The authors would like to thank all the software engineers from CIn/Samsung Groupware, which have supported the implementation of an innovation methodology in our team. Authors are also very grateful for the support received from Samsung/SIDI team, in particular from Helder Pinho, Miguel Lizarraga, Ildeu Fantini and Vera Bier. The National Council for Scientific and Technological Development (CNPq) has provided valuable support to the project through the Brazilian Federal Law no. 8010/90.

REFERENCES

[1] D. Mowery, and N. Rosenberg, “The Influence of Market Demand upon Innovation: a Critical Review of Some Recent Empirical Studies”, *Research Policy*, 1979, vol. 8, pp. 102-153.

[2] G. Ekvall, G. and J. Arvonen, “Leadership Profiles, Situation and Effectiveness”, *Creativity and Innovation Management*, 1994, vol. 3, pp. 139–161.

[3] J. Howell and C. Higgins, “Champions of Technological Innovation”, *Administrative Science Quarterly*, 1990, vol. 35, pp. 317–341.

[4] M. Tushman and C. O’Reilly, *Winning Through Innovation*, 2002, HBS Press, Boston.

[5] J. Galbraith, *Organization Design*, Reading, MA: Addison-Wesley, 1977.

[6] C. Garton and Erika McCulloch, *Fundamentals of Technology Project Management*, Mc Press, 2006.

[7] D. Mowery, and B. Sampat, “Universities and Innovation”, *The Oxford Handbook on Innovation*, Ed. J. Fagerberg, D. Mowery, R. Nelson, Oxford University Press, 2005.

[8] K. Beck, K. *Extreme Programming Explained*, Addison-Wesley, Reading, Mass, 2000.

[9] F. Damanpour and M. Schneider, “Phases of the Adoption of Innovation in Organizations: Effects of Environment, Organization and Top Managers”. *British Journal of Management*, 2006, vol. 17, pp. 215-236.

[10] P. Abrahamsson, J. Warsta, T. Siponen, and J. Ronkainen, “New Directions on Agile Methods: A Comparative Analysis”. In *Proc. International Conference on Software Engineering*, 2003, Portland, Oregon, USA, pp.244-254.

[11] S. Cheng, “A New Invention Theory: The Four Invention Methodologies and Logical Analysis as an Invention /Innovation Tool”, 1998, Available in [http://sc-innovation .com]

[12] R. Adams, J. Bessant and R. Phelps, “Innovation management measurement: A review”, *Int. Journal of Management Reviews*, 2006, Vol. 8(1), pp. 21–47.