

PROJECT MANAGEMENT IMPROVEMENT IN EXTREME PROGRAMMING

Houda Zouari Ounaies¹, Yassine Jamoussi², Mohamed Ben Ahmed³

ENSI, National School of Computer Science, 2010, Manouba, Tunisia

E-Mail : Houda.Zouari@isetr.rnu.tn¹ Yassine.Jamoussi@ensi.rnu.tn²
Mohamed.BenAhmed@riadi.rnu.tn³

Abstract: In recent years, a great interest has been advocated to agile methods, particularly, eXtreme Programming or XP. This method creates controversial critics: Some experts consider that XP includes essential engineering practices for every software project. Others consider that these practices are impractical and go against project development productivity.

We consider that XP is based on practices of good sense and can give many benefits to organization adopting it. However, we consider that the principle lack in XP is project management and especially the measurement process. This issue is non-sufficiently defined in XP. We propose in our approach to improve XP method taking into consideration Capability and Maturity Model Integrated (CMMI) objectives. We focus on measurement process and we provide guidance for its formalization. Improved XP process is critiqued from Capability and Maturity Model for software (SW-CMM) perspective. This model considers measurement as a key concept. We think that our proposed improvements allow organizations adopting XP to reach SW-CMM level 3 and to address some of level 4 practices.

Key Words: eXtreme Programming, process improvement, project management, measurement, SW-CMM, CMMI, PSM, experience capitalization, Case-based reasoning.

1. INTRODUCTION

Extreme Programming (XP), an agile method officially burned in 1999, creates controversial critics in software engineering communities: Some experts consider that most of XP consists of good practices that should be thoughtfully considered for any environment [1]. Others experts consider Extreme Programming to be harmful for reliable software development [2].

We share the opinion of experts who think that XP can guaranty the success of small or medium development projects if certain issues of XP are improved. In this paper, we propose XP improvement guidance. We think that this improvement can lead organization adopting XP to reach high level of the capability Maturity Model (CMM) for Software.

2. EXTREME PROGRAMMING

XP method dimensions one development project by four variables: cost, time, quality and scope [3]. One XP team control the three first variables by regulating its work rhythm basing on the scope variable and by adopting iterative cycle, basic practices and basic values.

2.1 XP Practices

Basic XP practices are:

1. Programming practices: Simple design, refactoring, automated test (acceptance and units),
2. Collaboration practices: Pair programming, Collective ownership of the code, Coding standards, Metaphor, Continuous integration (many times a day),
3. Management project practices: Small releases on a very short (two week) cycle,

iterative planning, On-site customer, 40-hour week (never work overtime two weeks in a row).

2.2 XP Values

Four values are essential in an XP project:

1. Simplicity -- developers should adopt the simplest solution work,
2. Communication -- between team members,
3. Feedback -- essentially by automated tests,
4. Courage -- to admit problem and to face them.

2.3 XP CYCLE

XP project cycle consists of two embedded cycles: releases and iterations (figure 1).

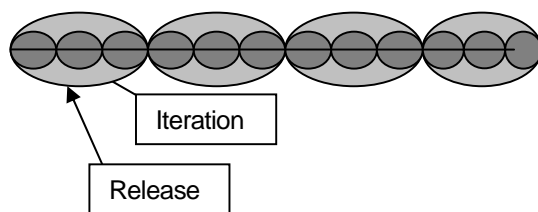


Figure 1. XP Project cycle

Release cycles concern functionalities visible by customer. At the cycle beginning, releases are planned. Control is carried out by measures such as failed acceptance tests rate and defects number. At the cycle end, release is made even if not all scenarios planned have been developed. Implementation is done in a success of iterations.

Iteration cycle concerns tasks realized by developers. At the start of iteration, these tasks are defined with customer presence. Monitoring is carried out by indicators presented by measures such as failed units tests. At the end of iteration, a table of principle indicators is displayed to all the development team (even the customer) in order to evaluate and to be aware of the real situation of the project. The customer is on

site and is integrated to the team so he can estimate regularly the evolution of its product.

3. XP PROCESS CRITICS

We think that XP presents practices of good sense, which place developers and customer in the centre of development process. Nevertheless, we have noticed that problem resolution in one XP project is not explicit, even if XP practices aims essentially to reduce risk in software development project. Although, XP does not emphasizes process definition or measurement to the degree that models such as the CMM (capability Maturity Model) do [1]. Thus, XP's "activities" are not formally identified or described [4]. Indeed, relations between problems, reflected by measures, and solutions, concerted by XP practices, are informal.

Measurement is actually considered as a key practice to control, improve and manage the development process and software quality. In this work we propose to improve this facet of XP project management process by formalizing its measurement process.

We think that XP measurement process formalization is not in disagreement with agility and simplicity. Indeed, it is inherently difficult to manage what cannot be measured objectively [5].

4. XP PROJECT MANAGEMENT IMPROVEMENT APPROACH

We propose to use Practical Software and Systems Measurement (PSM) [6] guidance to improve XP measurement process. This model is an implementation of ISO/IEC 15939 [7] Software Measurement Process standard. PSM process model is presented in Figure 2.

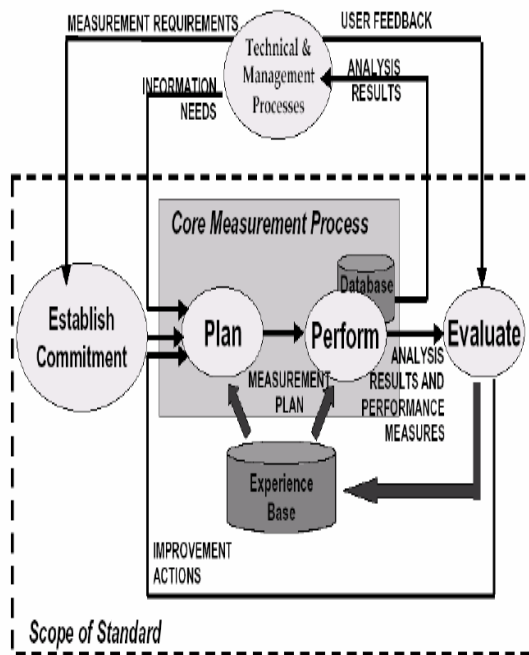


Figure 2. PSM process model

In the first step of our approach we bring up a mapping between actual XP measurement process and PSM process model (table 1). This comparison leads us to consider two principles lacks in the actual XP measurement process:

1. XP measurement planning should be based formally on information needs of decisions makers and team. PSM proposes a measurement information model that relates the need in information with what can be measured really.
2. XP measurement evaluation and improvement actions should be based on measured feedback achieved by using XP project measured experience and XP organizational measured experience. This means that we should use the experience acquired during the actual XP project, in one side, and in the other organization XP projects in the other side.

Our solution deals with the second issue concerning adopting XP measured feedback.

PSM process model phases	XP measurement process conformity
Commitment establishment	Largely adopted: XP method adopts measures.
Measurement planning	Partially adopted: In XP, this phase is <i>not formally</i> based on decisions makers and teams information needs.
Measurement perform	Partially adopted: Project measures are <i>not formally</i> structured in a database.
Measurement evaluation & improvement actions	Partially adopted: This activity is based on user feedback. In XP, feedback is possible thanks to practices such as frequent releases and automated acceptance and units tests. But it is <i>not formally</i> based in project measured experience (saved in the project database) and organizational measured experiences (saved in the Experience Base).

Table 1. Mapping between XP measurement process and PSM process model

The second step of our approach consists in bringing a solution to none conformity that we find in XP measurement process with PSM process model and especially XP measurement evaluation and improvement actions phase.

Our comparison leads us to formalize the feedback acquired during experiences in both actual XP project and others organization XP projects in order to assist project manager (all the team can participate in this task) to take decisions and adopt suitable

improvement actions. This practice is called experience capitalization.

Data mining approach are used the capitalization process. Various tools are associated to this approach, such as Neural Network, Decision Tree, Case-Based-Reasoning (CBR). The choice of one specific technique leads us to do a compromise between results prediction and results readability. Communication and interaction in the team are essential in XP and in agile environment generally. Thus, we propose to use CBR approach which is characterized by the extreme readability of their results.

4.1 CBR XP MEASUREMENT PROCESS FORMALIZATION

Capitalization of passed projects experiences is an important factor in the management of current projects. Indeed, feedback acquired by teams during development process is constantly improved and better interpreted. This will leads teams to improve their capacity to evaluate the project situation and to take effective decisions. In XP project management context, we propose to use CBR systems: Problems are resolved by comparing the situation of the actual project to others cases stored in the “Case Base”. In the PSM model, “Case Base” is referenced as “Experience Base”. If one passed experience is sufficiently similar to the actual case, solution adopted in the passed experience can be applied to the current situation [8].

XP software project case is characterized by its situation and the experimented solution. We propose to structure these to notions by hierarchic concepts. The XP project situation is described by its basic characteristics (type and size), its advancement (week, month or year) and by different measures performed during the development process (figure 3).

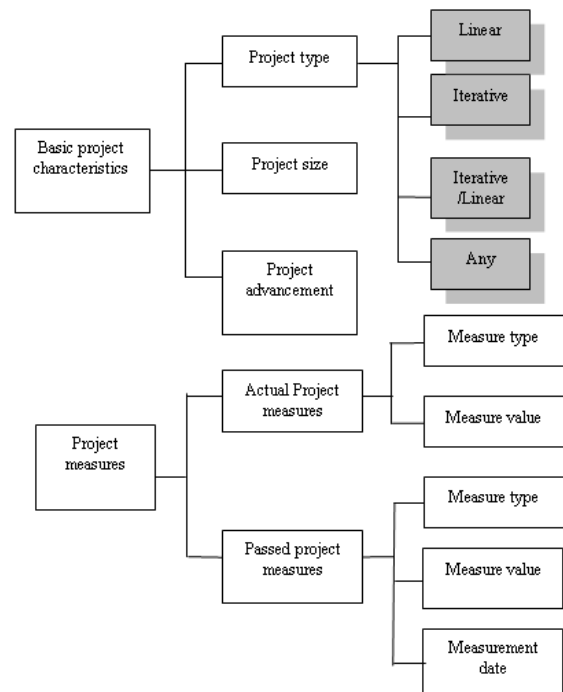


Figure 3. CBR XP structured situation

XP experimented solution stored in the “Case Base” is structured essentially by improvement actions used during XP process and by the evaluation of results obtained after using these actions (figure 4).

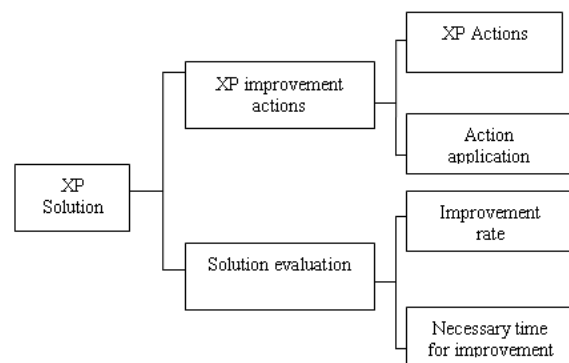


Figure 4. CBR XP structured solution

We have implemented this solution and we simulated some projects experiences. However, extracting knowledge necessitate an important Case Base, that will be available gradually by using this solution in the organization. We propose to show the availability of our approach by evaluating improve XP process from SW-CMM

(Software Capability Model) [9] perspective, a Software Engineering Institute (SEI) evaluation and improvement model.

4.2 EVALUATION OF XP IMPROVED PROCESS FROM CMM PERSPECTIVE

Our evaluation of improved XP process is based in a previous evaluation of XP from SW-CMM elaborated by Mark Paulk [1].

CMMI (Capability and Maturity Integrated Model) [10], is the last model proposed by SEI. It integrates SW-CMM and almost all previous CMM models.

CMMI model dedicates a key process area (KPA) for measurement: Measurement & Analysis. This KPA is based on ISO/IEC 15939 Software Measurement Process. The PSM model is an implementation of this standard. Therefore implementing PSM model leads to be conforming to CMMI Measurement and Analysis KPA.

All CMM models, and in our context SW-CMM, give much importance to measurement in the process: Measurement concepts are spread all over SW-CMM Key process areas. Therefore, we think that the formalization of XP measurement process from a CMMI perspective will improve results obtained in earlier XP evaluation from SW-CMM perspective.

We evaluate XP process basing on SW-CMM model and not on CMMI model because we think that adopting CMMI in the whole XP process will distort this method from its initial objective. But we use CMMI only in the measurement process. Improvements obtained affect almost all SW-CMM model KPA. We resume this evaluation in Table 2.

We can assume that adopting eXtreme Programming with a formalized XP measurement process allows an organization

adopting this method to improve its development process and to reach the level three of SW-CMM model and to address some issues of the level four.

5. CONCLUSION

Adopting XP, a controversial agile method, can bring much good practices of good sense. However, we consider that project management is none sufficiently defined in this method and particularly the measurement process.

Measures are critics to control, monitor and improve software development process. In this work we have proposed to improve XP management project by proposing guidance to XP measurement process formalization. This work supports PSM (Practical Software and System Measurement) guidance. This model is based on ISO/IEC 15939 Software Measurement Process norm. It is also considered as an implementation of CMMI Measurement & Analysis Key Process Area.

The mapping between PSM process model and XP measurement process leads us to consider XP measured experience, at project and organizational levels in order to learn from the past and to take appropriate decision in the development project. Therefore in our approach we propose to formalize XP measured experience capitalization by using Case Based Reasoning (CBR) approach. Finally we have evaluated the XP improved process from a SW-CMM perspective in order to prove the efficiency of our approach. We notice that such improvements would lead an organization to reach the level three of SW-CMM and to touch the level four.

SW-CMM maturity level	Actual XP process satisfaction	Improved XP process satisfaction	Improvement proposed solutions
Maturity level 2 : Repeatable			
Requirement management	++	++	-
Software project management	++	++	-
Software project tracking and oversight	++	++	-
Software subcontract management	--	--	XP is dedicated to small and medium project
Software quality assurance	+	++	Adopting a formalized XP measurement process conformed to ISO/IEC 15939 norm.
Software configuration management	+	++	Satisfied by XP project experience capitalization.
Maturity level 3: Defined			
Organization process focus	+	++	Formalizing XP measurement process: the strengths and weakness of the software process used are identified relative to a standard process.
Organization process definition	+	++	Guidance in project management process,
Training program	--	+	Satisfied by experience capitalization in the organization.
Integrated software management	--	--	-
Software product engineering	++	++	-
Intergroup coordination	++	++	-
Peer reviews	++	++	-
Maturity level 4: Manager			
Quantitative process management	--	+	Satisfied by project management using a formalized measurement process.
Software quality management	--	--	-
Maturity level 5: Optimizad			
Defect prevention	+	+	-
Technology change management	--	--	-
Process change management	--	--	-

++ Large satisfaction
+ Medium satisfaction
-- Low satisfaction

Table 2. Evaluation of XP improved process from SW-CMM perspective

The use of data-mining techniques can be interesting in project management context. In our solution we propose to use CBR systems: A simple technique in an agile environment. Nevertheless data-mining combination techniques can eventually bring to experience capitalization process a better compromise between predictability and legibility in proposed results. We think that more researches are needed to prove the effectiveness of such techniques in project management in order to assist organizations to improve their capacity to produce software.

6 REFERENCES

- [1] Paulk, M. C., (2001). "Extreme Programming from a CMM Perspective." IEEE Software. November/December, pp. 19-26.
- [2] Extreme Programming Considered Harmful for Reliable Software Development, Gerold Keefer, AVOCA GmbH, 2002. On-line at: http://www.agilealliance.com/articles/articles/XP_ConsideredHarmful-GeraldKeefer.pdf
- [3] Kent Beck. Extreme Programming Explained : Embrace change. Addison-Wesley, 1999, ISBN 201-61641-6.
- [4] John Smith, "A Comparison of RUP and XP." Rational Software White Paper, May 2001.
- [5] Walker Royce, Software Project Management: A Unified Framework. Addison-Wesley, 1998.
- [6] Dept. of Defence and US Army, "PSM: Practical Software and Systems Measurement – A Foundation for Objective Project Management," Version 4.0c, March 2003.
- [7] ISO, ISO/IEC, "IS 15939: Software Engineering – Software Measurement Process Framework," International Organization for Standardization, Geneva, 2003.
- [8] Kolodner, J. 1993. Case-Based Reasoning. San Mateo, CA: Morgan Kaufmann.
- [9] Paulk, M.C., B. Curtis, M.B. Chrissis, and C.V.Weber. The Capability Maturity Model: Guidelines for Improving the Software Process. Addison-Wesley, Reading, MA, 1995.
- [10] CMMI Product Development Team, CMMISM for Systems Engineering/Software Engineering, Version 1.02 (CMMI-SE/SW, V1.02) Staged Representation, Carnegie Mellon University, CMU/SEI-2000-TR-028, ESC-TR-2000-093, November 2000.