Journal of Advanced Computer Science & Technology, 4 (2) (2015) 225-230
www.sciencepubco.com/index.php/JACST
©Science Publishing Corporation
doi: 10.14419/jacst.v4i2.4173
Research Paper

Assessing quality in software development: An agile methodology approach

V. Rodríguez-Hernández [1]*, M.C. Espino-Gudiño [2], J.L. González-Pérez [2], J. Gudiño-Bazaldúa [3], V.M. Castaño [4]

[1] Facultad de Informática, Universidad Autónoma de Querétaro, Querétaro, México
[2] Facultad de Ingeniería, Universidad Autónoma de Querétaro, Querétaro, México
[3] Facultad de Lenguas y Letras, Universidad Autónoma de Querétaro, Querétaro, México
[4] Centro de Física Aplicada y Tecnología Avanzada, Universidad Nacional Autónoma de
**Corresponding author E-mail: vicente.rodriguez@uaq.mx*

Copyright © 2015 V. Rodríguez-Hernández et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

A novel methodology, result of 10 years of in-field testing, which makes possible the convergence of different types of models and quality standards for Engineering and Computer Science Faculties, is presented. Since most software-developing companies are small and medium sized, the projects developed must focuson SCRUM and Extreme Programming (XP), opposed to a RUP, which is quite heavy, as well as on Personal Software Process (PSP) and Team Software Process (TSP), which provide students with competences and a structured framework. ISO 90003:2004 norm is employed to define the processes by means of a quality system without new requirements or changing the existing ones. Also, the model is based on ISO/IEC 25000 (ISO (IEC 9126 – ISO/IEC 14598)) to allow comparing software built by different metrics.

Keywords: *Software Develoment; Quality Assesment; Agile Technology; Education; Software Training.*

1. Introduction

The development of software systems for research projects within the computer Science and Engineering Schools at the University of Querétaro have had a great impulse in recent years; however, they lack an appropriate methodology to determine, objectively, the quality of the software, as well as the quality of the finished product itself, thus lacking a reference model to organize different teams [2]. It is important to point out that most mexican companies developing software of different kinds do not have defined processes. In these sense methodologies, assessing software quality are due, since students themselves have no way of quantifying the product to be developed, the delivery time, the estimated cost, the technology and tools to use, the customer requirements, etc. In this context, quality of software is understood as the degree in which customers are satisfied with the development, whether it meets their current processes or rules. Also, the resulting software product must comply with international norms, which guarantee that the students know how to produce software without errors and in an optimal way. Likewise, they should have a previous e-training so that they can form a high-performance team, that is, independent teams, which are a self-directed and self-planned, basic philosophy of TSP and SCRUM. On the other hand, the quality methodology must be able to identify, in an early stage, the criteria to be met based on the customer requirements: thus, the standard ISO/TEC 25000 [3] is a good start point.

2. Description of the methodology

2.1. Agile methodologies

According to Priolo [20] they are a set of methodologies, which use similar practices, based on results, people and interactions. Due to the fact that there were different practices with analogous features, their creators decided to

organize themselves and to organize their criteria together. Many companies and organizations decided to implement clever methodologies that contain the necessary features to be implemented and since a modification in the process is not that expensive. Likewise, many professionals of the area seem to see software architecture in the context of the paradigm of development aimed at cleverness [18].
Agile processes have gained popularity among many software companies as a way to reduce costs and increase their ability to handle a dynamic change in the recent conditions of the market [16]. Such processes constitute a true frame of conceptual work of the engineering of software, which promotes iterations along the cycle of life of the project. There are many methods of agile development; the majority minimizes risks by developing software in the short term [19]. The software developed in a unit of time is called an iteration, which should last from one to four weeks. Each iteration of the cycle of life includes: planning, requirement analysis, design, coding, revision and documentation [1], [22]. Iteration should not add too much functionality to justify the launch of the product to the market, but the goal is having a demo (without errors) at the end of each iteration. At the end of each iteration, each team gets to evaluate the priorities of the project. Thus, these types of methodologies which have been used within the Engineering Faculty are Extreme Programming (XP), and SCRUM.

2.2. Extreme programming (XP)

XP works basically this way: work is done under the assumption that there is not enough time to fulfill the requirements and exhaustive documentation; there is a team of no more than twelve people, which work in pairs with matters referring to programming. Besides within each team, there is a customer as the center of all expectations. This last one is an important part due to the fact that during the development process, he/she may intervene to make possible modifications. Such modifications are made in very short periods of time (less than two days), what it is about is talking to the client about what he/she wants or needs and immediately starts programming, showing him/her what was done, if he/she is happy, then go on with the next activity, if not make the corrections right away.
XP is different from traditional methodologies, mainly in that it places more emphasis on adaptability than in previsibility. Defenders of XP consider that the changes of requirements “on the run” are a natural and inevitable aspect and even desirable in project development. They believe that being able to adapt to the changes of requirements at any point in the life of a project it’s a better way of approximating and more realistic than trying to define all the requirements at the beginning of the project and invest effort later in trying to control the changes in the requirements.

2.3. Scrum

This type of methodology is quite similar to Extreme Programming given that it counts on a series of ideas in which they coincide such as formation of small teams and client becomes part of the team too, besides the multiple iterations (sprints) with the process development of the software, as is defined “it is a clever process for the development of software. This method focuses on the management of software projects, dividing its development in iterations called sprints. Each sprint lasts between two to six weeks. Similarly, the project is monitored or controlled by means of daily meetings. Scrum is ideal for the rapid evolution of projects or very urgent needs [17].
Scrum applies ideas from industrial process control theory with the purpose of getting adaptability and productivity. Scrum does not define activities nor specific techniques for the construction of software. However, it does focus on the rules of work of teamwork to be able to produce flexible systems in an environment of change. Given that scrum makes little emphasis on engineering activities, it is common to complement it with other strong methods, in terms of practice of engineering such as XP.

2.4. System for quality management

International norm states that to get the requirements for a system of quality management an organization is required, and it has as its goal to improve the client’s satisfaction by means of the efficient application of the system, including processes for continuous improvement of the system and the assurance of conformity with the client and the appropriate applicable requirements [4]. Therefore, if this model were adopted, in a general way, it implies in the first place, identifying the system requirements for the quality management; there has been some work done in the degrees that are offered at the Schools of Engineering and Computer Science, of the University of Queretaro, the processes to be able to carry them out and then register the new systems. As part of this new system, the new documentation must be able of being recorded that shows that within the project that is being used new processes are being carried out. The development of new processes is the part that takes more time in the preparation for a new certification in ISO 9001 (ofr our case 90003). Because it implies that changes are made in the traditional way that those processes were being carried out, so that this way they are able to comply with the requirement of the norm. Degrees which are offered at Universities are different since the majority of them, have some processes defined, however, almost all of them need to make changes to begin complying with the norm.
All the requirements of this international norm are generic, and it is pretended that they all are applicable to all the academic teams that are being made within the Faculties, regardless of their type, size or supplied product. It is

particularly advantageous using it for research educational projects, innovation and development within the Faculties of engineering and computer science. The educational projects that are made, are with the intention of being self-planned and self-managed, to be able to establish, document, implement and maintain a system of quality management and to improve continuously its effectiveness according to the requirements of international norms. Teams must: a) identify the necessary processes for the system of quality management and its application in the entire organization, b) to determine the sequence and interaction of these processes, c) to determine the necessary criteria and methods to ensure the performance and control of these processes are effective, d) ensure the necessary availability of resources and information to support the operation and follow-up of these processes and e) superise, measure and analyze these processes and f)to put into practice the necessary actions to reach the planned results and the continuous improvement of these processes. These processes will be handled by the professors in charge of the projects and must be in conformity with the requirements of this international norm. When a team of high-performance decides to externalize a process that affects the conformity of the product with the requirements, the team must guarantee control of such processes [4].
Documentation of the management system must include: a) documented statements of a policy of quality and quality objectives, b) a quality handbook, c) recorded procedures required for this international norm, d) documents needed for the organization to ensure the efficient planning, operation and control of its processes, and of e) the required records in this international norm [4].

2.5. Personal software process (PSP)

PSP is a designed process to help software engineer's control, handle and improve their work [21]. In other words, PSP was designed to help software professionals to constantly use healthy software engineering practices teaching them to plan, and make a follow-up of a job, using a process well-defined and measured, to establish measurable goals and finally to track constantly to get the defined goals. Similarly, it shows how to apply advanced methods of engineering to their daily tasks. It gives detailed methods of planning and measurement; it shows to engineers how to control their outcome with respect to these plans and explains how the defined processes lead their job [21].
What PSP is about is about is assessing the personal process of software development, detecting errors, determining measures, making serious and constructive judgments at the moment of work, and in simple words, assessing details at the moment of programming such as estimating coding times of a program or perhaps getting a point of reference. In a similar way, it contains forms, guides and procedures for software development. When used properly it allows getting historical information, which is required to establish and comply with the requirements; additionally, it allows that routine tasks are more predictable and efficient. Application of PSP in several development processes and statistical analysis information is generated in each one of these, will allow the software engineer identify strengths and weaknesses, and growing through the process of self-learning and self-improvement. Quality in PSP is an aspect strongly related to the quantity of defects that software products have. At this level, some applicable methods are introduced to the software-development process, within a projects approach in big scale, but without dealing with problems of communication and coordination of working teams. PSP provides specific guidelines about how software developers can continuously improve their performance, based on standard practice to develop software.

2.6. Team software process (TSP)

TSP provides a framework that is built over PSP basis with well-defined foundations development in which software products are generated in several cycles and thus standard measures are established for the quality of products and team performance and the developers.
TSP uses the same basic measures as PSP –time, size and defects- and adds dates of termination to tasks. For all measurements, planned and real data are picked up at an individual level. The TSP measurement frame consolidates individual data in a team perspective. Data are analized weekly by the team to understand the state of the project against the quality of the objectives. The measurement TSP frame also makes available other points of view of the data, such as product, phase, task, week, day, etc. Personnel and equipment data are filed to provide a deposit of historical data for future use.
TSP provides specific guidelines about how developers (our students) trained in PSP can work effectively as members of a team of high-performance.

2.7. Product quality

The model of quality is useful to evaluate the product quality of software as well as the one of complex product – software plus hardware in which it is installed- Specifically it should serve as framework, at the time of defining objectives we want to reach, in final products as well as in intermediate ones. For this reason, we must establish a relationship between the quality of product and the set and subset of features that this norm offers, in such a way that

the list will be useful as a check-list at the time of assessing the results. It would be important to highlight that defining the quality model helps to systematize work at the time of defining features that our system complies with.
Likewise, it allows avoiding errors or omissions in a later assessment process.

2.8. ISO/IEC 25000

For our student formation through innovation and development engineering projects the use of Square project approach is highlighted (software product quality requirements and evaluation) which main goal is creating a series of international standards, organized in a logical way, that allows covering two processes: the specification of quality requirements and assessment of software quality. In general standard ISO/IEC 2500 is made of standard ISO/IEC 9126 plus ISO/IEC 14598 grants the high-performance teams getting to have the specification, planning, management and measure and assessment of software quality [5].
The family of norms ISO/IEC 9126 (quality in software product) has been made simultaneously to family ISO/IEC 14598 (assessment of software product). Specifically, part I of ISO/IEC 9126 explains the model of quality [6]; part II will provide external metrics to measure attributes of six features defines in part I [7]; part III defines internal metrics while [8] and part IV refers to quality in use [9].
Standard ISO/IEC 14598 provides methods to measure and assess product software quality that may be used by the clients who are going to acquire, develop or assess a product. These results may be used to measure the level of conformity with the requirements and to carry out improvements when necessary [10], [11], [12], [13], [14], [15].

3. Methodological proposal

According to this set of norms, methods, models and metrics posed in this article, it is proposed a combination of them to define a quality system at a lower-cost guaranteeing that the future graduates of our schools have the necessary capabilities for a successful performance in their professional life, which are detailed below under the premise of “doing what needs to be done in the right way."

1) Having training in clever methodologies for software development such as XP and SCRUM, in such a way that they make small teams of no more than 14 students, who are willing to work in subteams of two people, to begin with the activities of development of applications; similarly, they must accept the commitment of doing like a habit the use of PSP in each of the phases of software development, which for practical purposes, becomes easier with the use of tools dedicated to such assessment (the portable application PSP Dashboard is recommended).besides Besides the above mentioned, the role of a leader of the project must be assigned.
2) Each teamwork is in charge of a role within it; that is, the team counts on three basic roles: a) Master role.- is the person in charge of checking how the methodology is working, processes of the entire team and how the interaction with the client is taking place, b) programming role.- it is the person in charge of taking to the codable (recordable)environment all the activities defined in each of the iterations of our program with the client, c)Client role: is the person in charge of knowing the needs in the development of product software.
3) Once the above mentioned ideas have been considered it has to be defined how the process of activities develops. In the beginning when you start “from scratch," and a series of norms are taken into account, which are specified by ISO/IEC 90003. It has to be noticed that apparently we are contradicting ourselves, when trying to use two fast methodologies that do not require too much documentation such as SCRUM and SP with the definition of ISO 9001. However, it is relatively easy; you only have to define clearly the processes to develop, make a small count of tools and techniques used, and of course make a report of team training.
4) The interaction with the client (academically it is the professor) is essential; that is why sprints and iterations are defined; they must be carried out according to an estimated time, and with very detailed purposes. Every time you show what has been done, up to the moment of the product, the client will show a particular version called a demo, once the demo has been seen; the client will be able to decide if what he/she expressed as a need in the product is actually met.
5) In each of the consultations made to the client, he/she has to define, according to his/her needs , the priorities for each of the activities that require software product, this way the team leader or whoever carries out the master role, can get to the task of coordinating the activities that each of the programming couples has.
6) If it has been clearly defined who does which activities, the programming couple may begin to start their work without the need of being done other extenuating design activities. Here starts the assessment of development processes, since as mentioned above, the programming couples must be held responsible for applying correctly the PSP techniques and improve the development of software. Besides they must apply the software metrics so that it is within internal quality boundaries by means of standard ISO/IEC 9126-3.
7) Within the time the team master has chosen, the job done by programmers will be questioned up to now, if results are favorable, the process will continue. If not, the leader will decide a new strategy so that activities do not get stuck.
8) Within the process of development it is always established continuous improvement and feedback on the part of the teamwork, so an environment of cordiality is created.

9) If things go well up to the moment, a stage is reached where a first version or demo has to be released of an activity of the product, it has to be established with the client if needs were met, to move on to another activity or to start again with the development process.
10) The most important with the member of a high-performance team is working for pleasure and not for obligation; there is a rotation of roles with this purpose, so boredom may be avoided or harasssment.

To grasp a better understanding of how these ten points work, take a look at figure 1, where it can be noticed that there are two content blocks within a main one. The first one helps us to be able to define clearly which processes have to be carried out and as a consequence. there would be the need to figure out the estimations for each of the members who would be in the project, by means of PSP measurement, which in turn will be useful to the project leader to apply correctly the TSP. similarly, there is the second block, which shows us that to apply SCRUM correctly, there would be the need to train the team members in the procedures of PSP. With these two big blocks, can be joined easily, making them correspond bijectively XP with PSP, and SCRUM with TSP. Without forgetting the quality criteria gotten by means of the client requirements, which are measured continuously, by means of metrics of ISO/IEC 25000. Having as basis the definitions of software engineering processes, by means of ISO 90003.

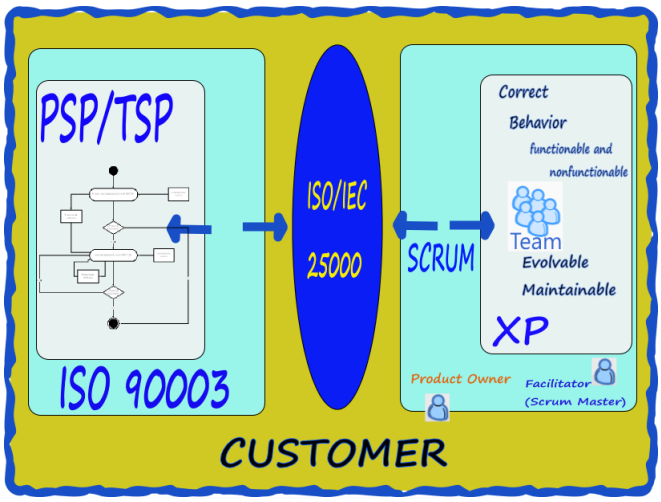


Fig. 1: Diagram of the Proposed Methodology.

4. Conclusions

It can be concluded that beyond any methodology, model or technique used, without underestimating anyone, it has to be assessed for which type of project one wants to apply. In this sense, the present article is considered for academic projects of innovation and development, within the Faculties of Engineering and computer Science in the University of Queretaro, so that our students can assess different models, methodologies, processes, metrics that are systematic, predictable and repeatable, with the purpose of improving productivity in the development and quality of software product of the academic project in innovation and development. Such projects are being carried out in a successful way in rapid projects, concise and usually of small and medium size, even though cases of greater proportion are coming up and these measures are working well.

In general, the underlying philosophy in the use of different models, standards and metrics, in projects of innovations and development, they are to a certain point similar: it is started from some requirements the clearest possible, defining criteria of acceptance or rejection, planning the work that needs to be done, defining points of decision in which to prove that what we are doing coincides with the requirements, or in the opposite case, it proceeds to take timely measures to avoid greater damages, always using the results of measures taken as points of support for decision-making. And above all, learning from our job following a process of continuous improvement.

References

[1] B. Kent (n.d.). *Manifesto for Agile Software Development*: http://www.agilemanifesto.org/ Retrieved 11/15, 2009.
[2] F. Scalone. *Estudio comparativo de los modelos y estándares de calidad del software.* Buenos Aires, Argentina, 2006.
[3] ISO/IEC 25000:2005 “Software Engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE”.
[4] ISO/IEC 90003:2004 “Sofware engineering – Guidelines for the application of ISO/IEC 9001:2000 to computer software”.
[5] ISO/IEC 25001:2005 “Software Engineering –Software product Quality Requirements and Evaluation (SQuaRE) – Planning and management”.
[6] ISO/IEC 9126-1:2001 “Sofware engineering – Product quality – Part I: Quality model”.
[7] ISO/IEC 9126-2:2003 “Sofware engineering – Product quality – Part 2: External metrics”.
[8] ISO/IEC 9126-3:2003 “Sofware engineering – Product quality – Part 3: Internal metrics”.
[9] ISO/IEC 9126-4:2004 “Sofware engineering – Product quality – Part 4: Quality in use metrics”.

[10] ISO/IEC 14598-1:1999 “Information technology –Software product evaluation. Part 1: General overview”.
[11] ISO/IEC 14598-2:2000 “Information technology – Software product evaluation. Part 2: Planning and management”.
[12] ISO/IEC 14598-3:2000 “Information technology –Software product evaluation. Part 3: Process for developers”.
[13] ISO/IEC 14598-4:1999 “Information technology –Software product evaluation. Part 4: Process for acquirers”.
[14] ISO/IEC 14598-5:1998 “Information technology –Software product evaluation. Part 5: Process for evaluators”.
[15] ISO/IEC 14598-6:2001 “Information technology –Software product evaluation. Part 6: Documentation of evaluation modules”.
[16] M. A. Babar. Architecture-Centric Methods and Agile Approaches. Limerick, Limerick, Ireland, 2008. http://dx.doi.org/10.1007/978-3-540-68255-4_39.
[17] *Mountain Goat Software*. (1998-2009). http://www.mountaingoatsoftware.com/terms-of-use Recovered on 15 November 2009.
[18] N. Tomayko. Software Architecture-Centric Methods and Agile Development. In I. Software, *IEEE Software.* 2006, pp. 47.
[19] S. Fraser. How Do Agile/XP Development Methods. San Diego, San Diego, Estados Unidos de Norte America, 2007.
[20] S. M. Priolo. Métodologías Ágiles. En S. M. Priolo, *Métodos Ágiles.* pp. 84.Buenos Aires, Argentina: Gradi S.A. 2009.
[21] W. Humprey. *PSP: A Self-Improvement Process for Software Engineers.* Addison Wesley. 2006.
[22] X. Shu. A Case Study of the Implementation of Agile Methods in a Bioinformatics Project. Calgary, Calgary, Canadá. 2006.