Collaborative Integrated Model in Agile Software Development (MDSIC/MDSIC–M)-Case Study and Practical Advice

José L. Cendejas Valdez[1], Heberto Ferreira Medina[2], Gustavo A. Vanegas Contreras[3]
Griselda Cortes Morales[4], Alfonso Hiram Ginori González[5]
Engineering in Information Technologies, Universidad Tecnológica de. Morelia, Morelia Michoacán, México[1, 3]
Unit of Information Technologies and Communications, Instituto de Investigaciones en Ecosistemas y Sustentabilidad–UNAM
TECNM/I.T. de Morelia, Morelia Michoacán, México[2]
Engineering in Computer Systems, Universidad Autónoma de Coahuila, Monclova Coahuila, México[4]
Computation and Systems, Instituto de Radioastronomía y Astrofísica–UNAM, Morelia Michoacán, México[5]

Abstract—The fast increase of mobile device users based on wider and easier internet access has detonated the development of mobile applications (APP) and web. Therefore, improvement and innovation have become a top priority for businesses and consumer relations. The functional quality and interface aspects in applications (software) drive companies to succeed in mobile apps market competition. This paper introduces an agile software development methodology denominated MDSIC and MDSIC-M focused on rapid application development as required by small and medium software enterprises (SMEs), and results in better quality and competitiveness. MDSIC and MDSIC - M proposes some levels with better practices that should be followed in software development projects. This article also aims to show matching indicators and results of MDSIC and MDSIC - M implementations in software projects, by assessing the needed parameters to generate quality software, and thus align technology with the goals of the organizations.

Keywords—Agile methodology; development Software (web–mobile); MDSIC / MDSIC – M; quality assurance

I. Introduction

Increasing market demand on mobile application development (APP) has detonated this industry, due to higher mobile device user adoption and wider and easier Internet access. Therefore, improvement and innovation has become a top priority in business and consumer relations. Literature does not consider full development environment to programmer demanding best practices on mobile applications development. Functional quality and interface aspects in applications to fulfill Mexican Official Norm (NOM) standards drive companies to succeed in mobile apps market competition. This paper introduces a design methodology oriented to rapid application development fulfillment required by small and medium software enterprises (SMEs), resulting in better quality and competitiveness.

Development of custom made software represents high costs for organizations and many of these projects finally do not meet its minimum requirements. Organizations with different business line want to enhance their information processes with the help of software, made by so called software factories. Those factories can help systematize and improve the processes of organizations.

According to [1] the term “software factories” conceptualizes an organization witch main objective is to produce quality software, implying a specific way of organizing work, with a considerable specialization, as well as processes formalization and standardization. For optimal software development several fundamental elements must converge to obtain a custom-made product that provides proper process functioning in organizations.

Among fundamental elements are: 1) hardware; 2) software; 3) qualified personnel (technically as well as working with processes); 4) project administration; 5) agile models for software construction. The purpose of these elements is to expedite, ease and fulfill different projects of software development towards covering organizations’ objectives.

Therefore, in this paper is presented MDSIC and MDSIC-M as part of the industry. MDSIC and MDSIC-M helps to achieve a product based on norms, quality assessment based on indicators and cover needs of enterprises with line of business in software development. Web and Mobile applications (also known as APPs) have peak in development, represent a growing market and have become a priority for IT developers. Quality request of mobile applications drive companies into market competition under the standards international of quality.

Mobiles software development brings new challenges to software industry, because mobility, interconnection and simpler applications are growing in demand. In many APPs global delivery stores, tendency can be observed towards buying-selling software known as "freemium", a free license with minimum features, which can be changed by paying to access the software full features, upgrades and improvements.

Mobile applications development presents both new techniques and challenges. Recently, agile methodologies had flooded this market. Examples of these methodologies are

Extreme Programming (XP), Scrum and Agile Process Unified (AUP) [2] [3].

In this paper is presented a methodology which is based on the experience of this industry in Mexico (with survey and interviews). This experience allowed building methodological process for MDSIC and MDSIC-M development, under quality standards established by norms to be fulfilled by enterprises committed on web and mobile software development market. Also, the paper presents the results obtained from the companies that develop custom software with the use of MDSIC and MDSIC-M.

II. Literature Review

This section offers a literature review on use of different models for software development; also experiences generated using the software development model implementation (MDSIC / MDSIC - M) in Mexico. Nowadays there is a need to create software based on models that give certainty to enterprises having quality products and allowing a direct impact on their objectives. With the goal that models will aid the enterprises developing software, not otherwise, enterprises end up working for the models.

At this time software has unique challenges, such as: a) Form factors, b) User's technology, c) Usability, d) Design/user interaction, e) Programmers' choice for mobile devices implementation, f) Development processes issues, g) Programming tools, h) User interface design, i) Applications portability, j) Quality and k) Security. Additionally, look for development process time reduction.

One of the best ways to fight complexity in software development is with abstraction decomposition and problem break out. This leads to use of models that allow all the elements mentioned to interact. Business process modeling role in informatics systems (software) construction has a great importance due to these systems grow in scale and complexity, [4]. An example of business process modeling is based in theoretical concepts of the DEMO methodology, which is built upon graphical notations using Petri Networks. Both, DEMO concept and Petri Networks have been studied broadly in different research lines. DEMO methodology was developed and implemented in several real life projects [3]. Therefore, models can be found in all areas such as software engineering.

In [5], it was concluded that: "The software industry remains reliant on the craftsmanship of skilled individuals engaged in labor intensive manual tasks. However, growing pressure to reduce cost and time to market, and to improve software quality, may catalyze a transition to more automated methods". In [5], is mentioned that for the last three decades software development has been immerged in a problematic from which has been difficult to get over. The main issue on this matter is, to develop quality products that satisfy organizations' needs and objectives.

In addition, the software is not aligned with the goals and objectives of the organization. Software is built by IT experts who are dedicated to analysis, design and development, but are never accompanied by experts in the organizational processes that benefit product development in a formal way. There is a need to analyze how to improve the software industry and describe the best technologies that can be used to support this view. "Therefore, it is suggested that the current software development paradigm, based on object orientation, may have reached the point of exhaustion, and models are proposed for its successor". In the last decade, this has progressed compared to what [6], one of the creators of UML estimated in 2002.

According to [6], in that year only 5% of developers used UML in its projects and the majority used it for documentation. In several studies, [3], concluded that: "The model-driven software development (MDSD) was founded with the objective of integrating models and code as participants in software production process. The development of any system software needs to be addressed with two different perspectives: a) the perspective that addresses issues related to the application domain (the problem domain) and b) the perspective that addresses aspects of software technology used to implement the system (the solution domain). The problem domain usually has nothing to do with the software technology. For the end-user, software is a mere tool that should not cause concerns".

Author discusses in [7], the role of models as fundamental in software development to enhance elements of software reuse and facilitate the work of the different roles involved in the process. In many cases the use of models and methodologies for software development requires time, effort and investment, and if the staff is not trained delays may occur in the delivery of software projects. Here is where the models help to solve real projects and provide flexible solutions to the needs of organizations through software development.

There are different models and methodologies that function as support tools for software development. In a recent study about models and methodologies [8], conceptualize the following:

- Software development model is a simplified representation of software development process, presented from a specific perspective.
- Software development methodology: Is a structured approach for software development including system models, notations, rules, designs suggestions and process guidance.

Another way of making software is through agile methodologies, allowing carrying out a more effective and faster tracking scheme. Author in [9] says that agile methodologies follow an iterative approach to build software quickly, where the entire software development life cycle is divided into smaller iterations, which helps minimize overall risk. Agile software development approach refers to the iterative and incremental strategy involving self-organizing teams and functional teams that work together to create software. Some of the existing agile methods are: Crystal Methodologies, Dynamic Software Development Method (DSDM), Lean Software Development, Scrum and Extreme Programming (XP). Table I describes each according to their references.

TABLE. I. DESCRIPTION OF LEADING METHODS FOR AGILE DEVELOPMENT

Method agile	Description	Reference
Crystal Methodologies	A family of methods for co-located teams of different sizes and criticality: Clear, Yellow, Orange, Red, Blue. The most agile method, Crystal Clear, focuses on communication in small teams developing software that is not life-critical. Clear development has seven characteristics: frequent delivery, reflective improvement, osmotic communication, personal safety, focus, easy access to expert users, and requirements for the technical environment.	[10,11, 12, 13]
Dynamic software development method (DSDM)	Divides projects in three phases: pre-project, project life-cycle, and post project. Nine principles underlie DSDM: user involvement, empowering the project team, frequent delivery, addressing current business needs, iterative and incremental development, allow for reversing changes, high-level scope being fixed before project starts, testing throughout the lifecycle, and efficient and effective communication.	[14]
Lean software development	An adaptation of principles from lean production and, in particular, the Toyota production system to software development. Consists of seven principles: eliminate waste, amplify learning, decide as late as possible, deliver as fast as possible, empower the team, build integrity, and see the whole.	[15]
Scrum	Focuses on project management in situations where it is difficult to plan ahead, with mechanisms for ‘‘empirical process control”; where feedback loops constitute the core element. Software is developed by a self - organizing team in increments (called ‘‘sprints”), starting with planning and ending with a review. Features to be implemented in the system are registered in a backlog. Then, the product owner decides which backlog items should be developed in the following sprint. Team members coordinate their work in a daily stand-up meeting. One team member, the scrum master, is in charge of solving problems that stop the team from working effectively.	[16]
Extreme Programming (XP)	Focuses on best practice for development. Consists of twelve practices: the planning game, small releases, metaphor, simple design, testing, refactoring, pair programming, collective ownership, continuous integration, 40-h week, on-site customers, and coding standards. The revised ‘‘XP2” consists of the following ‘‘primary practices”: sit together, whole team, informative workspace, energized work, pair programming, stories, weekly cycle, quarterly cycle, slack, 10-minute build, continuous integration, test-first programming, and incremental design. There are also 11 ‘‘corollary practices”.	[17, 18]

According to [17], the XP methodology receives more bibliographical attention because it applies conceptual premises to solve a problem that is slightly different from the evolutionary development of applications. Author in [19] comments that organizations are focusing their attention to the agile methodology named Scrum. Scrum is used for managing software development, whose main objective is to maximize the return on investment for the company and generate innovation.

Author in [19], proposes that the agile development promotes stakeholder involvement in projects where those stakeholders enable monitoring of the activities, which increases productivity and profit. Agile development encourages users to participate actively in the entire product development. Author in [18] found that "Modern computer software is characterized by continuous change, very short delivery times and an intense need to satisfy customers/users. In many cases, the time-to-market is the most important management requirement. If this requirement is lost, the software project itself may lose its meaning."

In recent years the technology acceptance has been investigated by the theory of diffusion of innovations and models of social psychology [20]. The main focus of the theory of diffusion of innovations and for the adoption of an innovation is communication. Often the diffusion of innovation within a population can occur from a very small proportion, which can be modeled mathematically for selection [21, 22]. The diffusion of an innovation can be a "special kind of communication", it comes from word of mouth and the existence of adopters will depend on the influence of early users.

In [23], the author proposed in his research at the Massachusetts Institute of Technology (MIT), published in MIT Sloan Management Review (MIT SMR) and Deloitte in the spring of 2012 that "social business is an activity that uses social media, social software and social networks to enable more efficient and effective mutual connections between people, information and resources.

These connections can facilitate business decisions, actions and outcomes in different areas of the companies" [24]; report that in the coming years there will be a growing interest in building business models based on social participation, because humans have an instinctive natural desire to improve the lives of their fellowmen when possible.

A real innovative option is the collaborative integrated software development model (MDSIC / MDSIC - M); mention that "the collaborative integrated software development model (MDSIC / MDSIC - M) offers experts an easy way to interact with it through five levels that provide best practices for software development; these levels also consider the basic functions proposed by the Project Management Institute (PMI), which allows generating quality software aligned with organizational goals.

MDSIC / MDSIC - M allows evaluating software quality using a series of indicators that must be considered for optimum performance of a given software. These indicators are supported by quality standards. A key part of MDSIC / MDSIC

-M is the creation of a knowledge base that feeds through social business, which is generated using social networks (Facebook, Twitter, StumbleUpon, Pinterest, etc.), thereby producing a data bank with opinions of experts in software development; propose the use of MDSIC / MDSIC - M through a series of steps that facilitate agile project management and software development.

This model consists of five levels: 1) Level 0: Problem detection; 2) Level 1: Analysis and design; 3) Level 2: Development; 4) Level 3: Implementation; 5) Level 4: Quality indicators. MDSIC also contemplates the five basic functions covered under the Project Management Institute (PMI), which are: 1) Integration of project management; 2) Scope; 3) Time; 4) Cost; 5) Quality. Fig. 1 presents the general structure proposed by the MDSIC including its elements.

Mobile development brings new challenges to software industry, because mobility, interconnection and more simple applications are constantly growing in demand. In many APP Global-Delivery stores, this tendency can be observed, towards buying-selling software known as "freemium" (free license with minimum features that can be changed by paying to access all corners of the APP) upgrading and applications improvement, these characteristics are usually only available when a license is paid [25].

Software development for mobile platforms comes with unique characteristics used for corresponding life cycle stages. Development environment and technologies supporting mobile devices software are different compared with "traditional" development values. Another point of view is associated to restrictions of mobile applications, which is described in [26]. The author mentions two types of those restrictions: 1) constant and inherent evolution and 2) evolution restrictions such as: bandwidth, coverage and security.

On the other hand, inherent limitations, such as limited screen display, limited text capturing capacity (limited keyboard, for instance), memory capacity, processing power, slow startup and execution, are permanent, at least when compared to desktop environments. Programmers have attacked these limitations with agile development approach. Using agile methods for software development has a lot of approval but in some cases it also has opposition.

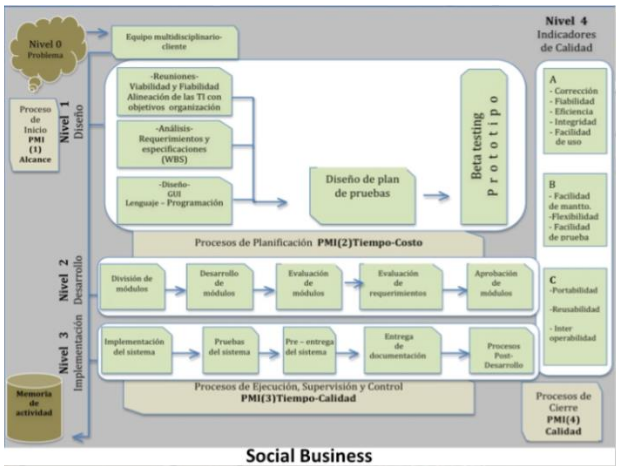


Fig. 1. Collaborative Integrated Software Development Model (MDSIC).

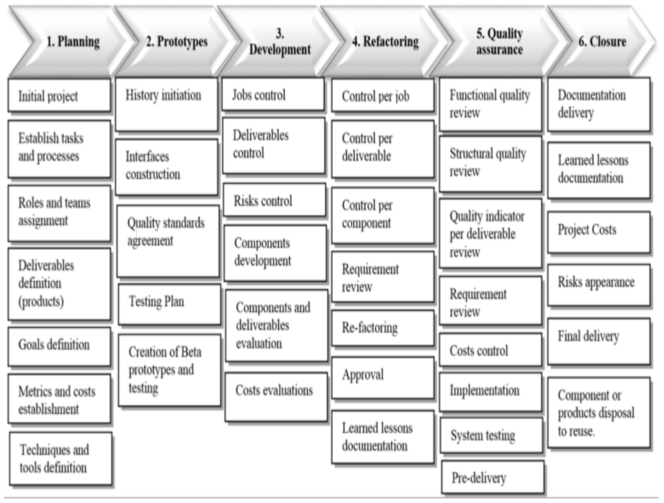


Fig. 2. Stages and Processes in MDSIC-Mobile Methodology with Reuse, Best Practices Approach.

The methodology proposed considers all of the four levels explained in MDSIC (stages 1, 2, 3 and 4) and adds control using scheduler and QA in development teams (stages 2 and 5); who can use a methodology of extreme programming, by pairs, or other method. In Fig. 2 stages to assure a quality mobile development are shown.

There is also uncertainty when distinguishing ad-hoc programming agile methods and development needs. However, as shown in [27], agile methods provide an organized development approach. In [28], a researchers group analyzed publications and agile development advances in the past 10 years, standing out XP methodologies, Scrum, Lean, Crystal, among others.

III. Methodology

The methodology of this research was to implement MDSIC and MDSIC - M in different projects and use its indicators to measure the quality of the software produced. Having identified the problem, the research objectives were established and the nature of the investigation, which defines procedures to obtain the information needed to solve the problem is described.

A fundamental part of this project is to evaluate and measure indicators in SMEs developing mobile applications. In order to do it, a transversal study was performed, and the investigation landscape included: quantitative, field research in software development, quasi-experimental studies and explanative [29]. A synthesis analysis of different models and methodologies found in the literature was driven, from this analysis were gathered variables needed to measure and propose a methodology ad hoc [5] [30]. In Fig. 3, it is shown the methodological process.

Utilizing the main models and methodologies characteristics analyzed, questions were obtained to determine the importance of a mobile application development, using planning edges, costs, risks, quality programming [31] and metrics [32]. To establish the number of probable enterprises, several databases were reviewed, and census population was obtained. The databases inspected were AMITI [33], Yellow

Section (YP) [34], infoisinfo (IiI) [35] and finally it was an investigation of Mexican Enterprise Information System (SIEM) of Economic Development Bureau (SEDECO) [36], giving the next table results, you can find in Table II.

According to previous data, the survey for populations was calculated based on a confidence interval of 95% and 97% [38], Table III shows the results.

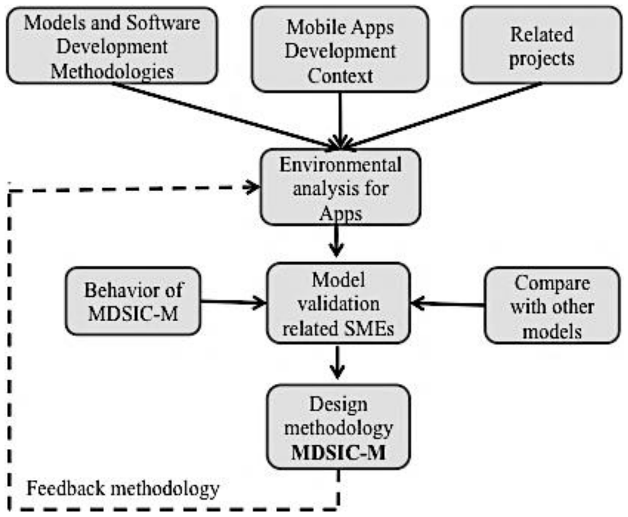

Models and Software Development Methodologies
Mobile Apps Development Context
Related projects
Environmental analysis for Apps
Behavior of MDSIC-M
Model validation related SMEs
Compare with other models
Design methodology MDSIC-M
Feedback methodology

Fig. 3. Methodological Process for MDSIC-M.

TABLE. II. POPULATION ANALYSIS FOR THE APPLICATION OF SURVEYS BASED ON SEARCH TAGS

States	Active population*	YP	IiI	SIEM	Population %
Ags	510,541	18	23	5	6 %
Col	342,702	17	14	12	4 %
Gto	2,412,886	38	32	15	27 %
Jal	3,369,238	75	14	59	37 %
Mic	1,896,174	20	6	24	21 %
Nay	529,436	3	4	5	6 %
Total	9,060,977	171	93	120	

* INEGI [37]. Economically-active population.

TABLE. III. POPULATION CALCULATION WITH A STANDARD ERROR OF 5%

Population Calculation				
Confidence interval	**YP** (171)	**IiI** (93)	**SIEM** (120)	Average
97%	126	78	96	100
95%	119	75	92	95

Surveys were carried out with support of social networks in order to reach enterprises; 70 companies answered, only 38 of those develop mobile applications (54%). In Fig. 4, 5, 6, 7, 8 and 9, the survey's results are shown (including open questions).

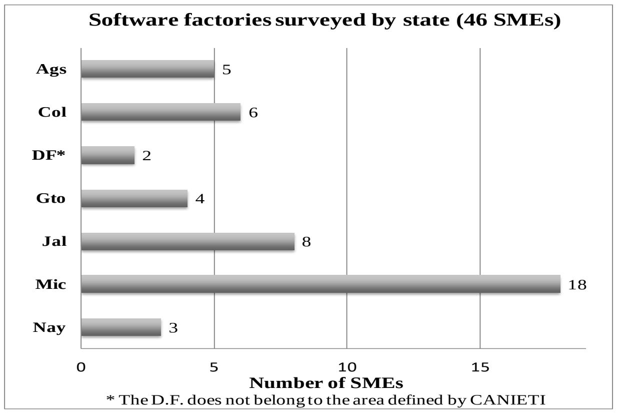


Fig. 4. SMEs Pooled by State on the Region (66% Answered, 46/70).

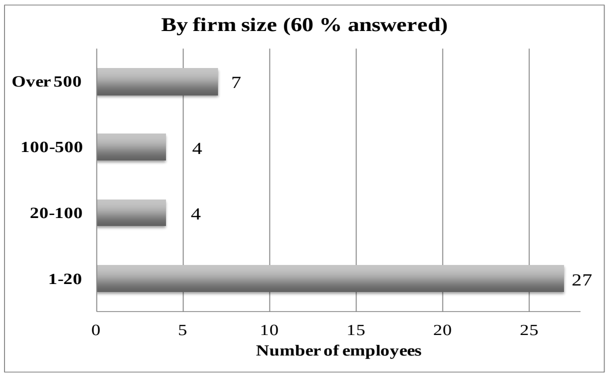


Fig. 5. Enterprise Size, based on Answers (42/70).

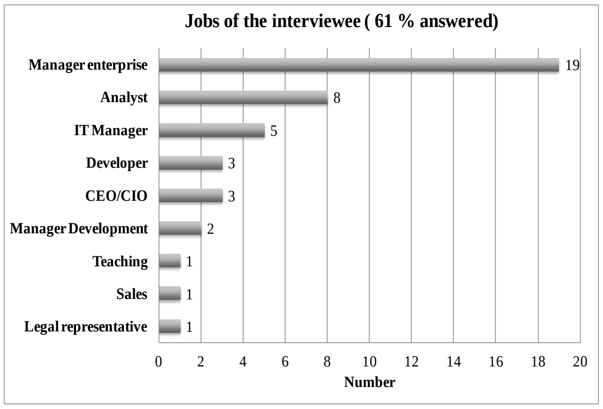


Fig. 6. Current Position on the Ones Pooled (43/70).

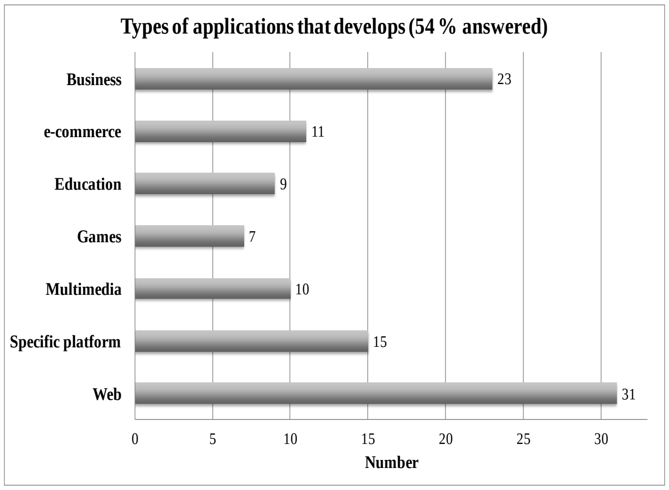


Fig. 7. APP Type Developing (38 SMEs Answered).

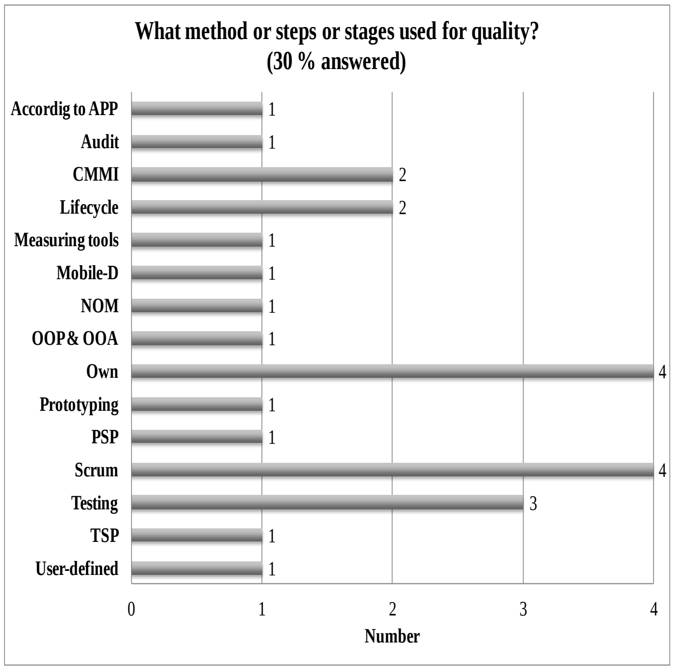


Fig. 8. Quality Analysis Methodology (21/70).

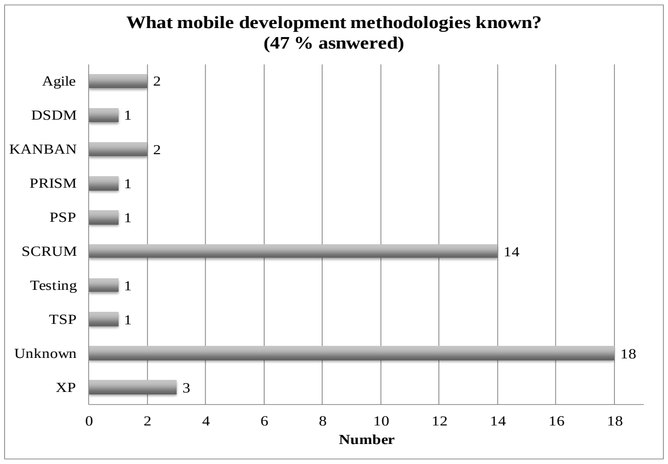


Fig. 9. Mobile Development Methodology Known by SMEs (33/70).

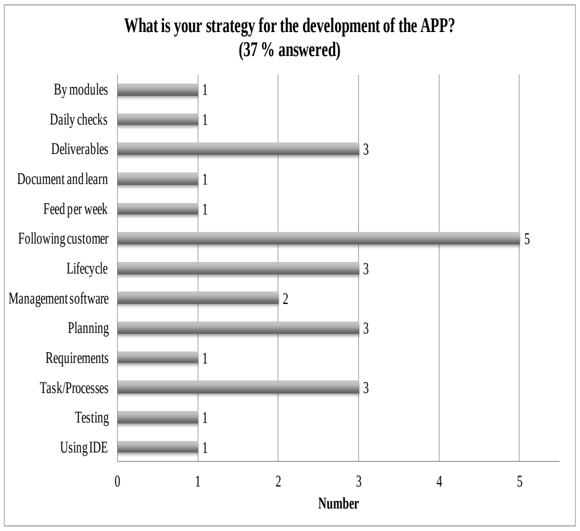


Fig. 10. Strategy to Develop an APP and be Competitive (26/70).

Finally, companies were asked about their strategy to develop an APP and be competitive, in Fig. 10 the result is displayed.

IV. Results

A methodological framework was generated that was used to propose and validate the methodology shown in Fig. 10 in which the elements considered, into account for software development are described. The existing methodologies, the context of mobile development and related projects are the source of this analysis. A model was created and their behavior was observed in SMEs, which helped to provide feedback methodology in order to obtain the best proposal [39].

To guarantee quality in development, the methodology was based on the following components that should be oriented to collaboration and project's plan maintenance: 1) requirements definition (Tasks and deliverables), 2) initial plan (WBS), 3) establishment of WBS and deliverables, 4) construction of a job control planner, 5) deliverables control, 6) risks control, 7) costs and metrics control (establishment), 8) Quality assurance (KPIs and project scope), 9) Collaborative tool usage based on Scheduler, 10) Testing plan and delivers (deliverables) and 10) document learn lections (project information database).

The scheduler is the methodology core; sustains the follow up to the proposed activities in WBS, and in best practices proposed in PMI agile and Lean [40]. MDSIC-M contemplates the next elements, Fig. 11.

An essential part of MDSIC and MDSIC – M is the "activity report", which has a presence through a system that is implemented based on a technology known as "responsive web", which is a way of programming that allows the system to adapt to the size and shape of any device that connects to it. The software accompanying MDSIC/MDSIC M aims to capture and store the information generated from software projects. In addition to creating a knowledge base enhanced by expert developers looking to propose improvements in the processes of software development. This allows collaborative work from its multiuser nature as shown in Fig. 12 and 13.

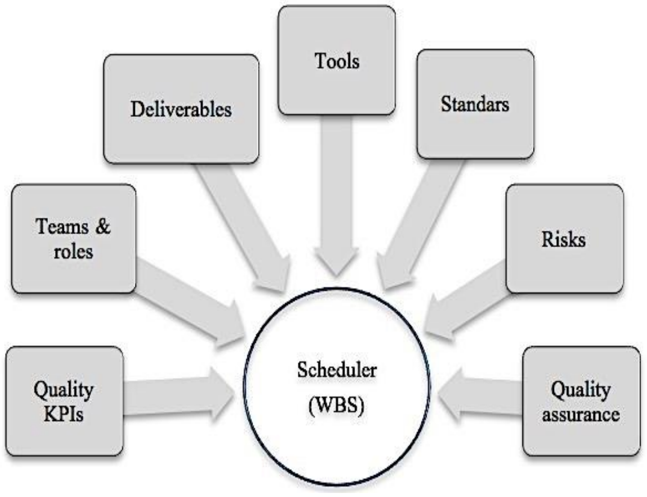
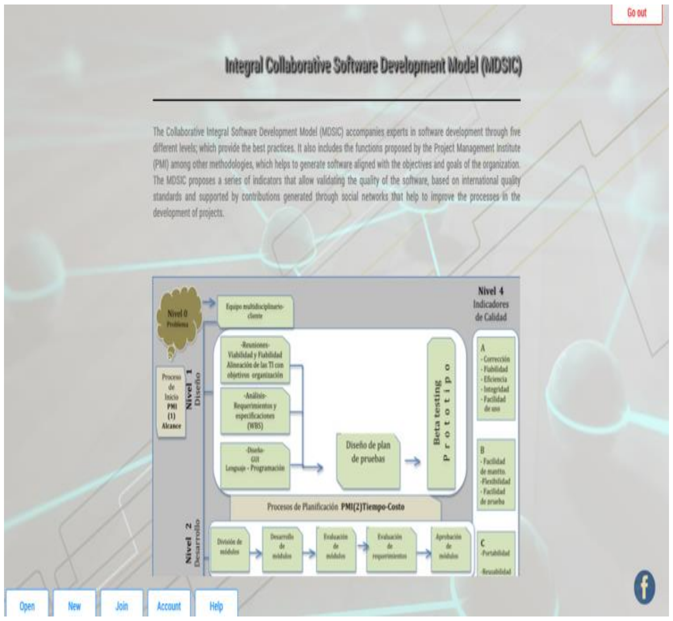


Fig. 11. MDSIC-M Elements to Assure Quality in APP Development.


Go out
Nivel 0 Problema
Equipo multidisciplinario-cliente
Nivel 4 Indicadores de Calidad
Nivel 1 Diseño
Nivel 2 Desarrollo
Diseño de plan de pruebas
Beta testing Prototipo
Procesos de Planificación PMI(2)Tiempo-Costo
Open
New
Join
Account
Help

Integral Collaborative Software Development Model (MDSIC)

The Collaborative Integral Software Development Model (MDSIC) accompanies experts in software development through five different levels; which provide the best practices. It also includes the functions proposed by the Project Management Institute (PMI) among other methodologies, which helps to generate software aligned with the objectives and goals of the organization. The MDSIC proposes a series of indicators that allow validating the quality of the software, based on international quality standards and supported by contributions generated through social networks that help to improve the processes in the development of projects.

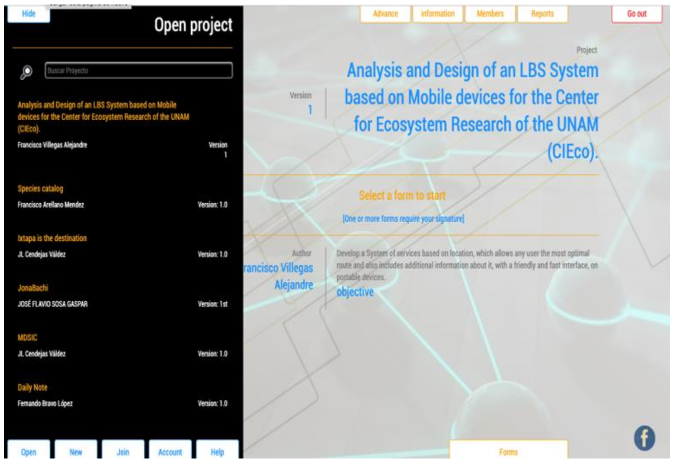
Fig. 12. Welcome Interface in MDSIC v1.0.

The projects developed through MDSIC v1.0 and v2.0 has the facility to measure the progress of these projects through the quality module, which allows measuring the progress of each of the levels. Thus the project manager, quality assurance (QA) and the collaborative team can measure the progress of each project graphically according to plan.

This type of projects seeks to be delivered quickly and having beta prototypes allows an organized development, contemplating quality standards by having processes and quality evaluations [40]. "Global delivery" is left out of this methodology; however, it can be an opportunity to upgrade in the future in order to fulfill the mobile development process. Experimental tests with regional enterprises have been carried out to observe the utility of this methodology and to gather information whether or not is a good alternative for software factories in Mexico, see results in Fig. 14 and 15.

It has been observed that there is a rising demand for Web applications. The proposed methodology can be applied for this type of development and gain a competitive advantage; a future project is to prove this Web development approach and document its behavior.

You can see in Fig. 16 the comparison between different methodologies that use quality indicators.



Analysis and Design of an LBS System based on Mobile devices for the Center for Ecosystem Research of the UNAM (CIEco).
Francisco Villegas Alejandre — Version 1

Species catalog
Francisco Arellano Mendez — Version: 1.0

Ixtapa is the destination
JL Cendejas Váldez — Version: 1.0

JonaBachi
JOSÉ FLAVIO SOSA GASPAR — Version: 1st

MDSIC
JL Cendejas Váldez — Version: 1.0

Daily Note
Fernando Bravo López — Version: 1.0

Project: Analysis and Design of an LBS System based on Mobile devices for the Center for Ecosystem Research of the UNAM (CIEco).
Version: 1

Select a form to start
[One or more forms require your signature]

Author: Francisco Villegas Alejandre
Develop a System of services based on location, which allows any user the most optimal route and also includes additional information about it, with a friendly and fast interface, on portable devices.
objective

Fig. 13. Project Selection Interface in MDSIC v1.0.

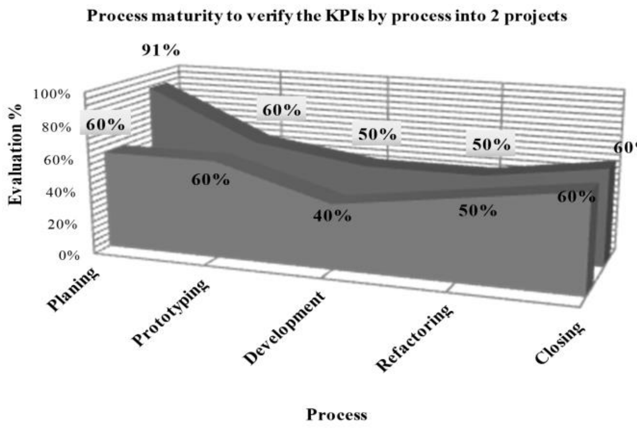


Fig. 14. Verify KPIs in MDSIC-M.

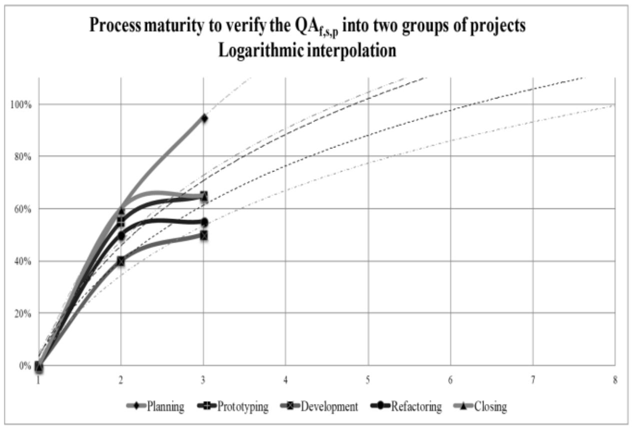


Fig. 15. QA Process Maturity.

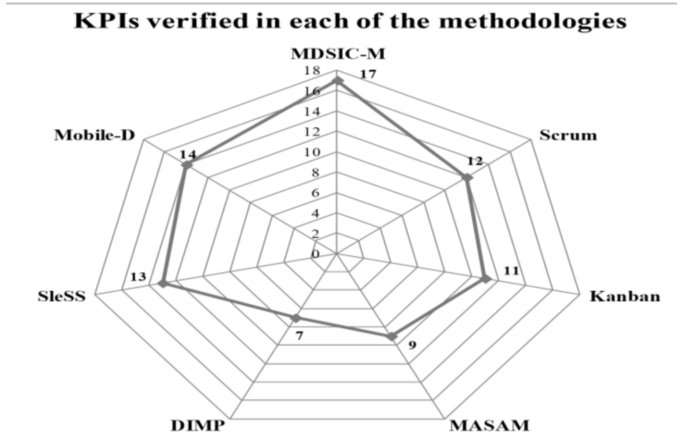


Fig. 16. KPI used in different Methodologies [31].

V. Conclusion

The problems identified in the field of software development in the last three decades is mainly due to not having well defined methods for building software; this can be offset by using the model MDSIC; it has proven to be a tool that helps software development companies to develop projects that line up with the goals and objectives of organizations, thus contributing to their productivity. MDSIC aims to integrate all involved by forming teams of collaborative work that allow significant progress in building the software.

The need for documenting software projects is very important and MDSIC, with its system MDSIC v1.0, enables to register and document all the processes of software development. This application has multi-user features and was designed to function as a responsive technology; MDSIC v1.0 automatically adjusts to any device. This work contributes with relevant information to research focused on software engineering and process modeling, in addition to professionals in the use of agile methodologies, allowing the identification and best practices to achieve success in agile software development. In the area of statistics, this study confirms that research in software engineering can be certified and validated by the multivariate analysis. Furthermore, the work contributes a quantitative research that encourages organizations to use agile principles in software development.

CMMI and MOPROSOFT propose stages or maturity levels on software factory creation. The evaluated KPIs in MDSIC-M will be a significant aid to achieve the objective in accordance with a standard or quality norm. The generation of documentation and reports provided by MDSIC 2.0 offers an easy way to reach this goal. Finally, as an advantage the methodology proposes to store project histories, documenting size, time, costs and risks that took place during the project, allowing the Company to consult this database and not repeating the mistakes, improving its development process. Reused components are the best alternative to be competitive in mobile development market.

Acknowledgment

The authors thank the Cluster TIM de Michoacán (Cluster of Information Technologies of Michoacan,), the Cluster de tecnología de Jalisco (Cluster Technology of Jalisco), Cluster de tecnologías de Colima (Cluster Technology of Colima), Instituto Tecnológico de Morelia, Universidad Tecnológica de Morelia, IIES-UNAM, UPAEP and regional developers for assistance and support answering the survey. We thank the academic group "Unidad de transferencia tecnológica para la construcción de software - TRATEC" for their contribution to this document. We also thank the professor Alberto Valencia G., Atzimba G. López M, Sergio L. Gúzman (IIES), Hugo A. Zavala V. (CIGA) and Alfonso H. Ginori G. (IRyA) for their assistance and support for MDSIC 2.0 software.

References

[1] Piattini V. M. G., & Garzás P. J., 2010. Fábricas de software, experiencias, tecnologías y organización. (2da. Edición) Madrid, España. Editorial Ra – Ma.

[2] Xiaofeng, W.; Kieran, C. and Oisin, C. "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development. Journal of Systems and Software, vol. 85, (Feb 2012), pp. 1287-1299.

[3] De Lara, J.; Guerra, E. and Sánchez, J. Model-driven engineering with domain- specific meta-modelling languages. Software & Systems Modeling, vol. 14, nº 1, (2015), 429-459.

[4] Barjis, J. (2008). The importance of business process modeling in software systems design. Science of Computer Programming, 71(1), 73–87. doi:10.1016/j.scico.2008.01.002

[5] Dietz, J.L.G.(2006), Enterprise Ontology - Theory and Methodology, Springer, New York.

[6] Greenfield, J., Short, K., Cook, S. y Kent, S. (2003). Software factories: Assembling Applications with Patterns, Models, Frameworks, and Tools. Wiley.

[7] Cendejas Valdéz, J. L., Vega Lebrún, C. A., Careta Isordia, A., & Ferreyra Medina, H. (2014). Design of the integrated collaborative model for agile development software in the central-western companies in Mexico. Nova Scientia , 7 (13), 133-148.

[8] Booch, G. (2002). Growing the UML. Software and Systems Modeling, 1(2), 157-160.

[9] Quintero, J. B., & Anaya, R. (2007). MDA y el papel de los modelos en el proceso de desarrollo de software. Redalyc, (8), 131–146. Escuela de ingeniería de Antioquia.

[10] Someerville, I. (2005). Ingeniería del software (septima ed.). (A. B. María Isabel Alfonso Galipienso, Trad.) Madrid, Madrid, España: pp:130-145 Pearson.

[11] Harleen K. Flora, Wang X., Swati V Chande. "An Investigation into Mobile Application Development Process, Challenges and Best Practices". International Journal of Modern Education and Computer Science (IJMECS). 2014.

[12] Cokburn A., Selecting a project's methodology, IEEE Software 17 (4) (2000) 64–71.

[13] Cockburn A., Crystal Clear: A Human-Powered Methodology for Small Teams, Addison-Wesley, 2004, ISBN 0-201-69947-8.

[14] Stapleton, J. (Ed.). (2003). DSDM: Business focused development. Pearson Education.

[15] Poppendieck M., Poppendieck T., Lean Software Development – An Agile Toolkit for Software Development Managers, Addison-Wesley, Boston, 2003, ISBN 0-321-15078-3.

[16] Schwaber K., Beedle M., Agile Software Development with Scrum, Prentice Hall, Upper Saddle River, 2001.

[17] K. Beck, Extreme Programming Explained: Embrace Change, Addi-son-Wesley, 2000, ISBN 0-201-61641-6.

[18] K. Beck, Extreme Programming Explained: Embrace Chage, second ed., Addison-Wesley, 2004, ISBN 978-0321278654.

[19] Schwaber, K. & Sutherland, J., 2011. The Scrum Guide: The Definitive Guide to Scrum, The rules of the game, s.l.: Scrum.org .

[20] Pressman, R. Ingeniería de software. Un enfoque práctico. España,McGraw.Hill, 2006. 61 p.

[21] Bhattacherjee, A., 2000. Acceptance of e-commerce services: the case of electronic brokerages. IEEE Trans. Syst., Man Cybern., Part A: Syst. Hum. 30 (4), 411–420.

[22] Rogers, E.M., 2003. Diffusion of Innovations, 5th ed. Free Press, New York

[23] Kiron, D., Palmer, D., Phillips, A., & Kruschwitz, N. (2012). Social Business : What Are Companies Really Doing? MIT Sloan Management Review , 31.

[24] Yunus, M., Moingeon, B., & Lehmann-Ortega, L. (2010). Building Social Business Models: Lessons from the Grameen Experience. Long Range Planning Elsevier, 43(2-3), 308–325. Doi :10.1016 /j.lrp.2009 .12.005

[25] Zhechao, C.; Au, Y. and Seok, H. Effects of Freemium Strategy in the Mobile App Market: An Empirical Study of Google Play, Journal of Management Information Systems, 31:3, (Mar 2015), 326-354.

[26] Abowd, G.; Hayes, G.; Iachello,G.; Kientz,J. ; Patel, S.; Stevens, M. and Truong, K., Prototypes and paratypes: designing mobile and ubiquitous computing applications, Pervasive Computing, IEEE, (2005), 67 - 73.

[27] Salo, O. and Abrahamsson, P. An iterative improvement process for agile software development. Software Process: Improvement and Practice, (2006). 81-100.

[28] Dingsøyra, T.; Nerurc, S.; Balijepallyd, V.; Brede, N. A decade of agile methodologies: Towards explaining agile software development. Journal of Systems and Software, vol. 85, (Apr 2012). 1213-1221.

[29] Kothari, C. Methods of Data Collection, in Research Methodology methods and thecniques, New Delhi, New age International (P) Limited, Publishers, (2004), 95-151.

[30] Pérez, P. & González C. Guía comparativa de metodologías ágiles (A comparative guide for methodologies agile). Universidad de Valladolid, Escuela Universitaria de Informática. Universidad de Valladolid. (2012).

[31] Rai, A.; Song, H. and Troutt, M. Software quality assurance: An analytical survey and research prioritization, Journal of Systems and Software, vol. 40, no. 1, (jun 1998), 67-83.

[32] Saraiva, J.; de Franca, M.; Soares, S.; Filho F. and de Souza, M. Classifying metrics for assessing Object-Oriented Software Maintainability: A family of metrics' catalogs, Journal of Systems and Software, vol. 103, (Jan 2015), 85-101.

[33] AMITI, "Asociación Mexicana de la Industria de Tecnologías de Información (Mexican Industry Assosiation of Information Technology)," AMITI, [Online]. Available: http://amiti.org.mx/. [Accessed 5 Dec 2014].

[34] Sección Amarilla. Sección Amarilla (Yellow Page), [Online]. Available: http://www.seccionamarilla.com.mx. [Accessed 8 Aug 2014].

[35] Infoisinfo. Directorio de Empresas en México (Mexico Business Directory), infoisinfo, [Online]. Available: http://www.infoisinfo. com.mx/. [Accessed 15 Feb 2013].

[36] SIEM. Sistema de Información Empresarial Mexicano (Mexican Business Information System), SEDECO. [Online]. Available: http://www.siem.gob.mx/siem/. [Accessed 13 Sep 2013].

[37] INEGI. Instituto Nacional de Estadística y Geografía (National Institute of Statistics and Geography), INEGI, [Online]. Available: http://www.inegi.org.mx/sistemas/bie/cuadrosestadisticos/GeneraCuadro .aspx?s=est&nc=618&c=25436. [Accessed 16 Aug 2014].

[38] Hernández, R.; Fernández, C. and Baptista P. Metodología de la investigación (Research Methodology), J. Mares C., Ed., México: McGRAW-HILL, (2010).

[39] Peffers , K.; Tuunanen, T.; Rothenberger, M. and Chatterjee S. A. Design Science Research Methodology for Information Systems Research, Journal of Management Information Systems, 24:3, (Dec 2014), 45-77.

[40] Saraiva, J.; de Franca, M.; Soares, S.; Filho F. and de Souza, M. Classifying metrics for assessing Object-Oriented Software Maintainability: A family of metrics' catalogs, Journal of Systems and Software, vol. 103, (Jan 2015), 85-101.