# Table of Contents

## International Journal of Information Technologies and Systems Approach

Volume 10 • Issue 1 • January-June-2017 • ISSN: 1935-570X • eISSN: 1935-5718
*An official publication of the Information Resources Management Association*

### Special Issue on Software and Systems Engineering Lifecycle Processes and the ISO/IEC 29110 Standards and Guides

### Guest Editorial Preface



### Research Articles





The *International Journal of Information Technologies and Systems Approach* is indexed or listed in the following: ACM Digital Library; Bacon's Media Directory; Cabell's Directories; Compendex (Elsevier Engineering Index); CSA Illumina; DBLP; GetCited; Google Scholar; INSPEC; JournalTOCs; Library & Information Science Abstracts (LISA); MediaFinder; Norwegian Social Science Data Services (NSD); SCIRUS; SCOPUS; The Index of Information Systems Journals; The Standard Periodical Directory; Thomson Reuters; Ulrich's Periodicals Directory

**International Journal of Information Technologies and Systems Approach**
Volume 10 • Issue 1 • January-June 2017

# An Objective Compliance Analysis of Project Management Process in Main Agile Methodologies with the ISO/IEC 29110 Entry Profile

Sergio Galvan-Cruz, Universidad Autónoma de Aguascalientes, Aguascalientes, Mexico
Manuel Mora, Universidad Autónoma de Aguascalientes, Aguascalientes, Mexico
Rory V. O'Connor, Dublin City University, Dublin, Ireland
Francisco Acosta, Universidad Juárez Autónoma de Tabasco, Tabasco, Mexico
Francisco Álvarez, Universidad Autónoma de Aguascalientes, Aguascalientes, Mexico

## ABSTRACT

Software Process Improvement efforts (SPI) are pursued by organizations for improving the overall quality of their software development processes. However, very small entities (VSEs) avoid them by the lack of required financial and other organizational resources. In contrast, VSEs use agile software development methodologies (ASDMs), but these ASDMs do not foster adherence to best scholastic practices promoted by SPI. Furthermore, while a new ISO/IEC standard (29110) has been recently released for VSEs, it was not designed taking account the ASDM approaches. Thus, we investigate the extent of adherence of main ASDMs (two industrial and one academic type) with this new standard. Initial results provide evidence on the strong need to enhance the two industrial ASDMs (XP and SCRUM). In contrast, the academic ASDM (UPEDU) fits the standard very well but it is scarcely used by VSEs. Hence, it is concluded that there is a knowledge gap between the praxis with ASDMs and the recommended scholastic software processes like the ISO/IEC 29110 standard for VSEs.

## KEYWORDS

Agile Software Development Methodology, Compliance Assessment, IDEF0, ISO/IEC 29110, SCRUM, SPI, UPEDU, XP

## INTRODUCTION

Software process models and standards (SPMSs) (SP (like ISO/IEC 12207, ISO/IEC 15504, CMMI) have been developed by international associations for helping to software development organizations to meet the current demands for quality process and product improvements (Succi et al., 1998; Laporte et al., 2008). SPMSs are important for software development organizations because their correct implementation has generated relevant benefits (Clarke & O'Connor, 2012; SEI, 2006). However, according to Laporte et al. (2008) these software process standards "were not written for small projects, small development organizations, or companies with between 1 and 25 employees, and are consequently difficult to apply in such settings." Thus, very small entities (business or teams) while represent a high percentage of software business in the world (OECD, 2005), are under served potential users by normal software process standards and models (O'Connor & Laporte, 2010; 2012)

DOI: 10.4018/IJITSA.2017010105

(Coleman & O'Connor, 2008). To address this problematic situation, the ISO/IEC 29110 software process lifecycle standard (ISO, 2012) was elaborated. This standard focuses on VSEs.

However, such standards do not prescribe a particular software development methodology (SDM) (such as RUP (Kruchten, 2001; or MSF (Microsoft, 2008)), and thus software development teams – from large, medium, small or very small organizations - face a compliance problem of what SDM to use with any particular standard or model (Mora et al, 2009). This compliance problem between process standards or models with SDMs has been studied previously (Gallagher & Brownsword, 2001; Manzoni & Price, 2003; Mora et al., 2010; Irrazabal et al., 2011; Pasini et al., 2013; Fernandez-Monteiro, 2014; Larrucea & Santamaria, 2015; Cruz et al., 2015; Garcia et al., 2016) with the aim to help to organizations in achieving a satisfactory implementation and certification of the selected software process standard or model (Larrucea et al, 2016).

Furthermore, it has been identified that small and VSEs prefer to use Agile Software Development Methodologies (ASDMs) such as: SCRUM and XP among others (Dya & Dingsoyr, 2008; West and Grant, 2010; Stavru, 2014). However, these ASDMs do not foster adherence to best scholastic practices like SPMSs. Additionally, while a new standard (ISO/IEC 29110) (ISO, 2012) has been recently released for VSEs, it was designed independently of the ASDM approach. In this research, thus, we are motivated to study the compliance problem of ASDMs with the new ISO/IEC 29110 standard given the relevance for VSEs of both approaches and the lack of such compliance results in the extant literature. We study the compliance level of two well-known industry-based ASDMs (SCRUM (Sutherland & Schwaber, 2013) and XP (Beck, 1999)), and one academic-based (UPEDU (Robillard et al., 2001)) with the ISO/IEC 29110 standard. Our scope is limited to the Process Management process in this study. A subsequent research will address the Software Implementation process compliance.

This paper continues as follows: the research goals, questions, method, tools and materials are reported in the section 2; the theoretical basis on ISO/IEC 29110 standard and the three ASDMs are reported in the section 3; the compliance analysis are reported in the section 4; a discussion of implications, limitations and recommendations are reported in section 5; and finally the conclusions of this study are reported in section 6.

## Research Goals, Questions, Method, Tools, and Materials

This research pursues two specific goals: 1) to assess an objective compliance level on Roles, Activities-Tasks, and Artifacts to the ISO/IEC 29110 Entry Profile Project Management process within the SCRUM, XP and UPEDU agile methodologies; and 2) to identify a theoretical and practical implications on obtained compliance levels for improving such levels if required. The specific research questions were established as follows: 1) RQ.1 What is the objective compliance level to the Project Management process (on Roles, Activities-Tasks, and Artifacts) defined in the ISO/IEC 29110 standard Entry Profile from the Project Management process (on Roles, Activities-Tasks, and Artifacts) explicitly reported in SCRUM, XP and UPEDU agile methodologies? and RQ.2 What are the theoretical and practical implications required for improving the compliance levels (if required)?

For answering these two research questions, we used an Evaluative-Interpretative research approach rooted in a Conceptual Analysis methodology (Glass et al., 2002; Mora et al., 2008). We performed (adapted from Mora et al., 2008) the following tasks: 1) Knowledge gap identification, 2) Methodological knowledge assessment, 3) Conceptual Analysis, and 4) Conceptual Synthesis. In the first task 1) we formulated the research goals and questions for the identified knowledge gap, and we established the relevance of the knowledge gap. In the second task 2) we organized the required materials to be collected and analyzed, as well as the conceptual analysis tool. In this research the materials were identified as the official documents published for agile methodologies, three free-access electronic process guidelines (SCRUM-EPG, 2008; XP-EPG, 2006; UPEDU, 2016) and three specific books published on each one of them (Blankenship et al., 2011; Pearman & Goodwill,

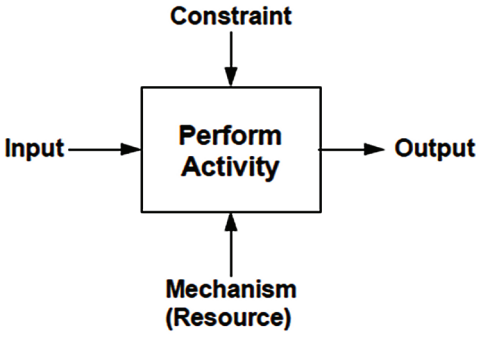
2006; Robillard et al., 2001). The conceptual analysis tool used was IDEF0 (IDEF, 1993). IDEF0 is a conceptual tool used for designing industrial processes (Presley & Liles, 1995). We extended its use for comparing software processes (Galvan et al., 2015).

IDEF0 (Definition of Integration Modeling Function) describes a language (semantics and syntax) and a protocol (rules and technical procedures) to develop structured graphical representations on the functions for potential existent or planned systems (IDEF, 1993). A system in this context can be a physical product, an organization, a process, among others concepts IDEF0 has been widely used for modeling process in diverse domains such as systems engineering (Lambert et al., 2006), industrial engineering (Tang et al., 2000), business process management (Kettinger et al., 1997) and software engineering (Marca & Perdue, 2000; Galvan et al., 2015).

An IDEF0 diagram is a block, which represents a process with four categories of incoming (top, left, and bottom) and outgoing (right) links. Top links correspond to fluxes of CONTROL or CONSTRAINTS (informational fluxes which establish limits, rules and goals for the process). Left links correspond to fluxes of INPUTS. Bottom links correspond to MECHANISMS (human agents, tools and machines required for executing the process) and Right links correspond to fluxes of OUTPUTS as illustrated in Figure 1.

In the third task 3) the first and second authors generated jointly the IDEF0 diagrams (in two levels) for the Project Management processes conducted in the ISO/IEC 29110 (Entry Profile), and the three SCRUM, XP and UPEDU agile methodologies. These IDEF0 diagrams were reviewed by third author who is an Editor of ISO/IEC 29110-2-1:2015 (O'Connor & Laporte, 2010; 2014). After an iterative review process for minor changes, final IDEF0 diagrams were decided upon. Based on these diagrams, and textual descriptions from the selected materials, the first and second authors elaborated detailed descriptive tables on roles-characteristics, activities-tasks and artifacts-characteristics for the Project Management process of the ISO/IEC 29110 standard, and the SCRUM, XP and UPEDU agile methodologies. We used three scales for these evaluations. The first scale was to assess the compliance objective level of operational characteristics of Roles, Activities-Tasks and Artifacts with four levels: null (when the required item of the ISO/IEC 29110 standard was not explicitly reported in the agile methodology), low (when the required item of the ISO/IEC 29110 standard was explicitly but minimally reported in the agile methodology), moderate (when the required item of the ISO/IEC 29110 standard was explicitly and partially reported in the agile methodology), and high (when the required item of the ISO/IEC 29110 standard was explicitly and totally covered in the agile methodology by using the same or a similar nomenclature). The second scale was used to weight the compliance level of the elements within of each roles-characteristics, activities-tasks and artifacts-characteristics evaluations for generating an overall compliance level of activities, roles and artifacts. The scale used was from 0 to 100 percent. The agreed weights are reported in section 4. The third scale was used to weight the overall compliance level of roles, activities, and artifacts to

**Figure 1. IDEF0 Diagram**



integrate a final compliance value. The scale used was from 0 to 100 percent. The agreed weights are reported also in section 4.

These analyses were conducted in several iterations by the first and second author until the elimination of interpretative discrepancies. The third author conducted a review of these assessments and minor changes were corrected. Finally, in the fourth task 4) the first, second and third authors elaborated the synthesis of the obtained results: main findings, theoretical and practitioner's contributions for improving the compliance levels of the agile methodologies, and methodological limitations of this research. The fourth and fifth authors conducted a general review and proofread of the paper. Your final suggestions were included in the final version of this article.

## Theoretical Background

### *The ISO/IEC 29110 Standard Entry Profile*

The ISO/IEC 29110 Software Process Lifecycle standard provides a process model developed for organizations classified as VSEs (business organizations or development teams from 1 to 25 people) (O'Connor & Laporte, 2010; 2014; Laporte et al., 2013). The standard has two main process areas: Project Management and Software Implementation (ISO, 2012). Project Management (PM) aims to establish and carry out the tasks of the software implementation and Software Implementation (SI) aims to systematically analyze, design, construction, integration and testing of software products processed according to specified requirements (ISO, 2012), as illustrated in Figure 2. At present, there have been proposed four profiles: Entry, Basic, Intermediate and Advanced, but only the first two have been released. In this research, we study the ISO/IEC 29110 Entry Profile Project Management process. IDEF0 diagrams were used as conceptual tools to identify and understand the roles, activities and artifacts proposed in the ISO/IEC 29110 standard. The Figure 3 reports the IDEF0 diagram of the overall ISO/IEC 29110 standard.

This Figure 3 reports two INPUTS ("PM.SI. Statement of Work (customer)" and "PM. SI. Change Request (customer)"); five CONTROLS ("PM Quality Control Issues," "PM Track/ Evaluation Changes," "PM Risk Issues," "PM Budget Control," and "PM Schedule Control"); three MECHANISMS ("Work Team," "Project Manager" and "Customer"); and seven OUTPUTS ("PM. SI. Project Plan," "PM.SI. Acceptance Record (customer)," "PM.SI. Meeting Record (customer),"

**Figure 2. Structure of the ISO/IEC 29110 Standard**

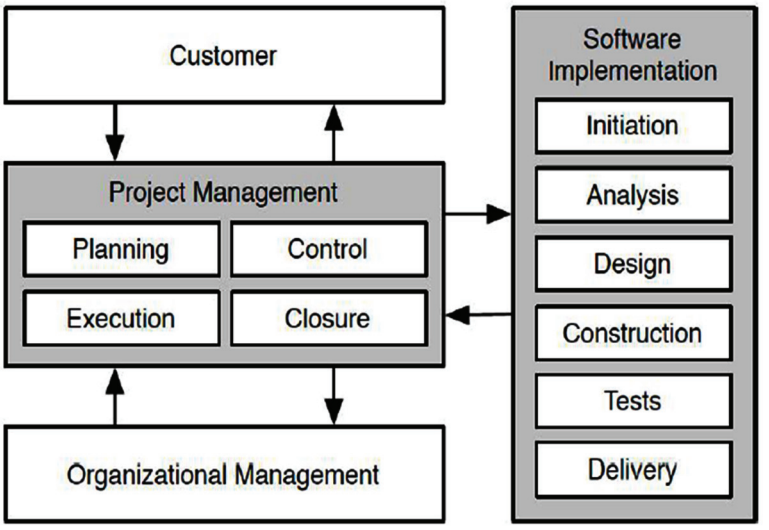
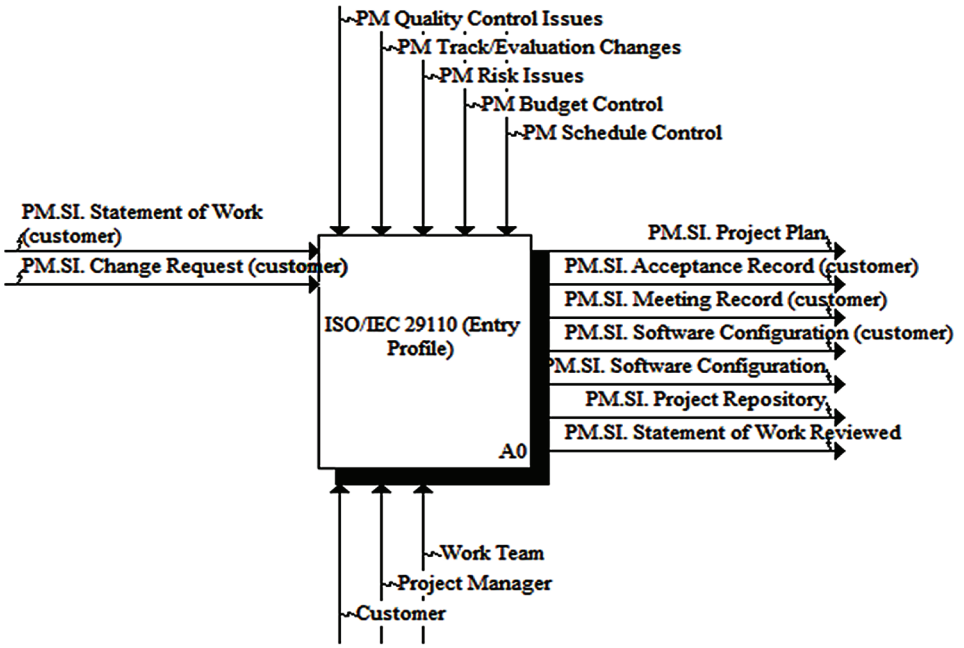


Figure 3. Diagram IDEF0 of the ISO/IEC 29110 Standard




"PM.SI. Software Configuration (customer)," "PM.SI. Software Configuration," "PM.SI. Project Repository" and "PM.SI. Statement of Work Reviewed").

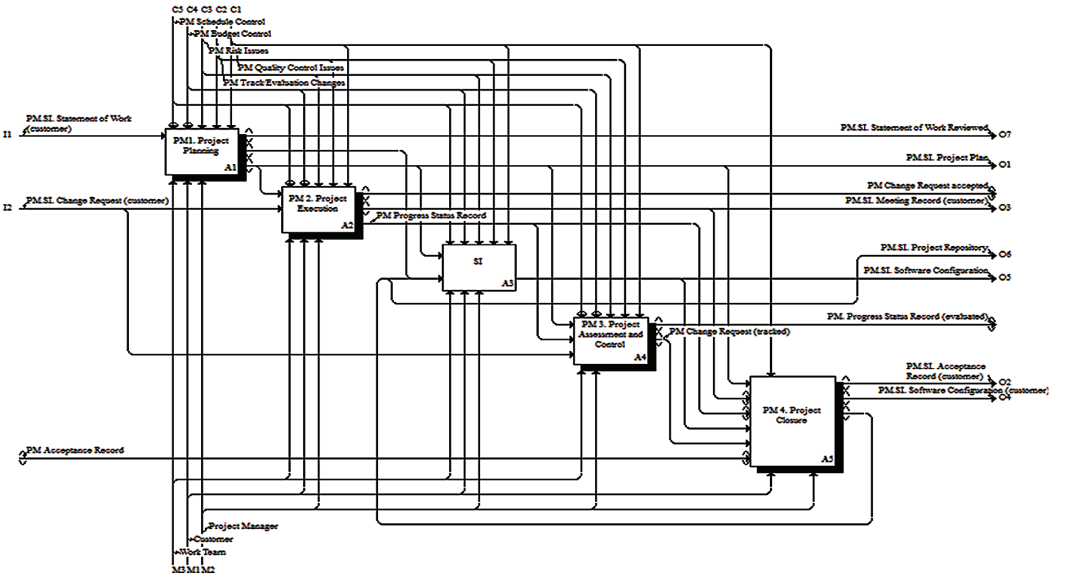
IDEF0 diagrams can be hierarchically decomposed. The Figure 4 shows the next internal level of the ISO/IEC 29110 Project Management Process. We have also included the ISO/IEC 29110 Software Implementation process as activity in this level to remark the importance in the relationship between PM and SI category of processes. The Figure 4 shows that the ISO/IEC 29110 PM process consist of four activities: PM 1. Project Planning, PM 2. Project Execution, PM 3. Project Assessment and Control, and PM 4. Project Closure.

As per Figure 4 shows, PM 1. Project Planning activity has a unique INPUT: "PM.SI. Statement of Work (customer)" which is provided by Customer; and three OUTPUTs: "PM.SI. Statement of Work Reviewed," "PM.SI. Project Repository" and "PM.SI. Project Plan." Its CONTROLs are the total five identified for the overall Project Management process: "PM Quality Control Issues," "PM Track/Evaluation Changes," "PM Risk Issues," "PM Budget Control" and "PM Schedule Control." Its MECHANISMs (roles) are: "Project Manager," "Work Team" and "Customer."

The second activity PM 2. Project Execution implements the "PM.SI. Project Plan" and monitors the status of the software project through meetings with the customers and tracks it carefully until its closure. This activity has two tasks: PM 2.1 Monitor and record project status, and PM 2.2 Conduct meetings with customers (for changes or updates). This activity has two INPUTS: "PM.SI. Change Request (customer)" which might or not be present in any project, and "PM.SI. Project Plan"; and three OUTPUTS: "PM Change Request accepted," "PM Meeting Record (customer)" and "PM Progress Status Record." Its CONTROLs are the five identified for the overall PM process: "PM Quality Control Issues," "PM Track/Evaluation Changes," "PM Risk Issues," "PM Budget Control" and "PM Schedule Control." Its MECHANISMs are also: "Project Manager," "Work Team" and "Customer."

The next block in the Figure 4 is the overall SI process. This process is naturally not part of the PM process. However, we report it because in order to understand the PM process, it is necessary understand the interactions of PM with SI. The process of SI starts with the revision of the "Project

**Figure 4. IDEF0 Diagram for Project Management Process ISO/IEC 29110 Entry Profile**




Plan" given that this part is the base for the activities on SI process. The activities of SI are: SI.1 Software Implementation Initiation, SI.2 Software Requirements Analysis, SI.3 Software Component Identification, SI.4 Software Construction, SI.5 Software Integration and Tests, and SI.6 Product Delivery. SI process has two INPUTS linked from the PM approach: "PM.SI. Project Plan" and "PM.SI. Project Repository." SI process has one core OUTPUT defined as: "PM.SI. Software Configuration." This OUTPUT is linked with PM 4. Closure activity with the goal to get the customer's acceptance to end the project. In summary, SI process accounts for the core engineering activities for producing a software product. In this research, the focus in on PM process and SI process will be not discussed more except for their interrelationships with PM process.

PM 3. Project Assessment and Control is the third activity of PM process. It has three tasks: PM3.1 Evaluate project progress against project plan, PM3.2 Evaluate and track customer asked changes, and PM 3.3 Establish corrective actions, if required. This activity has three INPUTs: "PM.SI. Project Plan," "PM Progress Status Record," and "PM.SI. Change Request (customer)"; and two OUTPUTs: "PM Progress Status Record (evaluated)" and "PM Change Request (tracked)." Its CONTROLs are the five identified for the overall Project Management process: "PM Quality Control Issues," "PM Track/Evaluation Changes," "PM Risk Issues," "PM Budget Control" and "PM Schedule Control." Its MECHANISMs are also: "Project Manager," "Work Team" and "Customer."

Finally, the last activity of PM process is PM 4. Project Closure. It has two tasks: PM 4.1 Formalize the completion of the project, and PM 4.2 Update the project repository. In the first task is tried to formalize the completion of the software project through getting the "PM Acceptance Record" already signed (for part of the customer) which is the main overall OUTPUT of the PM process. A signed document implies that the client has accepted the quality of the software product agreed in the software project. In the second task is necessary to update the "PM Project Repository" to keep evidence of the ended project. These documents will be useful for auditing procedures, statistics and software improvement initiatives. For this last activity we can identify six INPUTs: "PM.SI. Project Plan," "PM.SI. Meeting Record (customer)," "PM Progress Status Record," "PM.SI. Software Configuration," "PM Change Request (tracked)" and "PM Acceptance Record"; and three OUTPUTs: "PM.SI. Acceptance Record (customer)," "PM.SI. Software Configuration (customer)"

and "PM Project Repository" already updated. Its CONTROLs are the total five identified for the overall Project Management process: "PM Budget Control," "PM Schedule Control," "PM Quality Control Issues," "PM Track/Evaluation Changes," and "PM Risk Issues." Its MECHANISMs are also: "Project Manager," "Work Team" and "Customer." This last activity is key for delivering the software product that it has been generated through the different phases.

Hence, the PM process of the ISO/IEC 29110 standard Entry Profile reports a set of Roles, Activities-Tasks and Artifacts expected to exist in any software process that claims to fit this standard.

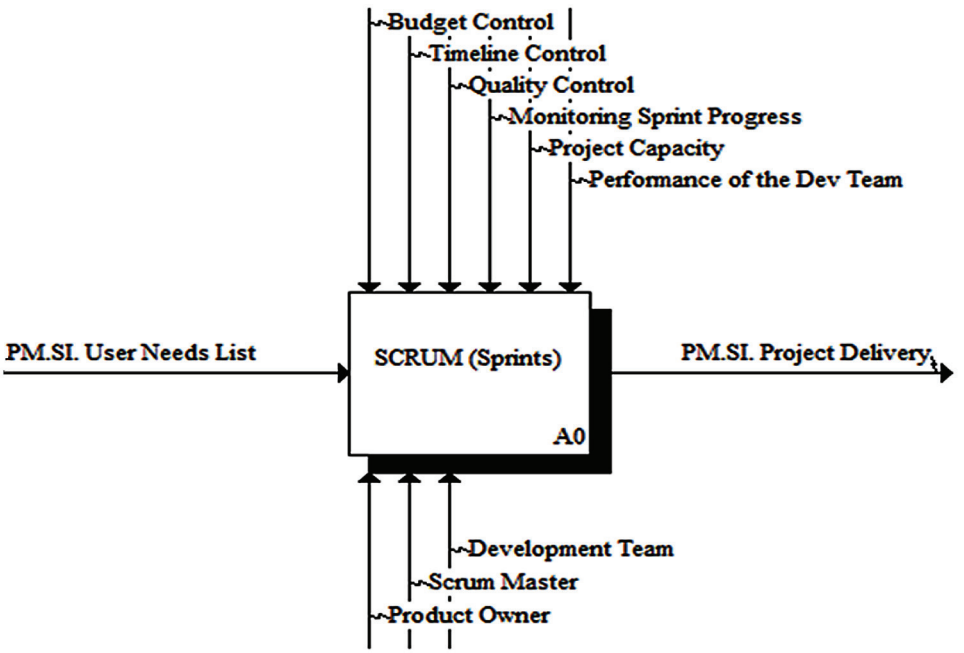
## *Review of the Agile Software Development Methodologies: SCRUM, XP and UPEDU*

### The SCRUM Agile Methodology

The Figure 5 describes graphically the high-level IDEF0 elaborated for SCRUM. SCRUM has one INPUT called "PM.SI. User Needs List," and one OUTPUT called "PM.SI. Project Delivery." The INPUT "PM.SI. User Needs List" refers to a list of everything that it will need in the project (initial raw requirements). This INPUT will be reviewed in "PM Planning (Pre-Game)" activity. This INPUT is a core one because it describes the user needs and involves the three roles in SCRUM method. The OUTPUT "PM.SI. Project Delivery" refers to the whole ended software product with all the requirements completed according the customer requirements.

Three main MECHANISMs (roles) are identified in SCRUM: "Product Owner," "Scrum Master," and "Development Team." The "Product Owner" is responsible of the product backlog (its content, availability and ordering). The "Scrum Master" can be considered the SCRUM expert and project leader that will interact with the other MECHANISMs for leading and guiding them toward the end goal. The "Development Team" "consists of professionals who do the work of delivering a potentially releasable increment of done product at the end of each Sprint" (Sutherland and Schwaber, 2013, p. 5). We identified also several main CONTROLs in SCRUM. They are: "Budget Control," "Timeline Control," "Quality Control," "Monitoring Sprint Progress," "Project Capacity" and "Performance of

**Figure 5. General IDEF0 Diagram of SCRUM Agile Methodology**



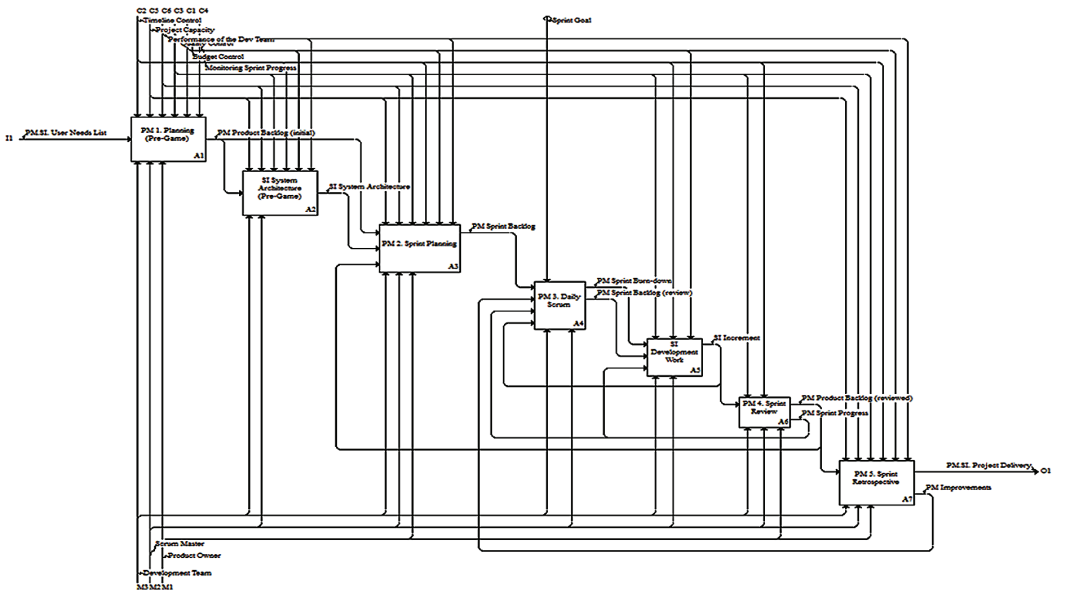
the Dev. Team." These CONTROLs manage the correct execution of the main activities of SCRUM. The Figure 6 describes graphically the next level of activities of SCRUM by using the IDEF0 diagram.

SCRUM can be structured in five main activities for Project Management process: PM 1. Planning (Pre-Game), PM 2. Sprint Planning, PM 3. Daily Scrum, PM 4. Sprint Review, and PM. 5 Sprint Retrospective. Moreover, SCRUM includes two activities for Software Implementation process: SI System Architecture (Pre-Game) and SI Development Work.

The first activity in SCRUM is PM. 1 Planning (Pre-Game). It includes all the roles in the methodology ("Product Owner," "Scrum Master" and "Development Team"). The only INPUT in the first activity is "PM.SI. User Needs List." While this task can be considered a SI one, it is reported here because key estimations on Sprint iterations, their costs and schedules are realized. Furthermore, in this activity the INPUT must be translated in specific user requirements (included in the OUTPUT "PM Product Backlog (initial)"). The "PM Product Backlog (initial)" is an ordered list of everything that might be needed in the product and is the single source of requirements for any change to be made to the product. It evolves as the product and the environment in which it will be used evolves. It also contains the cost and schedule estimations for the whole project. All roles ("Scrum Master," "Development Team," and "Customer") need to discuss on important decisions which will determine the future quality of the software product. This activity produces the initial OUTPUT "PM Product Backlog (initial)," which will be used in next activities of "SI System Architecture (Pre-Game)" and "PM Sprint Planning." This activity includes all the CONTROLS with the purpose to establish a solid mechanism of control and quality in this first activity.

The next activity is actually from SI process: "SI System Architecture (Pre-Game)." It is described here for obtaining a better integrated view of the PM processes. This activity receives as INPUT the "PM Product Backlog (initial)," from the first activity "PM 1. Planning (Pre-Game)." In this SI activity the "Scrum Master" and "Development Team" from a technical point of view, must elaborate the overall architecture for the software product. A "SI System Architecture" is a high-level design which defines the overall structure and behavior of the projected software product. This activity generates only one OUTPUT "SI System Architecture" and this item impacts in "PM 2. Sprint

**Figure 6. IDEF0 Diagram of SCRUM Main Activities**



Planning." It includes all the controls that we can found in the methodology and it only include two roles: "Development Team" and "Scrum Master."

The second activity for PM processes in SCRUM is PM 2. Sprint Planning. This activity can be repeated many times as be necessary (e.g. based on the required number of Sprints which was defined in the PM 1. Planning (Pre-Game) activity). Sutherland and Schwaber (2013; p.p. 7) defines the concept of Sprint as "a time-box of one month or less during which a done, useable, and potentially releasable product Increment is created." This activity has a feedback through the "PM Product Backlog (reviewed)" item. This item is an OUTPUT of the PM activity of SCRUM: PM 6. Sprint Review. Also, PM 2. Sprint Planning activity involves all roles in SCRUM. The suggested time-box for a "Sprint Planning backlog" is as maximum of 8 hours. The "Scrum Master" – in a role of project leader- must manage and teach to SCRUM "Development Team" all of the tasks in this initial activity. All roles collaborate to understand the work of each artifact "PM Sprint Backlog."

The next main activity in SCRUM is PM 3. Daily Scrum. This activity refers to coordinate activities and create a plan for a next job day. It refers to a short time-boxed event (15 minutes approximately), and in this period the "Development Team" synchronizes activities and creates a plan for the next working day (next 24 hours). A PM 3. Daily Scrum is conducted during the overall software project with the purpose to reduce misunderstandings, conflicts, and other team-based problems (Basri & O'Connor, 2011), to improve the communication, to encourage quick decision-making and to improve the level of knowledge of the team's members (Basri & O'Connor 2010). The "Development Team" uses PM 2. Daily Scrum like a monitoring tool of the progress toward the Sprint Goal. Thus the "Development Team or team members meet after the Daily Scrum for detailed discussions, or to adapt, or re-plan, the rest of the Sprint's work" (Sutherland and Schwaber, 2013, p. 10). Also, the "Scrum Master" is the principal responsible and it leads the meeting. In this activity we find four INPUTS: "PM Sprint Backlog," "SI Increment," "PM Sprint Progress" and "PM Improvements." "PM Sprint Backlog" is the product of the "Development Team" defined by the Sprint Goal. This INPUT comes from PM 2. Sprint Planning activity. "SI Increment" is "the sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints" (Sutherland & Schwaber, 2013, p. 14). This INPUT is a feedback of the main overall SI Development Work process. "PM Sprint Progress" is a feedback from the PM 3. Sprint Review activity, and is also known as "PM Task Board." "PM Improvements" defines the things to improve the product and the development of the project. This INPUT is a feedback, which comes from PM 4. Sprint Retrospective activity.

Furthermore, this third activity has two OUTPUTs: "PM Sprint Backlog (reviewed)" and "PM Sprint Burn-down Chart." "PM Sprint Backlog (reviewed)" is the result of the different agreements in the PM 3. Daily Scrum activity. "PM Sprint Burn-down Chart" is a product derived from a technique to forecast the progress of some specific tasks in the project. This item reports the estimated work remaining in the sprint and it is calculated daily and graphed, resulting in a "PM Sprint Burn-down Chart."

The SI Development Work process accounts for the overall engineering activities in SCRUM. This process in not from PM process category but it is included here for understanding the relationships between PM and SI processes. In this process we can identify three INPUTs: "PM Sprint Backlog (reviewed)" and "PM Sprint Burn-down Chart" from PM 3. Daily Scrum, and "PM Sprint Progress" (also called "PM Task Board") as a feedback from the PM 4. Sprint Review activity. This process has only one core OUTPUT: "SI Increment." The MECHANISMs participating in this overall process are: "Scrum Master" and "Development Team." Regarding its CONTROLs, we identify the following ones: "Timeline Control," "Quality Control" and "Budget Control." This process is very important because the different Sprints are executed here. In this process, also the "Development Team" is a key MECHANISM since they are the professionals for doing the software product through incremental Sprints. The item OUTPUT of "SI Increment" is also relevant and Project Management process because this output will be used as core INPUT in the activities of: PM 3. Daily Scrum and PM 4.

Sprint Review. In PM 2. Daily Scrum activity the "SI Increment" is used in every daily meeting to enhance and perfect the goals that it had been defined at the start of the project. In PM 4. Sprint Review activity the "SI Increment" is used in the review session with the goal to generate a well-defined "PM Product Backlog (reviewed)."

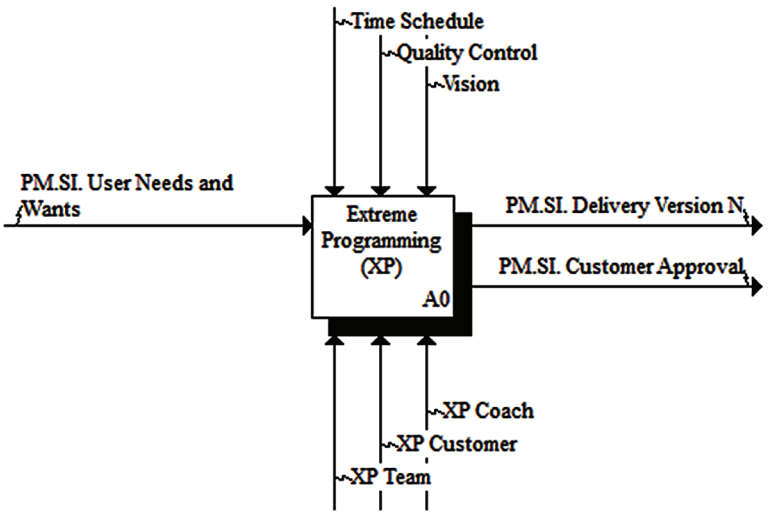
The next SCRUM activity is PM 4. Sprint Review. This activity can be considered of high criticality if we consider the number of involved MECHANISMs and CONTROLs. According to Sutherland and Schwaber (2013; p. 11). This activity "is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint." This activity needs strong collaboration between the "Scrum Team" and stakeholders through informal meeting for the purpose of feedback and foster collaboration. This activity receives as INPUT the "SI Increment," and generates as result (OUTPUT) updates for some items in the "PM Product Backlog (reviewed)" and "PM Sprint Progress." These changes will be considered in the next execution of a Sprint.

The final activity of SCRUM called PM 5. Sprint Retrospective occurs between the immediate past PM 4. Sprint Review activity and the next PM 2. Sprint Planning activity. An ending very important issue here is the verification of the status of done in the "PM Sprint Backlog (reviewed)" (main INPUT) by the "Development Team" according with the initial plans and the agreed overall software project metrics. Besides, the "Development Team" needs to identify the improvements that will implement in the next Sprint through the OUTPUT of "PM Improvements." PM 5. Sprint Retrospective activity is a formal way to focus on the inspection and adaptation issues that happens in any software project. The most important result (OUTPUT) of this fourth activity and in general of all of the SCRUM method is the "PM.SI. Project Delivery." This last item finally will be verified according with the customer's needs and goals.

### The XP Agile Methodology

Figure 7 shows the general IDEF0 diagram for the XP agile methodology. According to Figure 7, there is one unique INPUT in XP: "PM.SI. User Needs and Wants." "PM.SI. User Needs and Wants" are no structured requirements which contain the requests and requirements from the customer for a new software product that will be developed. They will be transformed in the internal OUTPUT "PM User Stories (Requirements)" and will be used for the "XP Team" to estimate the time, costs, tests, risks and other important factors in the new software project. The "PM User Stories (Requirements)"

**Figure 7. IDEF0 Diagram for XP**



item tries to avoid details like specific technology, databases, algorithms, and so on. Each user story will be used to estimate the effort for next release.

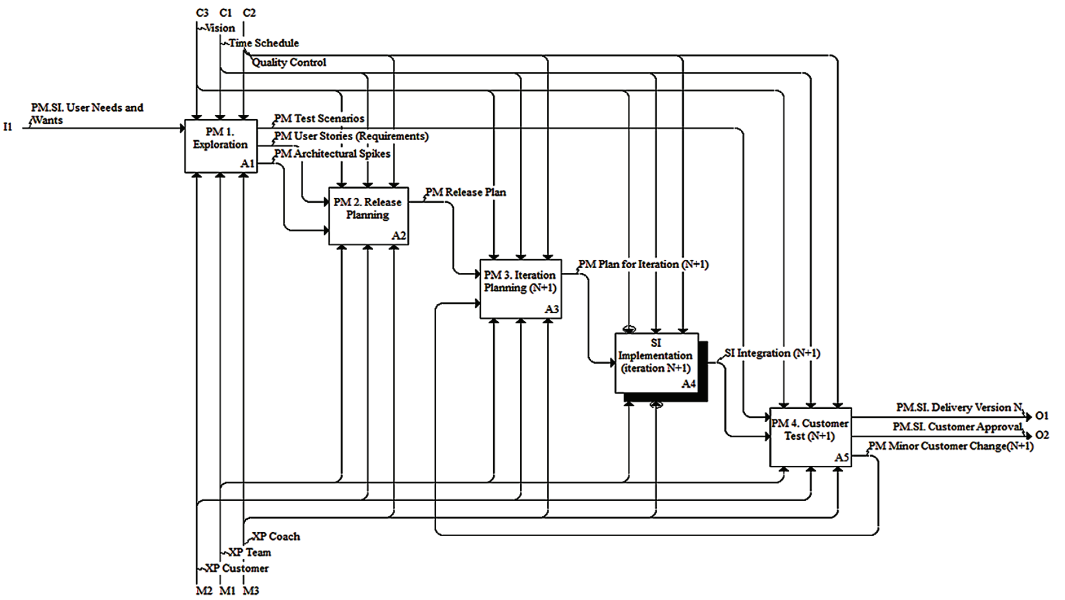
XP has the following MECHANISMs (roles): "XP Team," "XP Coach" and "XP Customer." "XP Team" is formed for different technical roles according with XP official documents (i.e. "Programmer," "Administrator," "Tester," "Tracker," "Consulter" (optional role) and "Doomsayer" (optional role)). "XP Coach" is responsible for the evolution of the project. The coach must have good technical skills. Finally, "XP Customer" specifies user stories (what to do), is responsible of provide the requirements, verifies the evolution of the software project, and accepts the final software product.

XP has three essential CONTROLs to manage the correct execution of XP methodology: "Time Schedule," "Quality Control" and "Vision." Finally, XP has two general OUTPUTs denoted as "Project Delivery Version N" and "PM Customer Approval." "PM Delivery Ver-sion N" represents the product (e.g. the current version generated in release N) obtained after a release. It must be noted that a project could generate different versions (products from several releases). "PM Customer Approval" represents the acceptance about the advance of every version (product of a release). The next Figure 8 describes graphically the next level of activities of XP by using the IDEF0 diagram.

XP can be structured in four Project Management activities: PM 1. Exploration, PM 2. Release Planning, PM 3. Iteration Planning (N+1), and PM 4. Customer Test N+1) and one overall SI process (SI Software Implementation (iteration N+1)).

In the first main activity of PM 1. Exploration, the unique INPUT used is "PM.SI. User Needs and Wants." This INPUT serves as basis for generating 3 OUTPUTS: "PM User Stories (Requirements)," "PM Test Scenarios" and "PM Architectural Spikes." The user (assisted by "XP Team") writes "PM User Stories (Requirements)" and "PM Test Scenarios." In the event of developers do not understand adequately the "PM User Stories (Requirements)," also, they can elaborate an additional feature or features into another OUTPUT called "PM Architectural Spikes." They are small prototypes without implementation details and they are used for exploring unclear customer requests. The CONTROLs in this activity are the overall "Time Schedule," "Quality Control" and "Vision." The MECHANISMs involved in this first activity are the three roles: "XP Customer," "XP Team" and "XP Coach."

**Figure 8. IDEF0 Diagram of Main Activities in XP Methodology**



The second activity is PM 2. Release Planning. In this activity, the time effort and number iterations required for elaborating the planned release are estimated. It produces the OUTPUT "PM Release Plan." This activity requires the INPUTS "PM User Stories (Requirements)" and "PM Architectural Spikes (if available). The CONTROLs in this activity are the overall "Time Schedule," "Quality Control" and "Vision." The MECHANISMs involved in this second activity are again the three roles: "XP Customer," "XP Team" and "XP Coach." Every planned iteration is detailed carefully with an approach quantified by four variables (Dudziak, 1999, p. 5): scope which describes what and how much will be done (functionality); resources (costs, developers, equipment); time which determines when the system (release) should be done; and quality which refers to the correctness of the system (as defined by the customer) and how well tested.

The third main activity is PM 3. Iteration Planning (N+1). At the beginning of each iteration a meeting is needed. An iteration has a fixed duration usually of two weeks. The activity has two INPUTS: "PM Release Plan" and "PM Minor Customer Change (N+1)." The iteration ends with the elaboration of the planned software product release. In this activity some "PM User Stories (Requirements)" are chosen to be developed. The "XP Customer" selects specific "PM User Stories (Requirements)" and the "XP Team" estimates the speed of the iteration. This PM 3. Iteration Planning (N+1) activity generates the OUTPUT "PM Plan for Iteration (N+1)," and this item is an INPUT of the overall process of SI Software Implementation. Later, in this Software Implementation process is elaborated the "SI Integration (N+1)" as main OUTPUT. This same output is an input feedback for next PM 3. Iteration Planning in next cycle. In Software Implementation process the "SI Integration (N+1)" includes the specific new implemented "PM User Stories (Requirements)" in this current iteration plus the previous one generated in past iterations. Finally, the PM 3. Iteration Planning activity uses the overall CONTROLs of "Time Schedule," "Quality Control" and "Vision," and the overall MECHANISMs of "XP Customer," "XP Team" and "XP Coach." Next overall process is Software Implementation (Iteration N+1). In SI process its main OUTPUT is the "SI Integration (N+1)." Its main INPUTS are: "PM Plan for Iteration N" and the "SI Integration (N+1)." Its CONTROLS are the overall "Time Schedule," "Quality Control" and "Vision," and its MECHANISMs involved are the "XP Team" and "XP Coach."

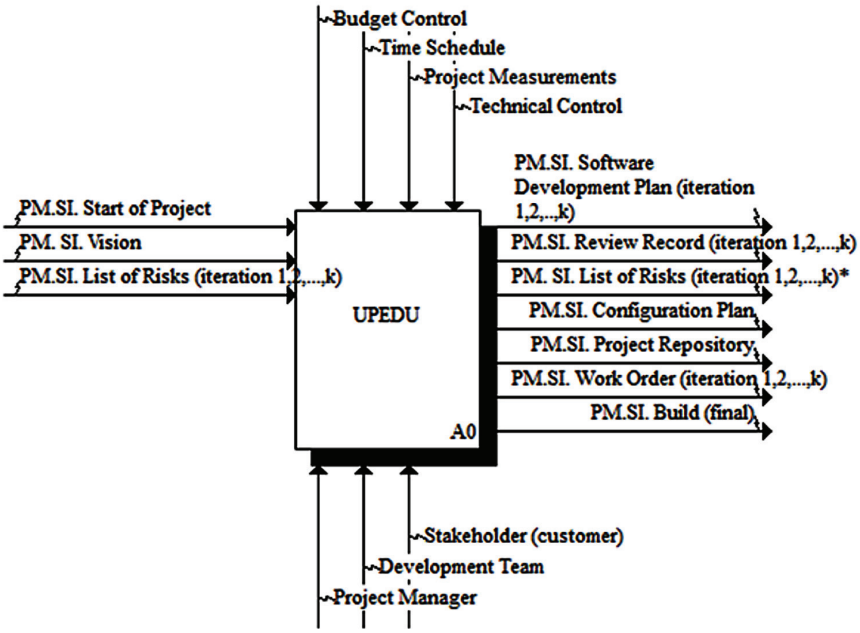
Finally, the PM 4. Customer Test (N+1) activity is the last main one in XP methodology. Each acceptance test ("PM Customer Approval") represents an expected result from the system which is authorized by "XP Customer." The INPUTs are: "PM Test Scenarios" and "SI Integration (N+1)." The "XP Customer" is who verifies and accept the different tests. A critical factor in this activity is the Quality Assurance (QA). In some cases, the QA is done for an external group but in other situations QA is done for the development team. Others roles involved are: "XP Team" and "XP Coach." The purpose of the activity is to guarantee the product according with the customer's requirement and that it will conclude with a project acceptable. We can find three CONTROLS in this activity:" Quality Control," "Time Schedule" and "Vision."

The results (OUTPUTs) of this last activity are very important for XP methodology because there are divided in different aspects as: 1) "PM.SI. Delivery Version N" which involves 1 or N iterations of the product or project; 2) "PM.SI. Customer Approval" where the customer gives the acceptance on the version N; and 3) "PM Minor Customer Change (N+1)" where the customer performs an evaluation of the advance of the product (iteration) and if something is wrong, the "XP Coach" and "XP Team" need to plan the corrections for the problem or lack that it was detected. It must be noted that this item is INPUT in for the PM 3. Iteration Planning (N+1) activity.

### The UPEDU Agile Methodology

The figure 9 reports the IDEF0 diagram of UPEDU. The PM process in UPEDU has three INPUTs: "PM.SI. Start of Project," "PM.SI. List of Risks (iteration 1,2,…, k)" and "PM.SI. Vision" and Its main OUTPUTs are: "PM.SI. Software Development Plan (iteration 1,2,…, k)," "PM.SI. Review Record (iteration 1,2,…, k)," "PM.SI. List of Risks (iteration 1,2,…, k)," "PM.SI. Configuration

**Figure 9. Top Level IDEF0 Diagram for UPEDU**

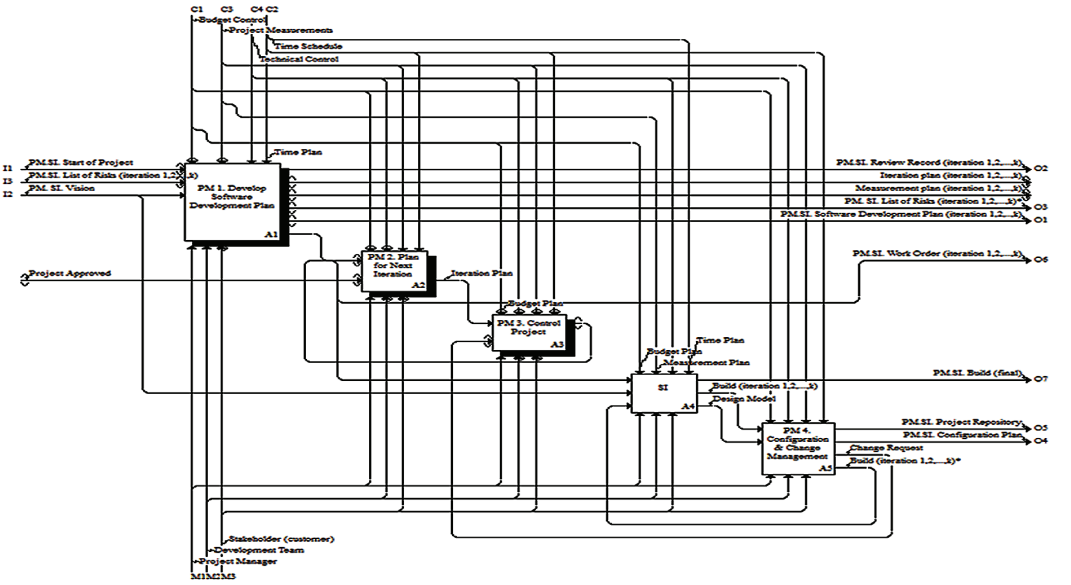



Plan," "PM.SI. Project Repository," "PM.SI. Work Order (iteration 1,2,…, k)" and "PM.SI. Build (final)." Its main MECHANISMs are: "Project Manager," "Development Team" and "Stakeholder." A "Project Manager" allocates resources, gives priorities, coordinates iterations between customers and users and provides a solid focus on the project goal. We include in this role the additional roles of "Change Control Manager" and "Configuration Manager." A "Development Team" is role which groups to the following UPEDU roles: "Analyst," "Designer," "Implementer," "Integrator," "Tester," and "Reviewer." A "Stakeholder (customer)" is a role defined in UPEDU as anyone who is affected by the project result. In this research, we consider it as Customer and User. Finally, the PM of UPEDU has four main CONTROLs to manage the correct function of Project Management stage: "Budget Control," "Time Schedule," "Project Measurements" (which can be used for controlling and measuring the overall quality of the process and the product) and "Technical Control."

The next Figure 10 describes graphically the next level of activities of UPEDU by using the IDEF0 diagram. The PM approach of UPEDU contains four main activities: PM 1. Develop Software Development Plan, PM 2. Plan for Next Iteration, PM 3. Control Project, Software Implementation and PM 4. Configuration and Change Management.

PM 1. Develop Software Development Plan is the first activity in UPEDU. We have identified that the Project Management process starts in this agile method by the initial external step of Conceive New Project producing "PM. SI. Start of Project" as an INPUT. The goal is simple and this is "obtain enough funding to proceed with a serious scoping and planning exercise" (UPEDU, 2016). The other two INPUTs in the first activity are: "PM. SI. Vision" and "PM.SI. Risk List (iteration 1,2,…, k)." The "PM.SI. Vision" gives to the stakeholders a view (key needs and features) about the product to be developed. In turn, "PM.SI. List of Risks (iteration 1,2,…, k)" artifact contains all the identified risks in the project, as well as problems and contingencies in the project. Additionally, some CONTROLs are very critical in this step as "Budget Control," "Project Measurements," "Time Schedule" and "Technical Control." Moreover, the first activity has three roles involved: "Project Manager," "Development Team" and "Stakeholder." This first activity has six OUTPUTs: "PM. SI. Review Record (iteration 1,2,…, k)," "Iteration Plan (iteration 1,2,…, k)," "Measurement Plan

**Figure 10. Main Activities IDEF0 Diagram of UPEDU**




(iteration 1,2,…, k)," "PM. SI. List of Risks (iteration 1,2,…, k)," "PM.SI. Software Development Plan (iteration 1,2,…, k)" and "PM SI. Work Order (iteration 1,2,…, k)." This activity will be performed during each start of iterations during the overall duration of the project. This activity includes three tasks: 1) Develop Measurement Plan; 2) Plan Phases and Iterations; and 3) Review Project Planning. In task 1) Develop Measurement Plan, the measures and metrics to be used to verify if the project is on good way are defined. In task 2) Plan Phases and Iterations, the effort is focused on the estimation of the work proposed for the project. Finally, on task 3) Review Project Planning, it is scheduled a review of checkpoints of previous activities. It includes also review meetings with all of the stakeholders.

The second main activity is PM 2. Plan for Next Iteration. Each iteration implies an agreed commitment to release a certain amount of work (of existing artifacts) with the agreed quality product by building intermediate products. This activity has an initial INPUT "Project Approved" and a feedback INPUT "Work Order (iteration 1, 2,…, k)" from the PM 3. Control Project activity. Also, the present activity has one OUTPUT: "Iteration Plan." In MECHANISMs this activity has all of the defined ones in the method (three) and it also contains all of the CONTROLs (four general controls). The quality of the final product is expected to be increased when changes are conducted in early iterations during the project. These changes are also expected to be of low costs. This activity contains a main task: Develop Iteration Plan. This task uses the regular INPUTs more the CONTROL "Technical Control." In particular engineering based items (like "Software Architecture Document" and "Vision") are also used for elaborating the next iteration plan. The "Software Architecture Document" artifact provides an overview of the system with different architectural aspects of the system. In the "Vision" document is defined the stakeholders' view of the project to be developed (requirements, contractual details and technical aspects). Finally, an "Iteration Plan" OUTPUT is elaborated.

The third main activity is PM 3. Control Project. This activity uses two INPUTs "Iteration Plan" and "Change Request." It generates a single OUTPUT called "PM.SI. Work Order (iteration 1,2,…, k)," which is an INPUT feedback for the PM 2. Plan for Next Iteration. In general, the PM 3. Control Project activity involves an adequate control of "Iteration Plan" and "Change Request," along the whole project. This activity has two main purposes: 1) dealing with change requests that have been sanctioned by the Change Control Manager, and scheduling these for the current or future iterations;

and 2) continuously monitoring the project in terms of active risks and objective measurements of progress and quality. This PM 3. Control Project activity has the next sub-activity: Schedule and Assign Work. The purpose of this sub-activity is to evaluate changes, defects, enhancements of products and process through the iterations. This PM 3. activity has the general CONTROLS defined at the first level according with the IDEF0 diagram of UPEDU. All of the MECHANISMs are also present in this activity. This activity tries to ensure the best control in project development.

SI process is reported as an overall process in the IDEF0 of Figure 10. This process has three main INPUTs: "PM.SI. Work Order (iteration 1,2,…, k)," "PM.SI. Vision" and "Build (iteration 1,2,…, k)." Its main OUTPUTs are: "PM.SI. Build (final)," "Build (iteration 1, 2,…, k)," and "Design Model." "PM.SI. Build (final)" defines the project delivery. It considers that the project has been finished of good way. "Build (iteration 1, 2,…, k)" defines the increment through the evolution of the project. "Design Model" defines the source code into the project in its different phases.

The main goal of SI process is developing and integrating every partial build of software developed by the "Development Team" for releasing next "Build (iteration 1, 2,…, k)" artifact. When the final release passes the final customer tests, a "PM.SI. Build (final)" is released. A key artifact is the increment generated and integrated as next "Build (iteration 1, 2,…, k)") artifact. This process involves the four general CONTROLs and all of the MECHANISMs defined at the first level of the IDEF0 of UPEDU.

The last activity is PM 4. Configuration and Change Management. It has two INPUTs "Build (iteration 1, 2,…, k)" and "Design Model." The activity has four OUTPUTs: "PM.SI. Project Repository" storages the different versions of the project. It saves data and metadata of files and directories. "PM.SI. Configuration Plan" links the requisites and control issues of the project. The responsibilities, resource, tools and others issues have a strong impact with this output. "Change Request" helps to keep the correct way in the different stages of the project and manages all required and asked changes. "Build (iteration 1, 2,…, k)*" is the build with a new increment respect to "Build (iteration 1, 2,…, k)." Finally, this last activity has all of the CONTROLs and MECHANISMs defined in UPEDU method.

### *Related Work on Compliance Analysis*

Due to the novelty of the ISO/IEC 29110 release, there is few published research in main software engineering journals at present. Main scientific literature comes from conferences and chapter books. Moreover, Conboy (2009) indicated that while there are famous agile methods (XP, SCRUM, and others) few research studies have been reported, and available ones are anecdotic from practitioners or consultant sources rather academic ones. Hence, an extensive literature review was conducted for the period 2000-2016 on a set of relevant software engineering journals (The Journal of Systems and Software, Information and Software Technology, Journal of Management Information Systems, Cutter IT Journal, MIS Quarterly), international research top conferences (EuroSPI and INCOSE International Symposium and books published by two international prestigious editorials: Springer and IGI. No particular research was found on a compliance analysis of the ISO/IEC 29110 standard and some agile methodology. We found a set of 12 related work for this research on the compliance level of some software processes and other software process standards. By space limitations, we report a summary of main findings of these related works in the Table 1.

Hence, the analysis of these 12 related research papers suggest that the compliance level of software process standards or process frameworks versus specific software development methodologies is of interest for academy and practitioners. This compliance analysis is done mainly with the general aim to fill identified gaps in methodologies regarding the software process reported in the standards. In particular, we remark that a compliance objective analysis on agile methodologies and the ISO/IEC 29110 standard was not found.

**Table 1. Summary of related work**

| Related Research | Research Goals, Purposes or Questions | Key Research Findings |
|---|---|---|
| Manzoni et al. (2003) | To assess the compliance level of RUP with the CMM Framework at levels 2 and 3. | 1. RUP must be complemented for achieving CMM levels 2 and 3. 2. RUP covers adequately most CMM level 2 and 3 Key Process Areas but managerial ones are under supported. 3. Software Process can be assessed versus Process Frameworks or Standards to establish their level of compliance. |
| Theunissen et al. (2003) | To investigate the adaptability of agile methodologies to software development standards laid down by ISO. | 1. There are interest in organizations for using combined agile methodologies and software process standards and models. 2. There is little guidelines for doing this combination. 3. Agile methodologies rely in a principle of flexibility and tailoring, and thus, can be adapted to standards and models. 4. The utilization of tools for supporting the process is recommended. |
| Fitsilis (2008) | To compare Project Management processes from PMBOK with a Project Management processes in 3 agile software processes (SCRUM, XP, FDD), and determine gaps, differences, and discrepancies. | 1. PMBOK is organized in 5 process groups and 9 knowledge areas. 2. Agile Project Management methodologies cannot be considered complete from PMBOK prescriptions by omitted recommendations or lack of explicit ones. 3. Agile Project Management processes report weak or null Risk Management, Cost Management, and Procurement Management processes. |
| Diaz et al. (2009) | To map CMMI Level 2 to Scrum Practices. It analyzes and compares the compliance between SCRUM and CMMI level 2. | 1. Agile approaches are considered as alternative for software process improvement in small and medium sized organizations. 2. SCRUM processes can be considered a valid process under CMMI-DEV paradigm. 3. SCRUM provides criteria to identify a minimum set of good practices to achieve CMMI capability level 2. |
| Mora et al. (2010) | To assess the compliance level of three software development methodologies: RUP, MSF and SA&D regarding to CMMI-SW, CMMI-DEV and CMMI-SVC. | 1. The research was focused on three main CMMI schemes: SW, DEV and SVC. 2. The research found that RUP and MSF fit CMMI-SW and CMMI-DEV in levels 2 and 3, but only level 2 for CMMI-SVC, and oldest SA&D approach failed in all of the three CMMI schemes. |
| Irrazabal et al. (2011) | This research was focused on the study of the relationship between agile practices (specially SCRUM) and a process subset (5 of the 43 processes) from the ISO/IEC 12207. | 1. It was found adequacy of SCRUM agile methodology to the ISO/IEC 12207 standard. 2. SCRUM provides a high compliance with PM processes, but a partial one with the remainder processes. 3. SCRUM must be enhanced in the event its utilization for achieving a high ISO/IEC 12207 compliance. |
| Pasini et al. (2013) | The research proposed a process model oriented in VSEs. This model offers roles, documents and activities able to satisfy the standard ISO/IEC 29110 with the main goal the business can improve their development process. | 1. Q-SCRUM was a new process model for VSEs with the conceptual claim to improve the development process. Empirical test was not reported. 2. Q-SCRUM is an adaptation of SCRUM with the purpose to satisfy the requirements of the ISO/IEC 29110 as standard. 3. In Q-SCRUM, the SCRUM activities were merged with the process of PM and SI in the ISO/IEC 29110, and it generates a model able to satisfy the requirements of the standard. |
| Chagas et al. (2014) | To identify characteristics of agile project management in organizations using agile methods and maturity models. | 1. There are interest in combining agile methods with software process standards and models. 2. There are still knowledge gaps on how integrate them without loss of agility. 3. The utilization of tools for supporting the integration is recommended. |
| Dyba et al. (2014) | To provide a characterization and definition of agile project management based on extensive studies of industrial projects. | 1. Project Management classic approach cannot be applied to agile methodologies. 2. Agile methodologies are characterized by complexity and uncertainty issues. 3. Agile Project Management approach relies on: minimum critical specification, autonomous teams, redundancy, and feedback and learning. |
| Larrucea and Santamaria (2015) | To try to map ITMark and ISO/IEC 29110 (basic profile) through the next questions: RQ1: Can ITMark be used to assess the ISO/IEC 29110 basic profile? RQ2: What is the coverage degree of ITMark over ISO/IEC 29110 basic profile? RQ3: Based on these assessments, what is the tendency during initial assessments? | 1. The study shows that RQ1 is satisfied. It means, ITMark can be used to assess the ISO/IEC 29110 (basic profile). 2. The research shows that RQ2 offers a summary of experiences providing an average of the basic profile areas. Also, the study could be concluded that ITMark coverage degree is important or relevant. 3. Through RQ3 is possible to identify some trends while assessing VSEs. |
| Cruz et al. (2015) | A study based on hybrid methodology (DC-DEP-FM) which is based on SCRUM and ISO/IEC 29110 to guide small teams of development of software in very strong changing environments. | 1. This hybrid methodology tries to solve some necessities of documentation, maintenance and software development. 2. This new methodology work with small teams and all the users (include customers) are in the same physical place of work. 3. The hybrid method are based in ISO/IEC 29110, however, the management activities, requirements, tests and releases are based on SCRUM and HU (traditional technique based on User Stories) adapted. |
| Garcia et al. (2016) | The development a metamodel to facilitate the definition of a Project Management process in small-sized software enterprises. | 1. It analyzed the effort to produce a quality product of software has two intertwined perspectives: Software Development and Project Management. 2. Software Development involves: activities, resource, skills and people to build a quality product. 3. Traditional methodologies provide guidelines about the steps and the order to manage the project to obtain some product. 4. The research supports evidences on the need to count with a simplified but still useful PM process in VSEs for helping in their growth and competitiveness. |

## The Objective Compliance Analysis

This thorough and objective compliance analysis advances on an initial and high-level analysis reported by authors (Galvan et al., 2015). In this new analysis was conducted in three steps. For space limitation, we report in overall the steps 1 and 2, and with detail the step 3.

**Analysis Step 1:** The first and second authors reviewed carefully the official ISO/IEC 29110 standard Entry Profile document (ISO, 2012) for generating the set of initial tables for analysis on Roles, Activities-Tasks and Work Products.

**Analysis Step 2:** The first and second authors reviewed carefully the selected materials for SCRUM, XP and UPEDU agile methodologies for generating a set of initial tables for analysis on Roles, Activities-Tasks and Work Products.

**Analysis Step 3:** The first and second authors used integrated tables from steps 1 and 2 for assessment objectively each Role, Activity-Task and Work Product reported in the SCRUM, XP and UPEDU agile methodologies regarding their fulfillment with the expected Role, Activity-Task and Work Product of the ISO/IEC 29110 standard Entry Profile. All generated tables are reported in the Appendix. The procedures used were as follows.

**Analysis Step 3.1:** For each Role, Activity-Task, and Work Product reported in the agile methodologies, we assessed the fulfillment of the essential characteristics in these agile methodologies vs the essential characteristics expected in the ISO/IEC 29110 standard Entry Profile. We used a four-scale of values (null ×, low Θ, moderate ◒ or high fulfillment ●). All assessments done to each essential characteristic of the ISO/IEC 29119 standard Entry Profile was summarized in an overall assessment with the same scale of values (null ×, low Θ, moderate ◒ or high fulfillment ●) under the following rules: 1) to assess a summarized high level ● for the specific column of essential characteristic to be assessed of the ISO/IEC 29110 (for instance "Coordinate Project" in the Role element) if it was assessed in a high level at least one characteristics of the set of essential characteristics of the agile methodologies ("Lead and coach the project" for SCRUM Role); 2) to assess a summarized moderate level ◒ when exists at least the half of moderate levels and none high level assessments (for instance "Manage project resources" for SCRUM Role); 3) to assess a summarized low level Θ when none high level and less than the half of moderate level were assessed (for instance "List of delivered elements" for the SCRUM Artifact of "Acceptance record"); and 4) to assess a summarized level of null × for all the remainder cases (for instance "Formalize project closure" for the SCRUM Activity of "Project closure" for SCRUM).

**Analysis Step 3.2:** For each element (Role, Activity-Task or Work Product) of the ISO/IEC 29110 standard Entry Profile we calculated its overall score with the average of the sum of assigned scores (0 for null, 1 for low, 2 for moderate, and 3 for high). This average score was mapped to the similar qualitative scale. For instance, for Customer Role in SCRUM the assessments were high, high and null in SCRUM. Then its average score of 2.0 corresponds to a moderate level.

**Analysis Step 3.3:** A final numeric score was calculated with a weighted sum of each expected element (Role, Activity-Task or Work Product) of the ISO/IEC 29110 standard Entry Profile. Because none reference in the literature was found on a distribution of weights, the research team agreed to assign own weight distribution based on the following rationale: 1) for the three Roles (Customer, Project Manager, and Work Team) the assigned weights were equally distributed (33.3%, 33.3% and 33.3%) under the assumption of each role has the same rights and power to cause a successful or failed project respectively; 2) for the four Activities-Tasks (Project Planning, Project Plan Execution, Project Assessment and Control, and Project Closure) the assigned weights were assigned under the evidence of tasks and work products generated in each activity (i.e. 30%, 30%, 30% and 10% respectively); and 3) for the eight Work Products (Statement of Work, Project Plan, Project Repository, Meeting Record, Progress Status Record, Change Request, Acceptance Record, and Software Configuration) the assigned weights were based on the extent of difficulty to be generated (i.e. 10%, 20%, 20%, 10%, 10%, 10%, 10%, and 20% respectively).

In this last case, the Project Plan, Project Repository and Software Configuration work products are considered the most demanded in effort for their generation. In contrast, the remainder five work products were considered mandatory but with a straightforward elaboration. Their weights are, however, relatively high (10%) because are considered as the set of all individual artifacts generated during the ASDM.

**Analysis Step 3.4:** We arrived a final total assessment with the previous numeric scores calculated for Roles, Activities-Tasks and Work Products categories for each agile methodology. This final score was calculated as the weighted sum of scores obtained for Roles, Activities-Tasks and Work Products categories. The agreed weights were 33.3%, 33.3% and 33.3% respectively. The final total scores were in the range of 0 to 100 points. This final and total numeric score was also mapped to a final qualitative scale of five values as follows: very low for the range of 0-29 points, low for the range of 30-59 points, moderate for the range of 60-79 points, high for the range of 80-89 points, and very high in the range of 90-100 points. This scale is ordinal and thus the interval sizes are different. This final assessment was reviewed by the third author, an official co-writer of the ISO/IEC 29110 standard. None discrepancy was found. The Figure 11 reports the final and total scores assessed for the three agile methodologies.

### *Process of Agile Methodologies with the ISO/IEC 29110 Standard Entry Profile*

The figure 11 reports the overall scores obtained before to be weighted (i.e. 33.3% for Roles, 33.3% for Activities-Tasks, and 33.3% for Work Products). This section of the figure reports the results in the range of 0-100 points. The used colors red, orange, yellow, light green and green corresponds to the very low, low, moderate, high and very high compliance level respectively. For instance, the SCRUM compliance level for Roles element with 78 points correspond to the range of 60-79 points, and thus uses the yellow color. This mapping is also reported in the third part of the figure. The middle part of the figure reports the weighted scores. For instance, the SCRUM score for Roles was 78 points before to be weighted and after its weight, it resulted in a final score of 26 points. Finally, the third part of the figure reports the final total weighted scores mapped to the qualitative scale previously reported. For instance, SCRUM compliance levels for Roles, Activities-Tasks, and Work Products were respectively of moderate, very high and moderate values, and its final total assessment of 79 points corresponded to moderate level.

**Figure 11. Summary of Final Total Assessment on Compliance of Project Management**

| | | SCRUM | XP | UPEDU |
|---|---|---|---|---|
| *ROLES* | 33% | 0.78 | 0.55 | 0.88 |
| *ACTIVITIES-TASKS* | 34% | 0.90 | 0.37 | 0.97 |
| *WORK PRODUCTS* | 33% | 0.70 | 0.60 | 0.93 |
| | TOTAL | 0.79 | 0.51 | 0.93 |
| | | | | |
| | | SCRUM | XP | UPEDU |
| *ROLES* | 33.3% | 0.26 | 0.18 | 0.29 |
| *ACTIVITIES-TASKS* | 33.3% | 0.30 | 0.12 | 0.32 |
| *WORK PRODUCTS* | 33.3% | 0.23 | 0.20 | 0.31 |
| | TOTAL | 0.79 | 0.51 | 0.93 |
| | | | | |
| | | SCRUM | XP | UPEDU |
| *ROLES* | 33.3% | MODERATE | LOW | HIGH |
| *ACTIVITIES-TASKS* | 33.3% | VERY HIGH | LOW | VERY HIGH |
| *WORK PRODUCTS* | 33.3% | MODERATE | MODERATE | VERY HIGH |
| | TOTAL | MODERATE | LOW | VERY HIGH |

## Discussion of Results, Implications, Limitations, and Recommendations

### *Results*

This research pursued two core questions: 1) RQ.1 What is the objective compliance level to the Project Management process defined in the ISO/IEC 29110 standard Entry Profile from the Project Management process explicitly reported in SCRUM, XP and UPEDU agile methodologies? and RQ.2 What are the theoretical and practical implications required for improving the compliance levels (if required)?

RQ.1 is objectively answered with the results reported in Figure 11. As it was expected, the most well-structured methodology (i.e. UPEDU) as a sub-set of the RUP rigor-oriented methodology (Kruchten, 2003) provided an overall very high compliance (93 of 100 points) on Roles, Activities-Tasks, and Artifacts to the ISO/IEC 29110 standard Entry Profile. In contrast, the agile methodologies achieved less compliance levels. The SCRUM and XP methodologies achieved an overall moderate (79 of 100 points) and low (56 of 100 points) compliance levels respectively. SCRUM is strongly focused on Project Management process with minimal or null explicit indications for the Software Implementation activities. In contrast, XP is strongly focused on Software Implementation activities and avoids or minimizes Project Management ones.

For UPEDU, its overall very high compliance level was conformed by a high compliance level by Roles, and very high compliance levels by Activities-Tasks, and Artifacts. UPEDU supports very satisfactory the eight of the nine essential functions identified by the three Roles of the ISO/IEC 29110 standard. For this Roles element, its unique weakness is the non-implicit consideration on "Formalize project start and closure" tasks in the Customer Role. Regarding the Activities-Tasks element, UPEDU covers very satisfactory the four Activities (A1. Project Planning, A2. Project Plan Execution, A3. Project Assessment and Control, and A4. Project Closure) and seventeen of the eighteen Tasks proposed in the ISO/IEC 29110 standard. The unique omitted Task (i.e. not explicitly reported) in UPEDU is A4.T1 Formalize Project Closure, in the fourth Activity of A4. Project Closure. Similarly, to Activities-Tasks element, UPEDU covers very satisfactory the Work Products element. UPEDU uses clearly seven of the eight Work Products. The unique weakness refers to the Software Configuration Work Product, which does not report explicitly the "user documentation" and the "technical documentation" items.

For SCRUM, its overall moderate compliance level was conformed by moderate compliance levels by Roles and Work Products, and a very high compliance level for Activities-Tasks. SCRUM supports satisfactory five of nine essential functions identified by the three Roles of the ISO/IEC 29110 standard. Each Role omits expected functions of the ISO/IEC 29110 standard. "Product Owner" Role does not consider its explicit function of "formalize project start and closure." "SCRUM Master" Role does not perform explicitly the function of "manage project resources." Finally, "Work Team" Role does not include explicitly the function of "produce technical documentation." Regarding the Activities-Tasks element, SCRUM covers satisfactory three (A1. Project Planning, A2. Project Plan Execution, A3. Project Assessment and Control) of the four Activities proposed, and twelve of the eighteen Tasks proposed in the ISO/IEC 29110 standard. SCRUM's weaknesses in the A1. Project Planning Activity refers to the omission of explicit practices for A1.T11 Establish the project repository. The other SCRUM's weaknesses are in the A4. Project Closure Activity, which omits descriptions on the two, expected Tasks: A4.T1 Formalize Project Closure and A4.T2 Update Project Repository. Similar to Roles element, SCRUM covers moderately the Work Products element. SCRUM covers satisfactory six of the eight Work Products expected for the ISO/IEC 29110 standard. The two unsatisfactory Work Products are: "Change Request" and "Meeting Record." SCRUM has omissions on expected items for the ISO/IEC 29110 standard. Some of them are: "project identification data and signatures," "cost of change," "impact, criticality and status of change" and "user documentation."

For XP, its overall low compliance level was conformed by low compliance levels by Roles and Activities-Tasks, and a moderate compliance level for Work Products. XP supports satisfactory six

of nine essential functions identified by the three Roles of the ISO/IEC 29110 standard. The weakest XP Role is "XP Coach." The expected functions for the ISO/IEC 29110 standard are: "coordinate project," "manage project resources," and "delivery an agreed software product." These functions are not explicitly reported in XP. The other two XP Roles also omit essential functions. These are: "formalize project start and closure" for "XP Customer" Role, and "produce technical documentation" for "Work Team" Role. Regarding the Activities-Tasks element, XP covers moderately three (A1. Project Planning, A2. Project Plan Execution, and A4. Project Closure) of the four Activities proposed, and eleven of the eighteen Tasks proposed in the ISO/IEC 29110 standard. The three Tasks of the A3. Project Assessment and Control Activity are not explicitly considered in XP. These Tasks are: A3.T1 Assess Project Plan vs Real Advance, A3.T2 Track Changes, and A3.T3 Establish Corrective Actions. XP covers satisfactory or moderately five of the eight Work Products expected for the ISO/IEC 29110 standard. These five Work Products are: "Statement of Work," "Acceptance Record," "Project Plan," "Project Repository," and "Software Configuration." In contrast, three Work Products are weakly considered in XP. These are: "Change Request," "Meeting Record," and "Progress Status Record."

### *Implications for Theory and Praxis*

RQ.2 can be answered with the next discussions on implications for theory and praxis. The previous results lead to several implications for update or challenge extant theory (or at least research findings from the current literature), as well as for proposing practical recommendations for agile software practitioners and ISO/IEC 29110 trainers.

From a theoretical perspective we identified the following implications. First, there are literatures on benefits obtained for using software process models and standards (Goldenson & Gibson, 2003; Pino et al., 2008; Clarke and O'Connor 2012; Laporte et al., 2013) and for using agile methods (Dyba & Dingsøyr, 2008; Chow and Cao, 2008; Bustard et al., 2013), and thus their integration might be not required. However, other literature has found interest of organizations for taking advantages of agile methods with the compliance of standards and models for marketing and contractual requirements (Heeager and Nielsen, 2009; Chagas et al., 2014). This latter literature reports, however, difficulties of integration. Thus, the first theoretical implication is the need of proposes standards that include from their conception agile principles. Second, the classic Project Management approaches reported in all software process standards and models (Dyba et al., 2008) are based on a strict planning, organizing, monitoring-control, and formal finalization of projects. In this classic approach the Project Manager has a clear authority on the Development Team. In contrast, the agile methodologies rely in self-autonomy, self-organization and a lax control. There is not strictly a Project Manager role but a Coach. Thus, the second theoretical implication is the need to establish a new theory of concepts, principles and interrelationships on agile Project Management. Third, there are agile methodologies derived from rigor-oriented ones (i.e. UPEDU in particular) where their authors have proposed a different concept of agility as process adaptability (Krutchen, 2001; Germain and Robillard, 2005). UPEDU, classified as an agile methodology was assessed with a very high compliance with a software process standard in this study. Thus, other theoretical path is to extend the concept of agility to embrace documented-based methodologies like UPEDU as a legitimate agile methodology. Fourth, according to several comprehensive surveys on agile methodologies (Abrahamsson et al., 2010; Dyba and Dingsoyr, 2008; Dyba et al. (2014)), they vary in their coverage on Project Management and Software Implementation activities. In particular, Dyba et al. (2014) reports that SCRUM methodology does not include any Software Implementation process, and XP does not provide a clear Project Management process. Our research was targeted on Project Management process, and thus given the differences of coverage by the three agile methodologies on this process, the theoretical value and need of Project Management process on agile project must be strengthened.

From a practical perspective we identified the following implications. First, there is a scarcity of published guidelines for using SCRUM or XP agile methodologies combined with an appropriate software process standard like the ISO/IEC 29110. Second, the ISO/IEC 29110 design international

group has not published also specific guidelines for SCRUM or XP practitioners. Third, given the natural interest on agile methodologies for VSEs, the target market for the ISO/IEC 29110 standards, emerges a practical strong need of elaborating and testing the usability of Deployment Packages (released as Electronic Process Guides) of enhanced SCRUM or XP methodologies (Mora et a, 2016).

### *Methodological Limitations*

Like any interpretative study, there are several sources of potential variation on results. These are the following ones. First, one of the three core researchers is an Editor of ISO/IEC 29110-2-1:2015 standard and an uncontrolled bias on the expected rigor for compliance of elements in the agile methodologies might be present. Second, we assign our own structures of weights for determining the compliance levels for the Roles, Activities-Tasks and Work Artifacts. Other research team changing these weights might arrive to different assessments. Third, we fixed a limit of iterations (3) and framing time (6 weeks for methodology) for achieving the evaluations. Other research team by using less or more iterations and framing time for analysis might derive different assessments. Fourth, all evaluations were realized on the selected material. No one real documented project using the ISO/IEC 29110 standard or some of the three agile methodologies was used. Other research team by using these kinds of materials might arrive to different results. Finally, the inclusion of new published materials on these three agile methodologies for analysis might generate variations in our evaluations.

Nevertheless, these inherent methodological limitations, we consider that a similar research team can derive similar evaluation results by using the same set of materials, the IDEF0 structures of the ISO/IEC 29110 standard, and their proposed Roles, Activities-Tasks, and Work Artifacts, as well the systematic procedures and assigned scales.

### *Recommendations for Further Research*

The results, implications and limitations of this research permit us to propose several recommendations for further research. These are the following ones. First, we suggest strongly conducting theoretical replications by using the same set of materials by other similar research teams for arriving more conclusive findings. Second, we suggest conducting applied replications by using the same set of materials plus documents of real projects performed with the ISO/IEC 29110 standard and the three agile methodologies (if available this documentation) or by including in the evaluators to expert practitioners in the agile methodologies. Third, we suggest to generate enhanced versions of the SCRUM and XP agile methodologies in order they achieve similar compliance levels to the reached by UPEDU methodology and test their suitability for a community of practitioners. The risk of enhance these methodologies is a rejection by part of practitioners if these methodologies are not finally perceived as agile ones. Fourth, we suggest elaborating Deployment Packages (implemented like Electronic Process Guides) for these enhanced SCRUM and XP methodologies and evaluate usability metrics such as usefulness, ease of use, compatibility, value and intention of usage. Finally, we suggest further empirical research on the implementation of the ISO/IEC 29110 standard combined with the utilization of the UPEDU.

These recommendations are considered for advancing on this specific research stream.

## CONCLUSION

This study was conducted for evaluating objectively the compliance levels on Roles, Activities-Tasks, and Work Products proposed and expected by the ISO/IEC 29110 standard Entry Profile for the SCRUM, XP and UPEDU agile methodologies. We targeted our study only for the Project Management process by delimiting an adequate scope. A further similar analysis will be realized for the other ISO/IEC 29110 process (i.e. Software Implementation).

For this aim, we used IDEF0 like main conceptual tool and for generating specific detailed tables of analysis (these are available upon request to first author). The systematic procedure for evaluating

was conducted by first and second authors, with a careful review for the third author. We obtained the following results: 1) UPEDU achieved an overall very high compliance level; 2) SCRUM achieved an overall moderate compliance level; and 3) XP achieved an overall low compliance level. These results are congruent with the expectations from the nature of each methodology. UPEDU is an agile methodology derived from a rigor-oriented well-known methodology: RUP, and its concept of agility is based on adaptability. SCRUM is an agile methodology focused on Project Management which omits the Software Implementation process. XP is an agile methodology focused on engineering activities and its Project Management activities are unclear.

Several theoretical and practical implications were reported. Essentially a new theory of agile methodologies and agile standards is required, as well as practical guidelines reporting the integrations of some of these agile methodologies with some appropriate standard. Methodological limitations were also reported, but we consider that other similar research team by using the same set of materials and tools, will reach similar results.

Finally, we reported several recommendations for further research. Given the huge economic value of the software industry located in VSEs, the relevance gained for software process standards and models, and the popularity of using agile methodologies, further research is highly encouraged.

## REFERENCES

Abrahamsson, P., Babar, M. A., & Kruchten, P. (2010). Agility and architecture: Can they coexist? *Software, IEEE*, *27*(2), 16–22. doi:10.1109/MS.2010.36

Basri, S., & O'Connor, R. (2010). Understanding the Perception of Very Small Software Companies towards the Adoption of Process Standards. In Riel et al. (Eds.), *Systems, Software and Services Process Improvement, CCIS* (Vol. 99, pp. 153–164). Springer-Verlag. doi:10.1007/978-3-642-15666-3_14

Basri, S., & O'Connor, R. (2011). Towards an Understanding of Software Development Process Knowledge in Very Small Companies. In A. Manaf et al. (Eds.), *Informatics Engineering and Information Science, CCIS* (Vol. 253, pp. 62–71). Springer-Verlag. doi:10.1007/978-3-642-25462-8_6

Beck, K. (1999). Embracing Change with Extreme Programming. IEEE Computer, October, 70-77.

Blankenship, J., Bussa, M., & Millett, S. (2011). Managing agile projects with scrum. In Pro Agile. NET Development with Scrum (pp. 13-27). Apress. doi:10.1007/978-1-4302-3534-7_2

Bustard, D., Wilkie, G., & Greer, D. (2013). Towards optimal software engineering: Learning from agile practice. *Innovations in Systems and Software Engineering*, *9*(3), 191–200. doi:10.1007/s11334-013-0216-9

Chagas, L. F., de Carvalho, D. D., Lima, A. M., & Reis, C. A. L. (2014). Systematic literature review on the characteristics of Agile project management in the context of maturity models. In *Software Process Improvement and Capability Determination* (pp. 177–189). Springer International Publishing. doi:10.1007/978-3-319-13036-1_16

Chow, T., & Cao, D.-B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, *81*(6), 961–971. doi:10.1016/j.jss.2007.08.020

Clarke, P., & O'Connor, R. (2012). The situational factors that affect the software development process: Towards a comprehensive reference framework. *Information and Software Technology*, *54*(5), 433–447. doi:10.1016/j.infsof.2011.12.003

Coleman, G., & O'Connor, R. (2008). Investigating Software Process in Practice: A Grounded Theory Perspective. *Journal of Systems and Software*, *81*(5), 772–784. doi:10.1016/j.jss.2007.07.027

Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research*, *20*(3), 329–354. doi:10.1287/isre.1090.0236

Cruz, E. (2015). Disciplinando Equipos Pequeños con Prácticas Ágiles. *Revista en Ingeniería y Tecnología*, 8(2).

Diaz, J., Garbajosa, J., & Calvo-Manzano, J. (2009). Mapping CMMI Level 2 to Scrum Practices: An Experience Report. In Software Process Improvement, CCIS (Vol. 42, pp. 93-104). Springer-Verlag.

Dudziak, T. (1999). eXtreme Programming An Overview. Methoden und Werkzeuge der Softwareproduktion.

Dyba, T., & Dingsoyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, *50*(9-10), 833–859. doi:10.1016/j.infsof.2008.01.006

Dybå, T., Dingsøyr, T., & Moe, N. B. (2014). Agile Project Management. In *Software Project Management in a Changing World* (pp. 277–300). Springer Berlin Heidelberg.

Fernandez-Monteiro. Paula. (2014). Tailoring CMMI-DEV and RUP frameworks for ML2/3-compliance analysis [PhD Dissertation]. Universidade do Minho Escola de Engenharia. Portugal

Fitsilis, P. (2008). Comparing PMBOK and Agile Project Management software development processes. In *Advances in Computer and Information Sciences and Engineering* (pp. 378–383). Springer Netherlands. doi:10.1007/978-1-4020-8741-7_68

Gallagher, B., & Brownsword, L. (2001). *The rational unified process and the capability maturity model–integrated systems/software engineering*. RUP/CMMI Tutorial–ESEPG.

Galvan, S., Mora, M., O'Connor, R. V., Acosta, F., & Alvarez, F. (2015). A Compliance Analysis of Agile Methodologies with the ISO/IEC 29110 Project Management Process. *Procedia Computer Science*, *64*, 188–195. doi:10.1016/j.procs.2015.08.480

Garcia, I., Pacheco, C., Arcilla, M., & Sanchez, N. (2016). Project Management in Small-Sized Software Enterprises: A Metamodeling-Based Approach. In Trends and Applications in Software Engineering (pp. 3-13). Springer International Publishing. doi:10.1007/978-3-319-26285-7_1

Germain, É., & Robillard, P. N. (2005). Engineering-based processes and agile methodologies for software development: A comparative case study. *Journal of Systems and Software*, *75*(1), 17–27. doi:10.1016/j.jss.2004.02.022

Glass, R. L., Vessey, I., & Ramesh, V. (2002). Research in software engineering: An analysis of the literature. *Information and Software Technology*, *44*(8), 491–506. doi:10.1016/S0950-5849(02)00049-6

Goldenson, D., & Gibson, D. L. (2003). Demonstrating the impact and benefits of CMMI: an update and preliminary results.

Heeager, L. T., & Nielsen, P. A. (2009). Agile software development and its compatibility with a document-driven approach? A case study. *Proceedings of the20th Australasian Conference on Information Systems Compatibility of Agile and Document-Driven Approaches*, Melbourne.

IDEF. (1993). IDEF Method Report - The Standard for Integration Definition for Function Modeling (IDEF0). Federal Information Processing Standards Publication 183. Retrieved from http://www.idef.com/pdf/idef0.pdf

Irrazabal, E., Vásquez, F., Díaz, R., & Garzás, J. (2011). Applying ISO/IEC 12207: 2008 with SCRUM and Agile Methods. In Software Process Improvement and Capability Determination (pp. 169-180). Springer Berlin Heidelberg.

ISO. (2012). *Software engineering-Lifecycle profiles for Very Small Entities (VSEs)-. ISO/IEC TR 29110-5-1-1*. ISO.

Kettinger, W. J., Teng, J. T., & Guha, S. (1997). Business process change: A study of methodologies, techniques, and tools. *Management Information Systems Quarterly*, *21*(1), 55–80. doi:10.2307/249742

Kruchten, P. (2001). Agility with the RUP. *Cutter IT Journal*, *14*(12), 27–33.

Lambert, J. H., Jennings, R. K., & Joshi, N. N. (2006). Integration of risk identification with business process models. *Systems Engineering*, *9*(3), 187–198. doi:10.1002/sys.20054

Laporte, C. Y., Alexandre, S., & O'Connor, R. (2008). A Software Engineering Lifecycle Standard for Very Small Enterprises, R.V. O'Connor et al. (Eds), Proceedings of EuroSPI (pp. 129-141). Springer-Verlag.

Laporte, C. Y., O'Connor, R., & Fanmuy, G. (2013). International Systems and Software Engineering Standards for Very Small Entities. Crosstalk, May/June, 28-33.

Larrucea, X., O'Connor, R. V., Colomo-Palacios, R., & Laporte, C. Y. (2016). Software process improvement in very small organizations. *IEEE Software*, *33*(2), 85–89. doi:10.1109/MS.2016.42

Larrucea, X., & Santamaria, I. (2015). ITMark as an ISO29110 profile: building an experience factory. Proceedings of EuroSPI '15, Ankara, Turkey. Springer.

Manzoni, L., & Price, R. (2003). *Identifying Extensions Required by RUP (Rational Unified Process) to Comply with CMM (Capability Maturity Model) Levels 2 and 3*. IEEECS.

Marca, D. A., & Perdue, B. A. (2000). A software engineering approach and tool set for developing Internet applications.*Proceedings of the 22nd international conference on Software engineering* (pp. 738-741). ACM Digital Library. doi:10.1145/337180.337619

Microsoft. (2008). Microsoft Solutions Framework (MSF) for CMMI® Process Improvement Guidance, Online document at www.microsoft.com

Mora, M., Gelman, O., O'Connor, R., Alvarez, F., & Macías-Luévano, J. (2009). An Overview of Models and Standards of Processes in the SE, SwE And IS Disciplines. In A. Cater-Steel (Ed.), Information Technology Governance and Service Management: Frameworks and Adaptations (pp. 371–387). Hershey, PA, USA: IGI Publishing; doi:10.4018/978-1-60566-008-0.ch021

Mora, M., Gelman, O., Paradice, D., & Cervantes, F. (2008, May 18-20). The case for Conceptual Research in Information Systems. *Proc. 2008 International Conference on Information Resources Management*, Niagara Falls, Ontario, Canada (pp. 1-10).

Mora, M., O'Connor, R., & Macías-Luévano, R. (2010). CMMI (SW, DEV, SVC) Compliance of SDLCs: A Service System View. *Proceedings of theInternational Conference on Information Resources Management(Conf-IRM)*, Jamaica.

Mora, M., O'Connor, R. V., Mahesh, R., & Gelman, O. (2016). Impacts of electronic process guides by types of user: An experimental study. *International Journal of Information Management*, *36*(1), 73–88. doi:10.1016/j.ijinfomgt.2015.10.001

O'Connor, R., & Laporte, C. (2010, June). Towards the Provision of Assistance for Very Small Entities in Deploying Software Lifecycle Standards. *Proceedings of the11th International Conference on Product Focused Software Development and Process Improvement (PROFES 2010)*. doi:10.1145/1961258.1961259

O'Connor, R. V., & Laporte, C. Y. (2012) Software Project Management with ISO/IEC 29110. *Proceedings of the 19th European Conference on Systems, Software and Services Process Improvement (EuroSPI 2012), CCIS* (Vol. 301). Springer-Verlag.

O'Connor, R. V., & Laporte, C. Y. (2014). An Innovative Approach to the Development of an International Software Process Lifecycle Standard for Very Small Entities. *International Journal of Information Technologies and Systems Approach*, *7*(1), 1–22. doi:10.4018/ijitsa.2014010101

Organization for Economic Co-Operation and Development (OECD). (2005). *SME and Entrepreneurship. (Outlook* ed.).

Pasini, A. C., Esponda, S., Boracchia, M., & Pesado, P. (2013). Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110. *Proceedings of theXVIII Congreso Argentino de Ciencias de la Computación*.

Pearman, G., & Goodwill, J. (2006). *Pro. NET 2.0 Extreme Programming*. Apress.

Pino, F. J., García, F., & Piattini, M. (2008). Software process improvement in small and medium software enterprises: A systematic review. *Software Quality Journal*, *16*(2), 237–261. doi:10.1007/s11219-007-9038-z

Presley, A., & Liles, D. H. (1995). *The Use of IDEF0 for the Design and Specification of Methodologies. Technical Report*. University of Texas at Arlington.

Robillard, P. N., Kruchten, P., & d'Astous, P. (2001). Yoopeedoo (UPEDU): a process for teaching software process. *Proceedings of the 14th Conference on Software Engineering Education and Training* (pp. 18-26). IEEE.

SCRUM-EPG. (2008). Mountain Goat Software.

SEI. (2006). CMMI for development, version 1.2: CMMI-DEV, V1.2. Software engineering institute. CMU/SEI-2006-TR-008, ESC-TR-2006-008. Retrieved from www.sei.edu

Stavru, S. (2014). A critical examination of recent industrial surveys on agile method usage. *Journal of Systems and Software*, *94*, 87–97. doi:10.1016/j.jss.2014.03.041

Succi, G., Valerio, A., Vernazza, T., & Succi, G. (1998). Compatibility, standards, and software production. *StandardView*, *6*(4), 140–146. doi:10.1145/338183.338185

Sutherland, J., and Schwaber, K. (2013). The scrum guide: the definitive guide to scrum: the rules of the game.

Tang, D., Zheng, L., Li, Z., Li, D., & Zhang, S. (2000). Re-engineering of the design process for concurrent engineering. *Computers & Industrial Engineering*, *38*(4), 479–491. doi:10.1016/S0360-8352(00)00059-0

Theunissen, W. H. M. (2003). A case-study based assessment of Agile software development [Doctoral dissertation]. University of Pretoria.

UPEDU. (2016). Retrieved from www.upedu.org

XP-EPG. (2006). IBM Corporation and Object Mentor.

*Sergio Galvan-Cruz is the Chief Executive Officer in TECOM since 2007 in the scope of software development for small business in different Mexican states. Master in Science since 2009. Professor in Electronic Systems in the Autonomous University of Aguascalientes (UAA), Mexico Currently PhD Student (since 2014) in University Juarez Autonomous of Tabasco (UJAT), Mexico.*

*Manuel Mora is a full-time Professor and Researcher in the IS Department in the Autonomous University of Aguascalientes (UAA), Mexico. He holds a BS in Computer Systems Engineering (1984) and MSc in Computer Sciences (Artificial Intelligence area, 1989) from Monterrey Tech (ITESM), and an EngD in Engineering (Systems Engineering area, 2003) from the National Autonomous University of Mexico (UNAM). He has published about 80 research papers in international top conferences, research books, and journals. He is a member of ACM (senior), of IEEE SMC Society, and of INCOSE, and serves in the ERB of several international journals.*

*Rory V. O'Connor is an Associate Professor of Software Engineering at Dublin City University (Ireland) where he is currently serving as the Head of the School of Computing. He is also a Senior Researcher with Lero, the Irish Software Research Centre. Prof. O'Connor is Ireland's Head of Delegation to the International Organization for Standardization (ISO) for Software & Systems Engineering (ISO/IEC JCT1/ SC7). His research interests are centered on the processes and standards whereby software intensive systems are designed, implemented and managed. His focus is on researching methods, techniques, tools and standards for supporting the work of software project managers and software developers in relation to software process improvement, and the management of software development projects. He can be reached at Rory.OConor@dcu.ie.*

*Francisco Acosta is a full professor and researcher in the Computer Science Division at the Juaréz Autonomous University of Tabasco (México). He holds a PhD in Computer Science from the University Montpellier II, France; within their current academic activities are the chair of a research seminar in PhD program in Computer Science and the coordination of a computer science PhD program. His current research interests include Ambient Intelligence (AmI), Ubiquitous Computing, Data Mining, Semantic Web and Software Engineering. He has published over 30 research papers in international top conferences, research books, and indexed journals.*

*Francisco Alvarez Rodríguez, is a Professor of Software Engineering. He holds a BA in Informatics (1994) and a MA (1997) from the Autonomous University of Aguascalientes and a EdD degree from the Education Institute of Tamaulipas, México and he is PhD from the National Autonomous University of Mexico. He has published research papers in several international conferences in the topics of software engineering and e-learning process. His research interests are software engineering lifecycles for small and medium sized enterprises and software engineering process for e-learning. He is currently president of the National Council for Accreditation of programs and Computing, A.C. (CONAIC)*

## APPENDIX

**Figure 12. SCRUM Assessment for Roles**

| SCRUM METHODOLOGY | Customer | | | Project Manager | | | Work Team | | |
|---|---|---|---|---|---|---|---|---|---|
| | Provide software requirements | Authorize and review project | Formalize project start and closure | Coordinate project | Manage project resources | Delivery an agreed software | Build the software product | Fit quality expectations | Produce technical documentation |
| **Product Owner** | | | | | | | | | |
| Represents Customer's interests | ● | × | × | | | | | | |
| Authorize and review project outcomes | × | ● | × | | | | | | |
| Accept or reject final product | × | × | ⊙ | | | | | | |
| **SCRUM Master** | | | | | | | | | |
| Lead and coach the project | | | | ● | ◒ | ● | | | |
| Enforce SCRUM approach | | | | ◒ | ◒ | ◒ | | | |
| Interface between Product Owner and SCRUM team | | | | ◒ | ◒ | ◒ | | | |
| **SCRUM Delivery Team** | | | | | | | | | |
| Choose the best way to accomplish their work | | | | | | | ● | × | × |
| Deliver product iteratively and incrementally | | | | | | | ● | ◒ | × |
| Fit expected quality from Product Owner and SCRUM Master | | | | | | | × | ◒ | × |
| LEVEL OF COMPLIANCE BY ITEM | HIGH | HIGH | NULL | HIGH | MOD | HIGH | HIGH | MOD | NULL |
| LEVEL OF COMPLIANCE BY METHODOLOGY | MOD | | | HIGH | | | MOD | | |
| % OF IMPORTANCE | 33% | | | 33% | | | 33% | | |
| SCORE | 2 | | | 3 | | | 2 | | |
| POINTS | 0.67 | | | 1.00 | | | 0.67 | | |
| LEVEL OF COMPLIANCE | 0.78 | | | | | | | | |

**Figure 13. XP Assessment for Roles**

| | Customer | | | Project Manager | | | Work Team | | |
|---|---|---|---|---|---|---|---|---|---|
| **XP METHODOLOGY** | Provide software requirements | Authorize and review project outcomes | Formalize project start and closure | Coordinate project | Manage project resources | Delivery an agreed software product | Build the software product | Fit quality expectations from Customer | Produce technical documentation |
| **XP Customer** | | | | | | | | | |
| Define requirements (User Stories) | ● | × | × | | | | | | |
| Authorize and review project outcomes | × | ● | × | | | | | | |
| Accept or reject final product | × | × | ⊙ | | | | | | |
| **XP Coach and Tracker** | | | | | | | | | |
| Lead technically the project | | | | ⊙ | × | ◒ | | | |
| Interface between Customer and XP Delivery Team | | | | ◒ | ⊙ | ◒ | | | |
| Tracks and control project status | | | | ◒ | × | ◒ | | | |
| **XP Delivery Team** | | | | | | | | | |
| Build the software product | | | | | | | ● | × | × |
| Make functional tests | | | | | | | ● | ◒ | × |
| Fit expected quality from Customer and Coach | | | | | | | × | ◒ | × |
| LEVEL OF COMPLIANCE BY ITEM | HIGH | HIGH | NULL | MOD | NULL | MOD | HIGH | MOD | NULL |
| LEVEL OF COMPLIANCE BY METHODOLOGY | MOD | | | LOW | | | MOD | | |
| % OF IMPORTANCE | 33% | | | 33% | | | 33% | | |
| SCORE | 2 | | | 1 | | | 2 | | |
| POINTS | 0.67 | | | 0.33 | | | 0.67 | | |
| LEVEL OF COMPLIANCE | 0.56 | | | | | | | | |

Figure 14. UPEDU Assessment for Roles

| | Customer | | | Project Manager | | | Work Team | | |
|---|---|---|---|---|---|---|---|---|---|
| UPEDU METHODOLOGY | Provide software requirements | Authorize and review project outcomes | Formalize project start and closure | Coordinate project | Manage project resources | Delivery an agreed software product | Build the software product | Fit quality expectations from Customer | Produce technical documentation |
| Stakeholder (customer) | | | | | | | | | |
| Define requirements | ● | × | × | | | | | | |
| Authorize and review project outcomes | × | ● | × | | | | | | |
| Accept or reject final product | × | × | ◒ | | | | | | |
| Project Manager | | | | | | | | | |
| Lead and coach the project | | | | ● | ● | ● | | | |
| Enforce UPEDU approach | | | | ◒ | ◒ | ◒ | | | |
| Interface between Customer and Develoment team | | | | ◒ | ◒ | ◒ | | | |
| Development Team | | | | | | | | | |
| Build the software product | | | | | | | ● | × | ● |
| Make functional tests | | | | | | | ● | ● | ● |
| Fit expected quality from Customer and Coach | | | | | | | × | ● | ● |
| LEVEL OF COMPLIANCE BY ITEM | HIGH | HIGH | LOW | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH |
| LEVEL OF COMPLIANCE BY METHODOLOGY | MOD | | | HIGH | | | HIGH | | |
| % OF IMPORTANCE | 33% | | | 33% | | | 33% | | |
| SCORE | 2 | | | 3 | | | 3 | | |
| POINTS | 0.67 | | | 1.0 | | | 1.00 | | |
| LEVEL OF COMPLIANCE | 0.89 | | | | | | | | |

**Figure 15. SCRUM Assessment for Activities-Roles**

| | ISO/IEC 29110 PM ENTRY PROFILE, ACTIVITIES & TASKS (Compliance %) | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A1. Project Planning | A1.T1 Review the Statement of Work | A1.T2 Identify the specific tasks of the plan | A1.T3 Establish the Estimated Task Duration | A1.T4 Identify and document the Project resources | A1.T5 Establish the Work Team (roles, responsibilities) | A1.T6 Create the Schedule of the Project Tasks (start-completions dates) | A1.T7 Determine Estimated Effort and Cost. | A1.T8 Identify and document the Project risks. | A1.T9 Generate the Integrated Plan | A1.10 Review and Authorization of Project Plan | A1.T11 Establish the project repository | A2. Project Plan Execution | A2.T1 Monitor and Registration of Project Status | A2.T2 Conduct Review Meetings | A3. Project Assessmet and Control | A3.T1 Assess Project Plan vs Real Adrance | A3.T2 Track Changes | A3.T3 Establish Corrective Actions | A4. Project Closure | A4.T1 Formalize Project Closure | A4.T2 Update Project Repository |
| SCRUM METHODOLOGY | | | | | | | | | | | | | | | | | | | | | | | | |
| Planning (Pre-Game) | | | | | | | | | | | | | | | | | | | | | | | | |
| Product Backlog | | ● | × | × | × | × | × | × | × | × | × | ◒ | | × | × | | × | × | × | | × | × |
| Delivery date | | × | ● | ● | × | × | ● | × | × | × | × | × | | × | × | | × | × | × | | × | × |
| Product Backlog prioritization | | × | ● | × | × | × | × | × | × | ● | × | × | | × | × | | × | × | × | | × | × |
| Structure of project team | | × | × | × | × | ● | × | × | × | × | × | ◒ | | × | × | | × | × | × | | × | × |
| Tools to develop the project | | × | × | × | ● | × | × | × | × | × | × | × | | × | × | | × | × | × | | × | × |
| Risk control list | | × | × | × | × | × | × | × | ● | × | × | ◒ | | × | × | | × | × | × | | × | × |
| Budget and cost | | × | × | × | × | × | × | ● | × | × | × | ◒ | | × | × | | × | × | × | | × | × |
| Sprint Planning | | | | | | | | | | | | | | | | | | | | | | | | |
| Product Backlog Creation | | ● | × | × | × | × | × | × | × | ● | × | ◒ | | × | × | | × | × | × | | × | × |
| User Stories Elaboration | | × | ● | × | × | × | × | × | × | × | × | × | | × | × | | × | × | × | | × | × |
| User Stories Size Estimation | | × | × | ● | ⊙ | × | ● | ◒ | × | × | × | × | | × | × | | × | × | × | | × | × |
| User Stories Priorization | | × | × | × | × | ⊙ | ● | ◒ | × | × | × | × | | × | × | | × | × | × | | × | × |
| Sprint Goal Definition | | ● | × | × | × | × | × | × | × | ● | ● | × | | × | × | | × | × | × | | × | × |
| Daily Scrum | | | | | | | | | | | | | | | | | | | | | | | | |
| Reporting of Yesterday's work | | × | × | × | × | × | × | × | × | × | × | × | | × | × | | ● | ● | ◒ | | × | × |
| Reporting of Today's work plan | | × | × | × | × | × | × | × | × | × | × | × | | × | × | | ● | ● | ◒ | | × | × |
| Review of Impediments | | × | × | × | × | × | × | × | × | × | × | × | | × | × | | ● | ● | ◒ | | × | × |
| Sprint Review | | | | | | | | | | | | | | | | | | | | | | | | |
| Review of Increment | | × | × | × | × | × | × | × | × | × | × | × | | ● | ● | | × | × | × | | × | × |
| Review of Project Plan | | × | × | × | × | × | × | × | × | × | × | × | | ● | ● | | × | × | × | | × | × |
| Update Product Backlog | | × | × | × | × | × | × | × | × | × | × | × | | ● | ● | | × | × | × | | × | ⊙ |
| Sprint Retrospective | | | | | | | | | | | | | | | | | | | | | | | | |
| Discussion of Sprint Process | | × | × | × | × | × | × | × | × | × | × | × | | ◒ | ◒ | | × | × | × | | × | × |
| Agreement of Improvements | | × | × | × | × | × | × | × | × | × | × | × | | ◒ | ◒ | | × | × | × | | × | × |
| LEVEL OF COMPLIANCE BY ITEM | | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | MOD | | HIGH | HIGH | | HIGH | HIGH | MODERATE | | NULL | NULL |
| LEVEL OF COMPLIANCE BY METHODOLOGY | | HIGH | | | | | | | | | | | | HIGH | | | HIGH | | | | NULL | |
| % OF IMPORTANCE | | 30% | | | | | | | | | | | | 30% | | | 30% | | | | 10% | |
| SCORE | | 3 | | | | | | | | | | | | 3 | | | 3 | | | | 0 | |
| POINTS | | 0.90 | | | | | | | | | | | | 0.90 | | | 0.90 | | | | 0 | |
| LEVEL OF COMPLIANCE | | | | | | | | | | | | | 0.90 | | | | | | | | | |

**Figure 16. XP Assessment for Activities-Roles**

| | ISO/IEC 29110 PM ENTRY PROFILE, ACTIVITIES & TASKS (Compliance %) | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| XP METHODOLOGY | A1. Project Planning | A1.T1 Review the Statement of Work | A1.T2 Identify the specific tasks of the plan | A1.T3 Establish the Estimated Task Duration | A1.T4 Identify and document the Project | A1.T5 Establish the Work Team (roles, | A1.T6 Create the Schedule of the Project Tasks (start-completions | A1.T7 Determine Estimated Effort and | A1.T8 Identify and document the Project | A1.T9 Generate the Integrated Plan | A1.10 Review and Authorization of Project | A1.T11 Establish the project repository. | A2. Project Plan Execution | A2.T1 Monitor and Registration of Project | A2.T2 Conduct Review Meetings | A3. Project Assessmet and Control | A3.T1 Assess Project Plan vs Real Adrance | A3.T2 Track Changes | A3.T3 Establish Corrective Actions | A4. Project Closure | A4.T1 Formalize Project Closure | A4.T2 Update Project Repository |
| Exploration | | | | | | | | | | | | | | | | | | | | | | | | |
| User Stories Elaboration | | ● | ◒ | × | ● | ◒ | ◒ | × | × | ◒ | × | × | | × | × | | × | × | × | | × | × |
| User Stories Estimation | | × | ◒ | × | ◒ | ◒ | ◒ | × | × | ◒ | × | × | | × | × | | × | × | × | | × | × |
| Spike Elaboration | | × | ◒ | × | ◒ | ◒ | ◒ | ◒ | × | ◒ | × | × | | × | × | | × | × | × | | × | × |
| Release Planning | | | | | | | | | | | | | | | | | | | | | | | | |
| User Stories Prioritization | | × | × | × | × | × | × | × | × | × | × | × | | × | ◒ | | × | × | × | | × | × |
| Commitment Schedule Elaboration | | × | × | × | × | × | × | × | × | × | × | × | | ◒ | ◒ | | × | × | × | | × | × |
| Iteration Planning | | | | | | | | | | | | | | | | | | | | | | | | |
| User Stories Iteration Selection | | × | × | × | × | × | × | × | × | × | × | × | | × | × | | × | × | × | | × | × |
| Engineering Tasks Breaking | | × | × | × | × | × | × | × | × | × | × | × | | × | × | | × | × | × | | × | × |
| Engineering Tasks Estimation | | × | × | × | × | × | × | × | × | × | × | × | | × | × | | × | × | × | | × | × |
| Iteration Plan Elaboration | | × | × | × | × | × | × | × | × | × | × | × | | × | × | | × | × | × | | × | × |
| Customer Test | | | | | | | | | | | | | | | | | | | | | | | | |
| Acceptance Tests Definitions | | × | × | × | × | × | × | × | × | × | × | × | | × | × | | × | × | × | | ◒ | ◒ |
| Acceptance Tests Execution | | × | × | × | × | × | × | × | × | × | × | × | | × | × | | × | × | × | | ◒ | ◒ |
| LEVEL OF COMPLIANCE BY ITEM | | HIGH | MOD | NULL | HIGH | MOD | MOD | LOW | NULL | MOD | NULL | NULL | | MOD | MOD | | NULL | NULL | NULL | | MOD | MOD |
| LEVEL OF COMPLIANCE BY METHODOLOGY | | LOW | | | | | | | | | | | | MOD | | | NULL | | | | MOD | |
| % OF IMPORTANCE | | 30% | | | | | | | | | | | | 30% | | | 30% | | | | 10% | |
| SCORE | | 1 | | | | | | | | | | | | 2 | | | 0 | | | | 2 | |
| POINTS | | 0.30 | | | | | | | | | | | | 0.60 | | | 0.00 | | | | 0.2 | |
| LEVEL OF COMPLIANCE | | | | | | | | | | | | | 0.37 | | | | | | | | | |

**Figure 17. UPEDU Assessment for Activities-Roles**

| | ISO/IEC 29110 PM ENTRY PROFILE, ACTIVITIES & TASKS (Compliance %) | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UPEDU METHODOLOGY | A1. Project Planning | A1.T1 Review the Statement of Work | A1.T2 Identify the specific tasks of the | A1.T3 Establish the Estimated Task | A1.T4 Identify and document the Project | A1.T5 Establish the Work Team (roles, | A1.T6 Create the Schedule of the Project Tasks (start- | A1.T7 Determine Estimated Effort and | A1.T8 Identify and document the Project | A1.T9 Generate the Integrated Plan | A1.T10 Review and Authorization of | A1.T11 Establish the project repository. | A2. Project Plan Execution | A2.T1 Monitor and Registration of | A2.T2 Conduct Review Meetings | A3. Project Assessmet and Control | A3.T1 Assess Project Plan vs Real Advance | A3.T2 Track Changes | A3.T3 Establish Corrective Actions | A4. Project Closure | A4.T1 Formalize Project Closure | A4.T2 Update Project Repository |
| **Software Development Planning** | | | | | | | | | | | | | | | | | | | | | | | |
| Plan Phases and Iterations | | ● | ● | ● | ● | ● | ● | ● | × | ● | × | × | | × | × | | × | × | × | | × | ◒ |
| Develop Measurement Plan | | × | ● | ● | ◒ | ◒ | ◒ | ◒ | × | × | × | × | | × | × | | × | × | × | | × | ◒ |
| Review Project Plannning | | × | × | × | × | × | × | × | ◒ | × | ● | ● | | × | × | | × | × | × | | × | ◒ |
| **Next Iteration Planning** | | | | | | | | | | | | | | | | | | | | | | | |
| Develop Iteration Plan - goals | | ● | × | × | × | × | × | × | × | × | × | × | | ◒ | ◒ | | × | × | × | | × | ◒ |
| Develop Iteration Plan - efforts | | × | ● | ● | ● | × | × | × | × | × | × | × | | ◒ | ◒ | | × | × | × | | × | ◒ |
| Develop Iteration Plan - risks | | × | × | × | × | × | × | × | ● | × | × | × | | ◒ | ◒ | | × | × | × | | × | ◒ |
| **Project Control** | | | | | | | | | | | | | | | | | | | | | | | |
| Schedule and Assign Work Orders | | × | × | × | × | × | × | × | × | × | × | × | | ◒ | ◒ | | ● | ● | ● | | × | × |
| Monitor Project Status | | × | × | × | × | × | × | × | × | × | × | × | | ● | ● | | ● | ● | ● | | × | × |
| **Configuration & Change Management** | | | | | | | | | | | | | | | | | | | | | | | |
| Plan Project Configuration | | × | × | × | × | × | × | × | × | × | × | × | | × | × | | × | × | × | | × | ⊙ |
| Change and Deliver Configuration Items | | × | × | × | × | × | × | × | × | × | × | × | | × | × | | × | ● | ● | | × | ● |
| Monitor and Report Configuration Status | | × | × | × | × | × | × | × | × | × | × | × | | ● | × | | ● | × | × | | × | ● |
| Manage Change Requests | | × | × | × | × | × | × | × | × | × | × | × | | × | × | | × | ● | ◒ | | × | ● |
| LEVEL OF COMPLIANCE BY ITEM | | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | | HIGH | HIGH | | HIGH | HIGH | HIGH | | NULL | HIGH |
| LEVEL OF COMPLIANCE BY METHODOLOGY | | HIGH | | | | | | | | | | | | HIGH | | | HIGH | | | | MODERATE | |
| % OF IMPORTANCE | | 30% | | | | | | | | | | | | 30% | | | 30% | | | | 10% | |
| SCORE | | 3 | | | | | | | | | | | | 3 | | | 3 | | | | 2 | |
| POINTS | | 0.90 | | | | | | | | | | | | 0.90 | | | 0.9 | | | | 0.2 | |
| LEVEL OF COMPLIANCE | | | | | | | | | | | | | 0.97 | | | | | | | | | |

**Figure 18. SCRUM Assessment for Work Products**

| | ISO/IEC 29110 PM ENTRY PROFILE, ARTIFACTS (Compliance %) | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SCRUM METHODOLOGY | Acceptance Record | | | Change Request | | | Meeting Record | | | Progress Status Record | | | Project Plan | | | | | Project Repository | | | | Software Configuration (Accepted by Customer) | | | |
| | List of Acceptance Criteria | List of delivered elements. | Project identification data and signatures | Purpose and Description of Change | Impact, Criticality and Status of Change | Cost of Change | Purpose and Attendees | Raised and Open Issues | Agreements | Status of Planned vs Realized Goals, Tasks and Schedule | Status of Costs vs Budgets | Status of Risks | Product Description, Scope and Deliverables | Task Plan, Work Team, WBS, Schedule, and Resources | Verification, validation and reviews Tasks with Customer and Work Team | Estimated Effort and Cost | Risks Lists | PM Work Products (Artifacts) | SI Work Products (Deliverables) | Changes Tracking | Status Project Reports | Requirements Specification | Released Software | User Documentation | Technical Documentation |
| **User Needs List** | | | | | | | | | | | | | | | | | | | | | | | | | |
| List of requirements in the project | ● | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | ● | × | × | × |
| Establish the direction of the team during the project | × | ◒ | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | ◒ |
| Give priorities and goals in the project | × | × | ◒ | × | × | × | × | × | × | × | × | × | ◒ | × | × | × | × | × | × | × | × | × | × | × | ◒ |
| **Product Backlog** | | | | | | | | | | | | | | | | | | | | | | | | | |
| List of User Stories (features, new functions, requirements, improvements and changes in the future versions). | × | × | × | × | × | × | × | × | × | × | × | × | ● | × | × | × | × | ● | × | × | × | ● | × | × | × |
| Description, order, estimate, risks and value for each User Story. | ⊙ | ⊙ | ⊙ | ◒ | ⊙ | ⊙ | × | × | × | × | × | ● | ◒ | × | × | ● | ● | ● | × | × | × | × | × | × | × |
| Product owner and final market user feed back and acceptance criteria. | × | × | × | × | × | × | × | × | × | × | × | × | × | × | ◒ | × | × | ● | × | ⊙ | × | × | × | × | × |
| **Sprint Backlog** | | | | | | | | | | | | | | | | | | | | | | | | | |
| List of Selected User Stories (features, new functions, requirements, improvements) for this Increment. | × | × | × | × | × | × | × | × | × | × | × | × | ● | × | × | × | × | ● | × | × | × | ● | × | × | ◒ |
| Sprint Goal. | × | × | × | × | × | × | × | × | × | ◒ | × | × | ● | × | × | × | × | ● | × | × | × | × | × | × | × |
| Task Plan for achieving the Sprint Goal | × | × | × | × | × | × | × | × | × | × | × | × | ● | × | × | × | × | ● | × | × | × | × | × | × | × |
| Burn Down Chart for a Sprint (optional) | × | × | × | × | × | × | × | × | × | ◒ | ● | × | × | ◒ | × | ◒ | × | ● | × | × | ⊙ | × | × | × | × |
| **Increment** | | | | | | | | | | | | | | | | | | | | | | | | | |
| Set of User Stories implemented (DONE) in the last Sprint. | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | ● | × | × | × | ● | × | ● |
| Value provided to previous Increments from previous Sprints. | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | ● | × | × | × | × | × | × |
| Review of potential adjustments to the Sprint Backlog | × | × | × | ● | ◒ | × | ◒ | ◒ | × | × | × | × | × | ◒ | × | ◒ | × | × | ● | ● | × | × | × | × | × |
| **Acceptance Criteria (Implicit artifact)** | | | | | | | | | | | | | | | | | | | | | | | | | |
| Specific and detailed specifications on User Stories to be used for Sprint acceptance. | ● | ◒ | × | × | × | × | × | × | × | × | × | × | ◒ | × | ◒ | × | × | ● | ⊙ | × | ⊙ | × | × | × | ◒ |
| LEVEL OF COMPLIANCE BY ITEM | HIGH | LOW | NULL | HIGH | LOW | NULL | LOW | LOW | LOW | MODERATE | HIGH | HIGH | HIGH | HIGH | MODERATE | HIGH | HIGH | MODERATE | MODERATE | LOW | LOW | HIGH | HIGH | NULL | MODERATE |
| LEVEL OF COMPLIANCE BY ARTIFACT | MODERATE | | | LOW | | | LOW | | | MODERATE | | | HIGH | | | | | MODERATE | | | | MODERATE | | |
| % OF IMPORTANCE | 10% | | | 10% | | | 10% | | | 10% | | | 20% | | | | | 20% | | | | 20% | | |
| SCORE | 3 | | | 1 | | | 1 | | | 1 | | | 3 | | | | | 2 | | | | 2 | | |
| POINTS | 0.30 | | | 0.1 | | | 0.1 | | | 0.1 | | | 0.60 | | | | | 0.40 | | | | 0.40 | | |
| LEVEL OF COMPLIANCE | 0.70 | | | | | | | | | | | | | | | | | | | | | | | | |

**Figure 19. XP Assessment for Work Products**

| | ISO/IEC 29110 PM ENTRY PROFILE, ARTIFACTS (Compliance %) | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| XP METHODOLOGY | Acceptance Record | | | Change Request | | | Meeting Record | | | Progress Status Record | | | Project Plan | | | | | Project Repository | | | | Software Configuration (Accepted by Customer) | | | |
| | List of Acceptance Criteria | List of delivered elements. | Project identification data and signatures | Purpose and Description of Change | Impact, Criticality and Status of Change | Cost of Change | Purpose and Attendees | Raised and Open Issues | Agreements | Status of Planned vs Realized Goals, Tasks and Schedule | Status of Costs vs Budgets | Status of Risks | Product Description, Scope and Deliverables | Task Plan, Work Team, WBS, Schedule, and Resources | Verification, validation and reviews Tasks with Customer and Work Team | Estimated Effort and Cost | Risks Lists | PM Work Products (Artifacts) | SI Work Products (Deliverables) | Changes Tracking | Status Project Reports | Requirements Specification | Released Software | User Documentation | Technical Documentation |
| **User Stories** | | | | | | | | | | | | | | | | | | | | | | | | | |
| Vision or Mission of Product | × | × | × | × | × | × | ⊙ | ⊙ | ⊙ | × | × | × | ● | × | × | × | × | ● | × | × | × | ● | × | ⊙ | ⊙ |
| Description of specific requirements (Index Cards) | ⊙ | ⊙ | ⊙ | ◒ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | × | × | × | ● | × | × | × | × | ● | × | × | × | ● | × | ⊙ | ⊙ |
| **Architectural Spikes** | | | | | | | | | | | | | | | | | | | | | | | | | |
| Small prototype | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | × | × | × | ● | × | × | ● | ⊙ | ● | × | × | × | ● | × | ⊙ | ⊙ |
| Technology Exploration | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | ⊙ | × | × | × | ● | × | × | ● | ⊙ | ● | × | × | × | ● | × | ⊙ | ⊙ |
| **Release Plan** | | | | | | | | | | | | | | | | | | | | | | | | | |
| Effort Estimation of User Stories | × | × | × | × | × | × | ⊙ | ⊙ | ⊙ | × | × | × | × | ● | × | ● | × | ● | × | × | × | × | × | × | × |
| List of Prioritized User Stories | × | × | × | × | × | × | ⊙ | ⊙ | ⊙ | × | × | × | × | ● | × | × | × | ● | × | × | × | × | × | × | × |
| Commitment Schedule | × | × | × | × | × | × | ⊙ | ⊙ | ⊙ | × | × | × | × | ● | × | × | × | ● | × | × | × | × | × | × | × |
| **Iteration Plan** | | | | | | | | | | | | | | | | | | | | | | | | | |
| Selected Subset of Prioritized User Stories | × | × | × | × | × | × | ⊙ | ⊙ | ⊙ | ◒ | ◒ | × | ● | ● | × | × | ⊙ | ● | × | × | × | × | × | × | × |
| Engineering Tasks by User Story | × | × | × | × | × | × | ⊙ | ⊙ | ⊙ | ◒ | ◒ | × | ● | ● | × | × | ⊙ | ● | × | × | × | × | × | × | × |
| Iteration Schedule | × | × | × | × | × | × | ⊙ | ⊙ | ⊙ | ◒ | ◒ | × | ● | ● | × | × | ⊙ | ● | × | × | × | × | × | × | × |
| **Iteration Integrated Code (SI)** | | | | | | | | | | | | | | | | | | | | | | | | | |
| Iteration code | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | ● | ⊙ | × | × | × | × | × | ● | × | ● |
| Integrated Iterations Code | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | ● | ⊙ | × | × | × | ● | × | ● |
| **Test Scenarios (SI)** | | | | | | | | | | | | | | | | | | | | | | | | | |
| Whitebox Tests | ● | ● | ● | × | × | × | × | × | × | × | × | × | × | ◒ | ● | × | × | × | × | × | × | × | × | × | ● |
| Unit Tests | ● | ● | ● | × | × | × | × | × | × | × | × | × | × | ◒ | ● | × | × | × | × | × | × | × | × | × | ● |
| **Customer Approval** | | | | | | | | | | | | | | | | | | | | | | | | | |
| Blackbox Tests | ● | ● | ● | × | × | × | × | × | × | × | × | × | × | ◒ | ● | × | × | × | × | × | × | × | × | × | × |
| Acceptance Tests | ● | ● | ● | × | × | × | × | × | × | × | × | × | × | ◒ | ● | × | × | × | × | × | × | × | × | × | × |
| **Release Version** | | | | | | | | | | | | | | | | | | | | | | | | | |
| Stable Version | ● | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | ● | × | × | × | × | ● | × | × |
| Objective Functionality | ● | ● | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | ● | × | × | × | × | ● | × | × |
| Signed documentation | × | × | ● | × | × | × | × | × | × | × | × | × | × | × | × | × | × | ● | × | × | × | × | × | × | ● |
| LEVEL OF COMPLIANCE BY ITEM | HIGH | HIGH | HIGH | MODERATE | LOW | LOW | LOW | LOW | LOW | MODERATE | MODERATE | NULL | HIGH | HIGH | HIGH | HIGH | LOW | HIGH | HIGH | LOW | NULL | HIGH | HIGH | LOW | LOW |
| LEVEL OF COMPLIANCE BY ARTIFACT | HIGH | | | LOW | | | LOW | | | LOW | | | MODERATE | | | | | MODERATE | | | | MODERATE | | |
| % OF IMPORTANCE | 10% | | | 10% | | | 10% | | | 10% | | | 20% | | | | | 20% | | | | 20% | | |
| SCORE | 3 | | | 1 | | | 1 | | | 1 | | | 3 | | | | | 2 | | | | 2 | | |
| POINTS | 0.30 | | | 0.1 | | | 0.1 | | | 0.1 | | | 0.60 | | | | | 0.40 | | | | 0.40 | | |
| LEVEL OF COMPLIANCE | 0.60 | | | | | | | | | | | | | | | | | | | | | | | | |

## Figure 20. UPEDU Assessment for Work Products

| UPEDU METHODOLOGY | ISO/IEC 29110 PM ENTRY PROFILE, ARTIFACTS (Compliance %) | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acceptance Record | | | Change Request | | | Meeting Record | | | Progress Status Record | | | Project Plan | | | | | Project Repository | | | | Software Configuration (Accepted by Customer) | | | |
| | List of Acceptance Criteria | List of delivered elements. | Project identification data and signatures | Purpose and Description of Change | Impact, Criticality and Status of Change | Cost of Change | Purpose and Attendees | Raised and Open Issues | Agreements | Status of Planned vs Realized Goals, Tasks and Schedule | Status of Costs vs Budgets | Status of Risks | Product Description, Scope and Deliverables | Task Plan, Work Team, WBS, Schedule, and Resources | Verification, validation and reviews Tasks with Customer and Work Team | Estimated Effort and Cost | Risks Lists | PM Work Products (Artifacts) | SI Work Products (Deliverables) | Changes Tracking | Status Project Reports | Requirements Specification | Released Software | User Documentation | Technical Documentation |
| **Software Development Plan** | | | | | | | | | | | | | | | | | | | | | | | | | |
| Measurements Plan (quality, schedule and cost metrics) | × | × | × | × | × | × | × | × | × | × | × | × | ● | × | × | ● | × | ● | × | × | × | × | × | × | × |
| Iterations Plan (status, tasks, resources, goals, and risks) | × | × | × | × | × | × | × | × | × | × | × | × | × | ● | × | × | × | ● | × | × | × | × | × | × | × |
| Risks List | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | ● | ● | × | × | × | × | × | × | × |
| Work Order 0 ( schedule and budget) | × | × | × | × | × | × | × | × | × | × | × | × | × | ● | × | × | × | ● | × | × | × | × | × | × | × |
| **Iteration Plan N** | | | | | | | | | | | | | | | | | | | | | | | | | |
| Iteration Plan (status, tasks, resources, goals, and risks) | × | × | × | × | × | × | × | × | × | × | × | × | ● | ● | × | × | ● | ● | × | × | × | × | × | × | × |
| **Work Order N** | | | | | | | | | | | | | | | | | | | | | | | | | |
| Work Breakdown Structure (WBS) identification | × | × | × | × | × | × | × | × | × | × | × | × | × | ● | × | × | × | ● | × | × | × | × | × | × | × |
| Associated Change Requests | × | × | × | ● | ● | ● | × | × | × | × | ● | × | × | × | × | × | × | ● | × | ● | × | × | × | × | × |
| Effort and Budget (Resources) | × | × | × | × | ● | ● | × | × | × | × | × | × | × | ● | × | × | × | ● | × | × | × | × | × | × | × |
| Description of work and expected outputs | ● | ● | × | ● | × | × | × | × | × | × | × | × | ● | × | × | × | × | ● | × | × | × | × | × | × | × |
| **Review Record** | | | | | | | | | | | | | | | | | | | | | | | | | |
| Objectives for Review | ● | × | × | ● | ◒ | ◒ | ● | × | × | × | × | × | × | × | ● | × | × | ● | × | × | × | × | × | × | × |
| Participants and Artifacts for Review | × | × | × | × | × | × | ● | × | × | × | × | × | × | × | ● | × | × | ● | × | × | × | × | × | × | × |
| Agreements (actions, status, problems) | × | ● | ● | × | ◒ | ◒ | × | ● | ● | × | × | × | × | × | ● | × | × | ● | × | ● | ● | × | × | × | × |
| **Project Measurements** | | | | | | | | | | | | | | | | | | | | | | | | | |
| Project Repository | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | ● | ● | ● | ● | ◒ | ◒ | ◒ | ◒ |
| Development and Integration Workspace | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | ● | × | × |
| Metrics (cost, schedule, quality) | × | × | × | × | × | × | × | × | × | ● | ● | ● | × | × | × | × | × | × | × | × | ● | × | × | × | × |
| LEVEL OF COMPLIANCE BY ITEM | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | HIGH | MODERATE | HIGH | MODERATE | MODERATE |
| LEVEL OF COMPLIANCE BY | HIGH | | | HIGH | | | HIGH | | | HIGH | | | HIGH | | | | | HIGH | | | | MODERATE | | | |
| % OF IMPORTANCE | 10% | | | 10% | | | 10% | | | 10% | | | 20% | | | | | 20% | | | | 20% | | | |
| SCORE | 3 | | | 3 | | | 3 | | | 3 | | | 3 | | | | | 3 | | | | 2 | | | |
| POINTS | 0.3 | | | 0.3 | | | 0.3 | | | 0.3 | | | 0.6 | | | | | 0.6 | | | | 0.4 | | | |
| LEVEL OF COMPLIANCE | 0.93 | | | | | | | | | | | | | | | | | | | | | | | | |

# Call for Articles

## International Journal of Information Technologies and Systems Approach

Volume 10 • Issue 1 • January-June 2017 • ISSN: 1935-570X • eISSN: 1935-5718
***An official publication of the Information Resources Management Association***

## MISSION

The **International Journal of Information Technologies and the Systems Approach (IJITSA)** publishes research articles that are globally relevant for IT stakeholders and where a systemic interdisciplinary and/or a multi-methodology research perspective are considered. IJITSA improves scientific and technological knowledge regarding the engineering and management of IT systems by fostering the utilization of the Systems Approach.

## COVERAGE AND MAJOR TOPICS

**The topics of interest in this journal include, but are not limited to:**

Agent-based simulation • Axiology of systems • Complex systems foundations • Complex systems frameworks, models and processes • Critical heuristics systems • Critical systems • Epistemology of systems • Information security systems • Mathematical analysis of systems • Multi-scale analysis • Ontology of sytems • Philosophy of systems sciences • Socio-Technical Systems • Soft systems methodology • System Dynamics • System of systems concepts • Systemic action research • Systemic analysis of ISO, IEEE, ANSI, TIA standards • Systemic business process-oriented frameworks • Systemic case study • Systemic conceptual study • Systemic decision-making models • Systemic design of IT systems • Systemic design of service systems • Systemic design of SoS (System of Systems) • Systemic design of systems • Systemic evaluation of IT systems • Systemic evaluation of software development tools • Systemic experiments • Systemic green IT frameworks • Systemic implementation of IT systems • Systemic IT service management frameworks • Systemic IT service management standards (ITIL, ISO 20000, CobIT, CMMI-SVC, etc) • Systemic review of ISO, CMMI and IEEE systems engineering standards • Systemic review of software development agile methods • Systemic review of software development business process-oriented methods • Systemic review of software development service-oriented methods • Systemic risk management • Systemic service-oriented frameworks • Systemic software process frameworks (CMMI, ISO 12207, ISO 29110, IEEE 1471, etc) • Systemic surveys • Systemic TOEP feasibility studies • Systems Analysis • Systems design • Systems engineering frameworks • Systems evaluation • Systems simulation • Total systems intervention • Viable system models

**ALL INQUIRIES REGARDING IJITSA SHOULD BE DIRECTED TO THE ATTENTION OF:**

Manuel Mora, Editor-in-Chief • IJITSA@igi-global.com

**ALL MANUSCRIPT SUBMISSIONS TO IJITSA SHOULD BE SENT THROUGH THE ONLINE SUBMISSION SYSTEM:**

http://www.igi-global.com/authorseditors/titlesubmission/newproject.aspx

IDEAS FOR SPECIAL THEME ISSUES MAY BE SUBMITTED TO THE EDITOR(S)-IN-CHIEF

**PLEASE RECOMMEND THIS PUBLICATION TO YOUR LIBRARIAN**

For a convenient easy-to-use library recommendation form, please visit:
http://www.igi-global.com/IJITSA