

A standard-based framework to integrate software work in small settings

Mary-Luz Sanchez-Gordon^a, Antonio de Amescua^a, Rory V. O'Connor^b, Xabier Larrucea^c

^a *Universidad Carlos III de Madrid, Department of Computer Science and Engineering, Leganés, Madrid, Spain*

^b *Dublin City University, Dublin, Ireland*

^c *Tecnalia, Bizkaia, Spain,*

Abstract

Small software companies have to work hard in order to survive. They usually find it challenging to spend time and effort on improving their operations and processes. Therefore, it is important to address such needs by the introduction of a proposed framework that specifies ways of getting things done while consciously encourage them to enhance their ability to improve. Although there are many software process improvement approaches, none of them address the human factors of small companies in a comprehensive and holistic way. Samay is a proposed framework to integrate human factors in the daily work as a way to deal with that challenge. This study suggests managing human factors but pointing out the software process life cycle. The purpose is to converge toward a continuous improvement by means of alternative mechanisms that impact on people. This framework was developed based upon reviews of relevant standards (such as ISO/IEC 29110, ISO 10018, OMG Essence and ISO/IEC 33014) and previously published studies in this field. Moreover, an expert review and validation findings supported the view that Samay could support practitioners when small software companies want to start improving their ways of work.

Keywords: Small Companies; VSE; Small Settings; Software Process Improvement; Human Factors; Socio-technical System

1. Introduction

Although there is not worldwide official statistics, it is known that the vast majority of enterprises are very small or small. According to Eurostat, 98.8% of independent European enterprises have up to 9 employees (called microenterprises), and another 6,5% have from 10 to 49 employees [1]. Likewise, OECD (Organization for Economic Cooperation and Development) indicates that microenterprises account for 70 to 90% [2]. At a time when technology advances almost daily, software development companies are under increasing pressure to improve productivity while maintaining quality and keeping costs to a minimum [3] and countries need the capacity to adopt, adapt and develop relevant software [4]. Moreover, it has been observed that one of the first challenges for small companies is that their primary business objective is to survive [5,6] because their resources are scarce. They can have a low software development process priority [7], since they are focused on the product quality and delivery time rather than in the process quality [8]. Nevertheless, these companies develop and/or maintain systems and/or software that is used in larger systems [9,10]. Therefore, it is of particular importance to ensure that this sector can support the public and private sector local needs [4]. It is crucial that changes and adaptation processes be triggered only for factors that are really relevant for the company [1].

In practice, software development is beset with many challenges and constraints [11]. The implementation of controls and structures to properly manage their software development activity is necessary and constitutes a major challenge [12]. In fact, there are multiple approaches for organizing the software development process and multiple factors influencing the software development process [13]. Software is a byproduct of human activities that incorporates our problem solving capabilities, cognitive aspects, and social interaction [14]. People are fundamental in the software process and in its assessment and improvement [15]. Moreover, the software development process has been considered a “*socio-technical system*”, where organizational and human aspects have a key role and have to be supported by technology in a way that is human and organization-driven [16]. Indeed, it can be stated that the software industry is highly dependent on people [17,18]. The human factor in software development is the ingredient that ultimately gives a project team its soul [19]. As result, technical aspects are not enough to ensure the success of a human activity due to human factors impact software process and software process improvement (SPI) initiatives.

In spite of their importance, recent studies show that very small enterprises still struggle to implement software process [12,20] and SPI initiatives successfully [21–23]. Most of them usually lack the knowledge and practical experience about it and cannot afford the resources for, or see a net benefit in, establishing software processes as defined by current standards (e.g. ISO/IEC 12207) and maturity models such as the Capability Maturity Model Integration CMMI [18]. Recent initiatives, such as ISO/IEC 29110, show similar challenges [21,24], but industrial experience demonstrate that small enterprise are currently not too far from satisfying those practices [25]. Taking these drawbacks into account and due to the apparent absence of any established framework for integrating human factors and lifecycle development process in small software companies, this paper presents Samay framework for improving ways of work on a daily basis in small software companies. Samay (Quechua word meaning spirit or soul) provides a core and two groups of elements: complement and support. The core includes a *software development process* along with *techniques* for dealing with human factors. First group is directly linked to the development process. Second group facilitates the adoption of the framework. In order to validate the framework an expert validation was carried out and a case study will be conducted in the future.

The rest of this paper is organized as follows. Section 2 gives an overview of the existing international standards to define Samay framework. Section 3 outlines the research approach carried out. Section 4 presents the different elements that comprise the proposed framework and Section 5 shows how the validation was done and its results. Finally, Section 6 draws some conclusions and future work.

2. Overview of International Standards

The relationship between the success of a software company and the software process it utilized has been investigated [26,27] showing the need for all organizations, not just very small companies to pay attention to software process practices such as international standards [28]. It is the premise of this study that in order to support small software companies for improving their daily work, international standards can be used as a comprehensive point of reference. However, small companies require creation of a framework structured and systematic, and this paper presents an approach suited to integrate the essence of international standards into a practical framework in order to be ready or being prepared for dealing more appropriately with technical aspect of SPI initiatives.

3.1. Description of ISO/IEC 29110

ISO/IEC 29110¹ defines the minimum activities and work products that require VSEs to perform [29]. ISO/IEC 29110 provides a standard according to VSEs characteristics and needs [30] and it is considered an emerging SPI initiative [31]. Although, other initiatives are devoted to small entities such as Competisoft from Latin America and ITmark from Europe [32], ISO/IEC 29110 is becoming the widely adopted standard [18,31].

The overall objective of this standard is to assist and encourage VSEs in assessing and improving their software process. Until now, a series of pilot project for the software engineering profile standard have been completed in several countries [33]. However, there are many settings of VSEs and therefore have been proposed four profiles: entry, basic, intermediate and advanced, but only the first two have been released (2012 and 2011, respectively). The authors of this standard state that it is intended to be used by the VSE to establish processes to implement any development approach or methodology – e.g. agile, evolutionary, incremental, test driven development - based on the VSE organization or project needs. The ISO/IEC 29110 provides two main categories of processes: Project Management (PM) and Software Implementation (SI) [30]. PM process aims to establish and carry out in a systematic way the Tasks of the software implementation project, which allows complying with the objectives of the project in the expected quality, time and cost. SI process aims to systematically analyze, design, construction, integration and testing of the new or modified software products according to the specified requirements.

In summary, PM process uses the Customer's Statement of Work to elaborate the Project Plan. The PM project assessment and control tasks compare the project progress against the Project Plan and actions are taken to eliminate deviations or incorporate changes to the Project Plan. The PM project closure activity delivers the Software Configuration, produced by SI, and gets the Customer's acceptance to formalize the end of the project. A Project Repository is established to save the work products and to control its versions during the project. Therefore, the Customer provides a Statement of Work as an input to Project Management process and receives a Software Configuration as a result of Software Implementation process execution. The execution of the SI process is driven by the Project Plan. SI process starts with an initiation activity of the Project Plan revision. Project Plan will guide the execution of the software requirements analysis, software architectural and detailed design, software construction, software integration and test, and product delivery activities. In order to remove product's defects verification, validation and test Tasks are included in the activities workflow.

3.2. Description of ISO 10018

Late in 2012, the new ISO 10018 standard for quality management was published. Accordingly to ISO 10018 [34], the overall performance of a Quality Management System (QMS) and its processes ultimately depends on the involvement of competent people and whether they are properly introduced and integrated into the organization. The involvement of people is important in order for an organization's QMS to achieve outcomes which are consistent and aligned with their strategies and values. It is crucial to identify, develop and evaluate the knowledge, skills, behavior and work environment required for the effective involvement of people with the necessary competences.

The ISO 10018 standard has been released in 2012 and provides guidelines for human factors which influence people involvement and competence, and creates value that helps to achieve the organization's objectives. This standard is applicable to any organization, regardless of size, type, or activity [34]. Furthermore, the terms and definitions given in ISO 9000 and the following five apply: i) Competence mean ability to apply knowledge and skills to achieve intended results, ii) Competence acquisition means process to ensure that competence is attained by a person, a group of people, or an organization, iii) Competence development means process to increase the competence of a person, a group of people, or an organization, iv) Human factors mean physical or cognitive characteristics, or social behavior, of a person, and v) Involvement means engagement in, and contribution to, shared objectives.

¹ freely available at <http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>

The management of people involvement and competence requires both leadership involvement and strategy, and people involvement and competence acquisition process. The process is based on the four steps outlined below: i) Analysis: data are collected and analyzed in relation to the organization's short- and long-term objectives for people involvement and competence; ii) Planning: procedures are established and maintained to plan the people involvement and competence acquisition process on an organizational, group and individual level; iii) Implementation: the plans and associated actions are implemented in order to achieve the objective of people involvement and competence; iv) Evaluation: plans, actions and outcomes are reviewed and evaluated for continual improvement. There should be a review carried out at every step to ensure that the input and output data are correct [34]. These steps apply to all levels of the organization, group and individual. Furthermore, there are two annexes, human factors and self-assessment.

3.3. Description of OMG Essence

In 2010 Software Engineering Method and Theory (SEMAT)², an important initiative for the software engineering community, emerged for the purpose of generating a theoretical basis for the discipline [35]. Thus, SEMAT launched a call for action to re-found Software Engineering [36]. Later, the Object Management Group (OMG)³ endorsed it as a request for proposals to deal with SEMAT concerns. That is not surprise, because several influential members of the software engineering community who participate in SEMAT are involved in organizations that are OMG members. OMG is an IT consortium that was established in 1989.

In 2015, ESSENCE – Kernel and Language for Software Engineering Methods, version 1.1⁴, was published [35]. This specification defines a kernel and a language for the creation, use, and improvement of software engineering methods. The Language is a domain-specific language to define Kernels, Practices, and Methods [35]. The Kernel is a conceptual model of software development domain [37] and identifies a core set of alphas separated into different areas of concerns and their relationships. The kernel elements have three categories: alphas, activity spaces and competencies [35].

These alphas are namely, Opportunity, Stakeholders, Requirements, Software System, Team, and Work and Way of Working. Alphas, activity spaces and competencies are grouped in three areas of concern: customer, solution and endeavor. Moreover, it provides a framework for assessing the progress and health of software development projects and a foundation for method and practice definitions [37]. A practice provides a systematic and repeatable way of work focused on the achievement of an objective, that addresses a specific aspect of software development or teamwork [35]. A method is the composition of a Kernel and a set of Practices to fulfill a specific purpose [35].

3.4. Description of ISO/IEC 33014

In 2013, the standard ISO/IEC 33014 was published for process improvement in IT organizations. This standard describes the identification of the most important process improvement support elements to enhance and includes an evaluation [38]. However, it focuses on activities without precisely defining the required artefacts [39]. ISO/IEC 33014 is based on the ISO/IEC 15504 – part 4 and 7 – and addresses two dimensions: i) three control levels (strategic, tactical and operational) and ii) three improvement perspectives (Organizational improvability, Project improvability and Process improvement) [40].

On one hand, the strategic level gives the basis for setting up an improvement program, including definition of scope and identification of the overall change strategy. The tactical level describes the best support elements to bolster a process improvement project's success or an organizational success rate with process improvement programs in general. The operational level includes activities related to the implementation of the improvement [40]. On the other hand, the process perspective deals with process improvement as a programme o project. Organizational perspective addresses improvement of organizational improvability in order to ensure success with improvement projects[38]. And project perspective improves a project's improvability and reach improvement success [40]. Note that the last two perspectives of the ISO/IEC 33014 are the ImprovAbility model and method used in the organizational perspective and project perspective [38]. The ImprovAbility model was made to support process improvement activities. The model for organizations suggests a defined approach to analyze 20 parameters that is known to be a source of risk for at process improvement project. The parameters are divided into four groups called Foundation, Initiation, Projects and In Use. The model for projects is a subset of the organizational level ImprovAbility model. The subset consists of 17 out of the original 20 parameters.

Finally, the main idea of ISO/IEC 33014 is to support organizations with a framework to enhance their understanding of their situation in relation to the ability to improve, and what to focus on to improve their abilities to improve, as well as to support organizations and individuals in their work with process improvement [38].

3. Research approach

² <http://www.semat.org/>

³ <http://www.omg.oadarg/>

⁴ freely available at <http://www.omg.org/spec/Essence/1.1/>

A structured and systematic research approach was used in this study, bearing in mind that the solution should be useful and practical. The approach can be outlined with the following three phases: State of the art, Design of the proposed framework and Validation.

3.1. State of the art

The first phase entailed conducting a systematic literature review (SLR) in order to identify and analyze various standards, models, methodologies and frameworks SPI in small companies [41]. It was carried out following Kitchenham and Charters' guidelines on SLR [42]. There were five important contributions [22,43–46] and their findings support the results of previous studies. From there, the most relevant and recent approaches are: one standard (ISO/IEC 29110), thirteen of the most recognized models and methods, such as iFlap, OWPL and Mares for their current publications, five well known frameworks - MoProSoft, COMPETISOFT, MPS.BR, ITMark, Tutelkan – and two techniques, Pisko and its extension LAPPI. In addition, previous research focused on the degree of knowledge and implementation of SPI models in this environment was collected, these results were contrasted with our own study, which was conducted by the first author, and grounded in a set of interviews held in Ecuador [24]. As a result, the need for supporting the management of software development process arose in the first instance because most of the approaches are focused on solving the technical aspects and overlook other factors related to the social aspects [23]. Indeed, human factors are still considered marginal and treated with common sense by most of practitioners [14]. Therefore, the limitations and success factors in SPI reported – included human factors– in the scientific literature [20,22,23,47,48] was collected in order to get an understanding of these issues from a holistic perspective. The snowballing technique, literature review with a systematic approach and thematic content analysis based on human factors as units of analysis [49] was also used in this study. The content analysis technique was carried out to accomplish a comparative analysis among the SPI Manifesto [50], ISO 10018 [34], and three relevant studies which are concerned about the entire development process and deepen research on the human factor in SPI [23,51,52].

3.2. Design of the proposed framework

The second phase involved the design of Samay by using the characterization of development process for small software companies in addition to embrace simple, easy and practical techniques which are aimed at fostering SPI by considering the previously identified human factors. The following describes the three stages included in the design of the proposed framework.

3.2.1. Characterization of the software development process

When selecting the reference frameworks for this study, the following three considerations had to be satisfied: essential, agnostic and reach by consensus. Due to the nature of small software companies these considerations were adapted from a previous study done by one of the authors [53] .

- Firstly, the chosen reference framework had to be essential in nature, identifying and describing the basic possible range of human factors and/or software development related processes, insofar as is possible in the small software companies environment.
- Secondly, it is important that the reference framework should be independent of any specific software development approach.
- Thirdly, in order to maximize the reliability of the proposal, the chosen reference frameworks should be consensually agreed and generally accepted in the software development community as long as possible.

While a number of possible reference frameworks could be harnessed in order to conduct this study, after reviewing the frameworks [42], interviews [24] and human factors [54] from the first phase, no single one addresses the three considerations aforementioned more completely than ISO/IEC 29110 [30], Essence [35] and OMG ISO 10018 [34]. In consequence, all of them are the foundation of Samay and a brief description of each one is presented in earlier section. Finally, ISO 33014 [40] was included because of it provides a guide for process improvement, and it also provides guidance on how to strengthen and maintain the abilities, to ensure success with continual process improvement – ImprovAbility [38].

It is important to note that the design is grounded in the strategy used in the ArSPI model [55], which was adapted to meet the limitations of SPI initiatives mentioned before. It is centered more precisely on the artifacts that define the desired results more than in specific methods. Based on ISO/IEC 29110, the essential artifacts in the software development were chosen from the analysis of their purpose and the interviews conducted with three small companies [24]. These artifacts are five: Statement of Work, Requirements Specification, Project Plan, Software, Acceptance record. Nevertheless, other artifacts would be used if the software project requires them. Table 1 shows the selected artifacts of ISO/IEC 29110 and the artifacts included in the framework design: Proposal, Contract, Requirements, Plan, Software developed, Software tested, Software deployed, and Acceptance Record.

Table 1. List of key artifacts of the proposed framework based on the ISO / IEC 29110

ISO/IEC 29110 [30]	Samay
Statement of Work	Expectations Needs Context
Requirements Specification	Requirements
Project Plan	Plan
Software	Software developed Software tested Software deployed
Acceptance Record	Acceptance Record

Following the design strategy, the activities of each reference framework - ISO / IEC 29110, OMG Essence and Improvability - were analyzed in order to determine how they could support the creation of Samay’s artifacts. As a result, Table 2 shows an overview of the relationships between the reference frameworks and the proposed artifacts.

Table 2. Relationships between the reference frameworks – ISO / IEC 29110, OMG Essence and Improvability – and artifacts of Samay.

REFERENCE FRA-MEWORKS			ARTIFACTS OF SAMAY							
			Proposal	Contract	Plan	Requirements	Software developed	Software Tested	Software deployed	Acceptance Record
ISO/IEC 29110	Project Management Process	Project Planning			•	•				
		Project Plan Execution					•	•	•	
		Project Assessment and Control					•	•	•	
		Project Closure							•	•
	Software Implementation Process	Software Implementation Initiation			•		•			
		Software Requirements Analysis				•	•			
		Software Architectural and Detailed Design					•	•		
		Software Construction					•	•		
Software Integration and Tests							•			
Product Delivery							•			
OMG Essence	Customer	Opportunity	•	•						
		Stakeholders	•	•					•	
	Solution	Requirements			•	•	•	•	•	
		Software System				•	•	•	•	
	Endeavor	Work			•	•	•	•	•	
		Way-of-Working			•	•	•	•	•	
Team				•	•	•	•	•		
Improvability Model	Initiation	Sensing urgency	•							
		Idea processing		•						
	Projects	Project goal and requirements			•	•				
		Project team			•					
		Project competence and knowledge			•					
		Project process			•		•	•		
		Project prioritizing		•	•					
		Management support		•	•					
Involvement of others	•	•	•		•					

	In Use	Product quality				●		●		●
		Deployment strategy		●	●	●			●	
		Deployment means							●	
		Roles and responsibility							●	
		Operations and maintenance							●	
	Foundation	Expectation management	●	●	●	●	●			
		Knowledge management	●	●	●	●	●			
		Management competence					●			

From there, those activities that could be included in the proposal were identified and grouped under the artifacts then the phases of the proposed process were defined (see Table 3). Finally, the specific objective each of the seven phases was defined and they were used as inspiration to name the phases: *Opportunity*, *Negotiation*, *Requirements Analysis*, *Planning*, *Software Development*, *Software Testing* and *Software Deployment*. For instance, the opportunity phase has as inputs expectations, needs and context. This phase explores the needs and expectations of potential market to determine a business opportunity in response to the context, and its output is a proposal.

Table 3. Phases of proposed framework

Input	Phase	Objective	Output
Expectations Needs Context	Opportunity	<i>Exploring needs and expectations of potential market to determine a business opportunity in response to the context.</i>	Proposal
Proposal	Negotiation	<i>Reaching an agreement with sponsors to convert a proposal into a contract that addresses the context.</i>	Contract
Proposal Contract	Requirements Analysis	<i>Understanding the needs, aligning expectations and capturing the context in order to achieve a defined set of requirements with sufficient detail so that the software solution defined in the proposal and signed contract can be built by the development team.</i>	Requirements
Requirements Contract	Planning	<i>Creating a plan with sufficient detail to support the management of software development and testing. In addition, it should provide maintenance of required infrastructure for both development and testing.</i>	Plan
Requirements Plan	Software Development	<i>Managing the development team in order to actively cooperate for building software.</i>	Software (developed)
Requirements Software (developed)	Software Testing	<i>Finding and fixing software defects in early stages of software development. Hence, tests are defined from software requirements analysis and carried out as soon as possible.</i>	Software (tested)
Software (tested)	Software deployment	<i>Delivering software version tested to achieve acceptance that allows contract closure. Deployment could include implementation on a production environment.</i>	Software (deployed) Acceptance record

3.2.2. Identification, selection and integration of factors in the development process

Keeping in mind the results of comparative analysis, the following selection of the factors given in the ISO 10018 was chosen: *communication, awareness, engagement, networking, teamwork and collaboration, education and learning, attitude and motivation, recognition and rewards, empowerment, responsibility and authority*. Moreover, *personality* was included due to its proven importance [56,54]. Thus, a fundamental factor is *communication* because it can encourage *participation, collaboration and awareness*. It also is facilitating understanding and knowledge sharing - learning - which makes it easy for people to engage with their work. In addition, personality assessments were selected in order to meet the common features that can streamline communication and enhance cohesion and conflict resolution. Furthermore, other factors were chosen to achieve a positive change in *job satisfaction* are *motivation, recognition, reward, autonomy, empowerment, decision making and learning*.

Once again, snowballing approach was carried out in order to identify a set of techniques that might promote human factors and fit in the phases proposed. After that, potential artifacts of the techniques were studied. They were analyzed considering their purpose, easy to learn, easy to use, human factor and phase. Table 4 shows a set of techniques which are classified by human factor and phase. Some of them have their origin in the professional field and, therefore, are found in the gray literature. It should be noted that more than one technique has been selected for the same purpose, for example personality assessments include five techniques because their application depend on the context of the team and have

different approaches. On the other hand, several techniques were born in the agile community and others one were born in the innovation area; however they are not restricted to these areas and could be taken in the small software companies. That means small software companies do not need to adopt agile methodologies or be engaged in innovation activities.

Table 4. Techniques of proposed framework

PHASES	TECHNIQUES		FACTORS [34]															
			Communication	Awareness	Engagement	Networking	Collaboration	Learning	Attitude and Motivation	Recognition and Rewards	Empowerment	Responsibility and authority	Personality					
Opportunity	Meetings	Storytelling	•		•	•	•			•								
		Interviews	•			•	•											
		Focus Group	•		•	•	•											
		Brainstorming	•		•	•	•											
	Display	Satellite Map	•	•		•												
		Empathy Map	•	•		•				•								
		Trust Spider	•	•	•	•	•			•								
		Context Map	•	•	•	•	•											
		Graphic GamePlan	•	•	•	•	•		•									
		Franklin-Covey Opportunity Overview	•	•		•	•											
Negotiation	Mind Maps	Before negotiating	•	•						•								
		During negotiation	•	•	•	•	•			•								
		Troubleshooting negotiations	•	•					•	•								
		Double Bubble Map	•	•	•	•	•			•								
Requirements	Meetings	Observations	•															
		Workshops	•			•	•											
		Roleplay	•			•	•											
	Descriptions	Natural language	•			•												
		Scenarios	•			•												
		Use case	•			•												
		User stories	•		•	•												
			Storyboarding	•		•	•											
Criteria	INVEST		•										•					
	SMART		•										•					
Planning	Estimate	Unit of measure	•			•												
		Experts opinion		•		•		•										
		By analogy		•		•		•										
		Three-point or expected value		•		•		•										
		Planning Poker	•	•	•	•	•	•										
		Relative	•	•	•	•	•	•										
	Prioritize	In or Out	•		•	•												
		Eisenhower matrix	•		•	•	•											
		Three-level scale	•		•	•	•											
		MoSCoW	•		•	•	•											
		Cost of delay	•		•	•	•											
		Kano Analysis	•		•	•	•											
		Relative Weighting	•		•	•	•											
		Theme Scoring	•		•	•	•											
	Assign	Assigning O'Connell	•	•	•					•						•		
		Skills Matrix	•	•	•											•		
		Flow	•	•	•				•	•						•		
		By responsibilities	•	•	•			•								•		

Development	Measure	Metrics Ecosystem	•	•	•			•			•			
		Single objective	•	•	•			•			•			
	Meetings	One-on-Ones	•			•		•	•			•		
		Pairing	•			•	•	•	•			•		
		Personal Coaching	•		•	•		•	•			•		
	Feedback	Feedback wrap	•	•		•		•	•			•	•	
		Mind Map to monitor your skills	•	•		•		•	•			•	•	
		Start-Stop-Keep	•	•		•		•	•			•	•	
	Awareness	Personal Map	•	•		•			•					
		Motivational Balance	•	•		•								
		Moving Motivators	•	•	•	•	•		•					
		Trust Spider	•	•	•	•	•		•					
		1 to 1 Diagnosis	•	•		•	•		•					
	Personality	Myers-Briggs Type Indicator	•	•		•								•
		Big Five Factors of personality	•	•		•								•
		Sixteen Personality Factor	•	•		•								•
		Enneagram of Personality	•	•		•								•
		Keirsey temperament sorter II	•	•		•								•
	Extrinsic Motivation	Salary and Bonus	•						•	•				
		Benefits	•						•	•				
	Intrinsic Motivation	Kudo Box	•			•	•		•	•	•			
		Identity Symbols	•		•	•	•		•			•		
		Work Expo	•	•	•	•	•		•	•	•			
		Happiness door	•	•		•	•		•			•		
		Exploration days	•	•	•	•	•	•	•	•	•	•		
		Celebration Grid	•	•	•	•	•		•	•	•	•		
	Delegation	Delegation Poker	•			•	•							•
Delegation Map		•		•	•			•					•	
Delegation Board		•		•	•			•					•	
Testing	Level	Unit Testing	•		•			•						
		Integration testing	•		•			•						
		System Testing	•		•			•						
		Acceptance testing	•		•	•		•						
		Regression Testing	•		•			•						
	Black-box	Equivalence partitioning	•		•			•						
		Boundary value analysis	•		•			•						
		Decision tables	•		•			•						
		State transition	•		•			•						
		Use case	•		•			•						
	White-box	Statement coverage	•		•									
		Decision coverage	•		•									
		Other Structure-based techniques	•		•									
	Experience-based	•		•										
	Reviews	Informal Review	•		•	•	•	•				•		
		Walkthrough Review	•		•	•	•	•				•		
		Technical Review	•		•	•	•	•				•		
		Inspection	•		•	•	•	•				•		
	Deployment Strategy	Big Bang	•			•								
		Parallel application	•			•								
Phase introduction		•			•									
Experimental diffusion		•			•		•							
User based experiment		•			•		•				•			
The whole product	The core product	•			•									
	The whole product	•			•									
	The expanded product	•			•									
Role Model	•		•	•	•							•		

Although the set of techniques can seem extensive, it is neither exhaustive nor exclusive, so it is open to adaptation and integration of any techniques that each small company requires, as needed. There are a variety of them and have been

tested and are used in various organizations. Therefore, further constraints were: those could best meet with the limitations of small companies, and generate value through the introduction of micro-changes arise from the application of the techniques.

3.2.3. Identification and integration of elements to support the adoption of the proposed framework

When reviewing the limitations for small companies [22], the need to define additional elements was identified. These elements support the adoption of the proposed framework. In Table 5 can be seen that the phases of development process proposed partially cover the limitations observed. Therefore, two groups of elements were defined: complement and support (see Figure 1). First group is directly linked to the development process: *software development process flow, roles and responsibilities*, and *competencies and competency levels*. These elements are based on OMG Essence. Second group facilitates the adoption of the framework: *initiation and change management*. This group is inspired by ImprovAbility model – knowledge management and change strategy.

Table 5. Additional elements defined in response to the limitations observed

GROUP	LIMITATION [22]	SOLUTION	ELEMENT
Organization	There is lack of knowledge about the importance of the development process with product quality.	To incorporate an <i>initiation mechanism</i> that contains a vision of the importance of the development process and product quality. Bearing in mind, the limitation faced by small companies and human factors that impact these processes.	<i>Initiation</i>
Financial Resources	They depend on external support to implement a process improvement	The <i>mechanism of initiation</i> will include references to the standard 29110 and techniques: LAPPI and Pisko which could be implemented on its own within a process improvement initiative after Samay is implemented.	<i>Initiation</i>
Human Resources	Lack of roles definition, so employees perform several functions.	To establish definitions of <i>roles and responsibilities, and competencies and competency levels</i> within Samay.	<i>Roles and responsibilities, Competencies and competency levels</i>
	Lack of employees with knowledge about SPI methods and/or standards	The <i>mechanism of initiation</i> will include SPI Manifesto like an introduction of SPI and references to the most relevant SPI initiatives in small software companies.	<i>Initiation</i>
Process	They do not have defined processes.	To provide a <i>development process</i> and workable <i>process flows</i> .	<i>Software development process</i> <i>Software development process flow</i>
Models y Standards	They have a few or any experience in the adoption of SPI models and/or standards.	To incorporate a mechanism of <i>change management</i> in order to facilitate the adoption of the Samay which in turn allow understand how to change and learn new skills. That is, to gain an initial experience in this kind of process change.	<i>Change management</i>

A brief description of Samay to integrate human factors in the daily work of small software companies is presented in section 4, it is taking account the visual representation depicts in Figure 1. The core of Samay is represented by a triangle, the base of which indicates each phases of *software development process* (Table 3) and the top of which indicates its *techniques* (Table 4). The Complement of Samay is represented by three arrows and three circles concentric. The arrows suggest some possible *flow processes* depending on the sequence and time: linear, iterative, evolutionary and parallel. . Although the development process proposed could give the impression of a cascade development cycle, just as standard ISO / IEC 29110 does not intend to dictate the use of any particular life cycle [57]. The concentric circles display *roles* (Table 6), *competency levels* (Table 7) and *competencies* (Table 8). People must know their roles and responsibilities within the software development and understand what is expected of them [35]. Several roles may be played by a single person and one role may be assumed by several persons [30]. Without defining clear roles and responsibilities, the software development runs the risk that some of its important aspects may omitted or neglected – whether deliberately or unconsciously. Competencies are the abilities needed, that means the key capabilities required to carry out the work [35]. Each competency has three levels of achievement. The Support of Samay is represented by an ellipse and a rectangle. The

ellipse depicts the *change management* which aims to improve the success of change efforts. The rectangle shows the *initiation* element as the necessary knowledge for implementing Samay. It can be seen that each element has a unique representation.

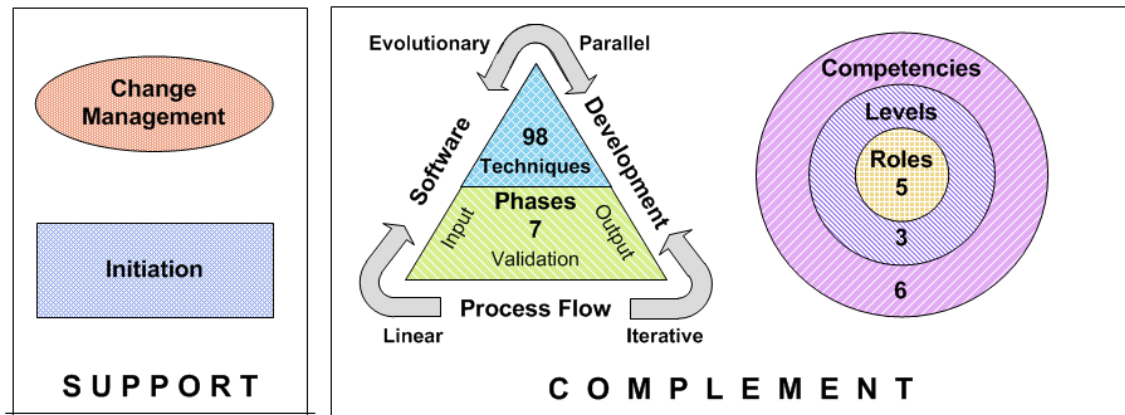


Fig. 1. Samay: support and complement elements

3.3. Validation

The third phase, validation, is ongoing and consists of two ways (see details in section 5): i) the framework proposed was validated by experts and ii) the framework proposed is being validated in a case study, their final results are not yet available but it is expecting that will give us empirical evidence to support its validity because the qualitative research (interviews and observation) done shows its pertinence.

4. A comprehensive framework to integrate human factors in the daily work of small software companies

The goal of this proposal is to outline a route towards the integration of SPI models in small software companies. The approach is centered in mechanisms that support human factors in the daily work, so that regardless of development methodology and project management they have a set of best practices to guide team members through the software development process and to work on the core activities. Figure 2 shows the overall design of Samay, the image on the left of the figure represents the core and complement elements after the *initiation* is implemented, The *initiation* is an element of knowledge that incorporates state of the art on SPI in the small companies and the description of Samay. In turn, the image on the right of the figure represents the same elements after the change management process. The *change management* facilitates an instance of Samay for a particular setting. Therefore, only the elements that respond to the urgency and vision will be implemented in order to gradually introduce changes following the principle “keep it simple” and ensure a short and continuous cycle. In Figure 2, the *software development process* is represented by a regular hexagon which is made up of triangles that indicates six of the seven phases proposed and the top of each triangle indicates its *techniques*. The *Opportunity phase* is different from the other six phases because it articulates the reason for the creation of the new or changed software. Therefore, it is represented by the small hexagonal figure within the larger one. The circle around the *Opportunity phase* depicts the *roles* Manager-Owner (M-O), Project Manager (PM), Technical Leader (TL), Software Developer (SD), Stakeholders (S). Likewise, two circles around the large hexagonal display the *competency levels*: Junior (1), Senior (2), Master (1) and the *competencies*: stakeholder representation, analysis, development, testing, leadership and management. The curved arrow next to the large circle symbolizes the *flow processes*. It is worth noting that the *roles* and *competencies* that facilitate the process execution through the *process flow* are integrated in order to stimulate the acquisition of strengths which lead to improve its ability to improve, as well as improving software process.

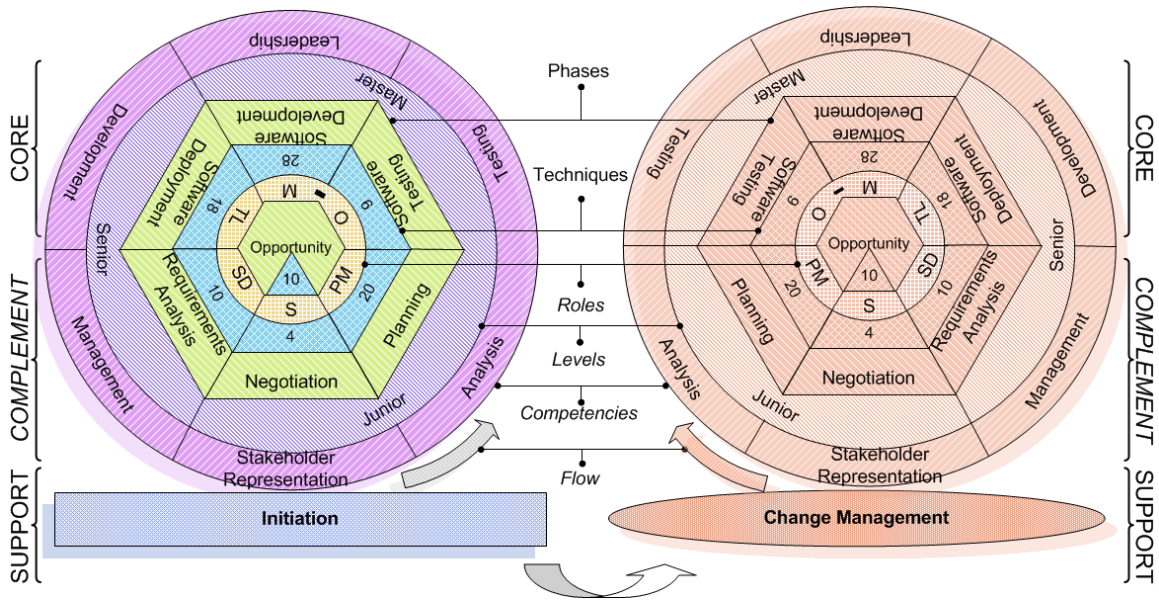


Fig. 2. Overall design of the proposed framework

Additionally, the *change management* is based on the model change management "Defying Doom" [58]. This model was chosen due to the fact that its author explains in a pragmatic and simple way, the main factors and tools we can use in the business world to tackle a difficult situation that requires transformation of the company to adapt it to a new reality. Although, this model was made for leading urgent large-scale transformations it was chosen due to its nature and approach. In fact, there are positive insights about its usefulness in a larger context – small companies, career and personal life–, since more than 30000 people have participated in the four editions of the MOOC (Massive Online Open Course) entitled "Leadership for Change"⁵ which is based on "Defying Doom". This model is grouped around three simple steps: What's the story?, Who is on board? and Getting things done (see Figure 3). The first step is getting people to understand what the story is. In other words, people achieve a shared understanding of the need for change by answering the questions where we are and how urgent it is for us to adapt and move – vision and urgency. The second step, once a basic level of awareness is achieved, is to convince each and every person that change really does start with them (i.e. personal implication). At this point, a board is used to group and indicate the names of each person in three areas: leader, top team and everyone. The key question is what can "you" the individual do to turn this around?. Once the story is clearly communicated, and the people are involved and aligned, the final step is execution and tracking the progress of the plan. The plan includes three parts: vision, objectives and the necessary actions to meet the objectives of the vision – things to start or stop doing.

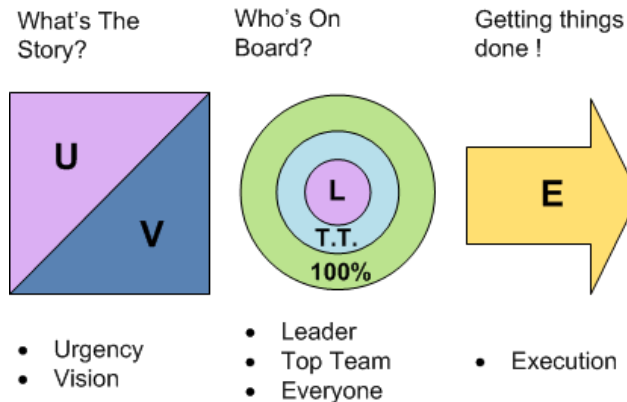


Fig. 3. "Defying Doom" model [58]

⁵ <https://miriadax.net/web/leadership-for-change-liderazgo-en-tiempos-de-cambio-4-edicion->

Table 6 shows a brief description of each of the roles and their main responsibilities: Manager-Owner (M-O), Project Manager (PM), Technical Leader (TL), Software Developer (SD), Stakeholders (S).

Table 6. Description of Roles

Role	Responsibilities
Manager-Owner (M-O)	<ul style="list-style-type: none"> Provides resources for the fulfillment of the objectives. Reviews and formally approves the performance of each phases of project development. Is able to keep the track and control of project development effort, solving contingencies that may arise during implementation. Promotes a positive work environment. Keeps integrity.
Project Manager (PM)	<ul style="list-style-type: none"> Supports and motivates the team through the coordination of team members' work activities and feedback. Encourages cooperation among stakeholders and between team members and stakeholders through the promotion and respect of the agreements between the parties. In conjunction with the person who plays the role of analyst, capture the context, understanding the needs of stakeholders and align the expectations in order to translate it into requirements. Estimate the effort and artifacts that arises during the development in order to put them into a clear work plan with a timetable and milestones. Leads the project and is able to keep the track and control of project by the review and assessment of results which derives in the closure of the project when it is completed. Additionally, Project Manager is responsible for the management of incidents and changes in planning.
Technical Leader (TL)	<ul style="list-style-type: none"> Supports Project Manager in the development of technical work. Coordinates training activities for team members for overcoming the deficiencies detected.
Software Developer (SD)	<ul style="list-style-type: none"> Performs technical work. That means requirements analysis, software design, build software base on design and testing, considering skills and levels of each role. In addition, software developers should maintain the infrastructure to ensure the availability of software development and testing environments. Learns how to cooperate with the team and stakeholders in order to achieve the project objectives at the individual and team level. Learns to do the technical work.
Stakeholders (S)	<ul style="list-style-type: none"> Stakeholder representation To embrace agreements To cooperate

Table 7 shows core competencies of each of the above roles which stakeholders should have to work in a software development project. It is highlighted the technical and general skills directly related to the development of software products, and basic management skills as mentioned in Essence [37]: Stakeholders Representation, Analysis, Development, Testing, Leadership and Management. Furthermore, Table 8 describes the generic competency levels: Level 1 (Junior), Level 2 (Senior) and Level 3(Master).

Table 7. Key Competencies

Competency	Description
Stakeholder Representation	<p>It encapsulates the ability to gather, communicate and balance the needs of other stakeholders, and accurately represent their views. Essential skills include:</p> <ul style="list-style-type: none"> Negotiation Facilitation Networking Good written and verbal communication skills Empathy
Analysis	<p>It encapsulates the ability to understand opportunities and their related stakeholder needs, and transform them into an agreed and consistent set of requirements. Essential skills include:</p> <ul style="list-style-type: none"> Verbal and written communication Ability to observe, understand, and record details

Development	<ul style="list-style-type: none"> • Agreement facilitation • Requirements capture • Ability to separate the whole into its component parts • Ability to see the whole by looking at what is required <p>It encapsulates the ability to design and program effective software systems following the standards and norms agreed by the team. Essential skills include:</p> <ul style="list-style-type: none"> • Knowledge of technology • Programming • Knowledge of programming languages • Critical thinking • Re-factoring • Design
Testing	<p>It encapsulates the ability to test a system, verifying that it is usable and that it meets the requirements. Essential skills include:</p> <ul style="list-style-type: none"> • Keen observation • Exploratory and destructive thinking • Inquisitive mind • Attention to detail
Leadership	<p>It enables a person to inspire and motivate a group of people to achieve a successful conclusion to their work and to meet their objectives. Essential skills include:</p> <ul style="list-style-type: none"> • Inspiration • Motivation • Negotiation • Communication • Decision making
Management	<p>It encapsulates the ability to coordinate, plan and track the work done by a team. Essential skills include:</p> <ul style="list-style-type: none"> • Communication • Administration • Organization • Resource planning • Financial reporting

Table 8. The Generic Competency Levels

Level	Description
Level 1 Junior	<p>Demonstrates a basic understanding of the concepts and can follow instructions. That means:</p> <ul style="list-style-type: none"> • Understands and conducts his or her self in a professional manner. • Is able to correctly respond to basic questions within his or her domain. • Is able to perform most basic functions within the domain. • Can follow instructions and complete basic tasks.
Level 2 Senior	<p>Able to apply the concepts in simple contexts by routinely applying the experience gained so far. That means:</p> <ul style="list-style-type: none"> • Is able to collaborate with others within the Team • Is able to satisfy routine demands and simple work requirements. • Can handle simple challenges with confidence. • Can handle simple work requirements but needs help in handling any complications or difficulties. • Is able to reason about the context and draw sensible conclusions.
Level 3 Master	<p>Able to apply the concepts in most contexts and has the experience to work without supervision. That means</p> <ul style="list-style-type: none"> • Is able to satisfy most demands and work requirements. • Is able to speak the domain language with ease and accuracy. • Is able to communicate and explain his or her work • Is able to give and receive constructive feedback • Knows the limits of his or her capability and when to call on more expert advice. • Works at a professional level with little or no guidance.

Finally, Samay has been conceived and designed to guide small software companies, formally constituted, in the acquisition of strengths that lead to the integration of SPI models. The activity of these companies is focused on the development of software products, which are not mission-critical or life support. It is also focused on development teams up to 10 people that are located in the same physical location. That means Samay is intended to support small companies whose employees are sharing the same physical workspace, and, therefore those teams that use offshore or outsourcing strategies are not considered.

5. Validation

The main aim of validation is to determine the theoretical validity of Samay. An expert review is a process asking the opinions, suggestions, feedback or comments from experts. Therefore, an expert review was designed and involved renowned experts in SPI to confirm that the proposed framework Samay could support small software companies in their daily work when management of people involvement and competence are needed for continual improvements. Therefore, the participants were purposefully selected due to their credibility, knowledge and experience in SPI (see Table 9). The criteria for selecting experts were as follows: Country of workplace (at least three different countries), education (PhD), years of experience in SPI (they had an average of 16 years' experience in SPI), Number of scientific publications (at least 20). In addition, one of them had taken part in the creation of a well-known processes model addressed for Small and Medium Enterprises (SMEs), and has experience in this topic. Group size theory varies in its suggestions regarding the ideal number of expert participants. Some general rules-of-thumb indicate five to 10 people or as many as possible, and no less than three, so that discrepancies among experts can be handled. However, given the difficulties in securing large numbers of experts to participate in such a study, we were satisfied that 5 would achieve an appropriate balance for the purposes of this study. All of experts accepted to participate in the validation and their confidence in reviewing the framework is quite familiar (near 4). Given their backgrounds it was anticipated that these experts would provide the most valuable information. Furthermore, three of them have published extensively on SPI in SMEs. The five participants were invited by e-mail to participate in the validation. Their workplaces are in North America (USA and Mexico) and Europe (Spain). The validation process was done on-line (electronic mail) with two attachments, questionnaire and framework. The participants responded to the six open questions listed in Table 10. The digital questionnaire makes it easy to analyze the data and read the comments compared to the non-digital questionnaire but does not help when the expert requires clarification of the questions or misunderstands some points. In addition, the study also used the unstructured interview to discuss with the experts about any unclear information (e.g. the framework, the questionnaire or the answers). The unstructured interview took the form of a free-flowing conversation.

Table 9. Background of experts

Background	Expert E1	Expert E2	Expert E3	Expert E4	Expert E5
Country of workplace	Spain	Mexico	Spain	Mexico	USA
Education	PhD.	PhD.	PhD.	PhD.	PhD.
Years of experience in SPI	14	20	20	8	20
Confidence [1 - 5] 1=not familiar, 5=very familiar	4	5	2	5	5

Table 10. Criteria for validation

Criteria	Questions
Objectives	In your opinion, do you consider appropriate the design goal of the proposed framework?
Methodology	What is your opinion about of the methodology used for design of the proposed framework?
Theoretical innovation	In your opinion, does the design framework represent any kind of theoretical innovation?
Pertinence	After the implementation of the framework, will the results be potentially applicable in small companies?
Areas for improvement	What aspects of the proposed framework (flow and phases of the software development process, techniques, roles and responsibilities, competences and levels of roles, initiation, and change management) could be improved?
Additional comments	Please, feel free to express yours opinions and thoughts about the proposed framework

The findings from the validation are categorized into six criteria (see Table 10) and are presented below. The answers are analyzed critically. This means that the responses of all the experts on a particular subject are read and understood. The decision on whether or not to accept the experts' suggestions is made according to the following rules:

1. If the experts disagree, the majority view is accepted.
2. If more than one expert proposes a change, the recommendation will be taken into account.
3. If only one expert recommends a change, it will be accepted if it is not due to a misinterpretation. But, it will be left for future research.

5.1. Objectives

The objectives of the framework were successfully validated by four experts as is evident in this example quotation from E1 “...*the objectives seem adequate and appropriate* ...”, in addition E3 claims “*building a framework that encourage the creation and strengthening of qualities which support the integration of SPI is very appropriate*” and E2 said “*Overall I like your goal, as I interpret it, but the wording could be improved* ...”. As a result of the comments from the last expert the wordings was revised in order to reduce misunderstandings and make the content clear and understandable without assuming knowledge. It should be mentioned that her concern is different from E5 because she is referring to the terms “success factors” and “SPI models”. Conversely, E5 point out that “*the proposed techniques make reference to their impact not in the adoption of software process improvement (SPI) but in the development process*” and continue “*...not that this is wrong, but the focus of the work is different than the proposed objective. I believe this objective, in the light of work done should be reworded*” and “*while this inconsistency between objective and the work done is not resolved, it is difficult to answer the other points [questions]*”. In the light of that, and considering also the insights of the assessment previously made by three expert professors in this area and, of course, the answers of the others four participants in this study, the authors decide to analyze the results of this validation as it. Therefore, four responses from participants were achieved.

5.2. Methodology

The methodology was successfully validated by four experts. One of them (E1) claims “*The defined methodology is interesting*” and suggests “*a minor improvement*”. Although, three of them (E2, E3, E4) point out that the methodology seems appropriate, E3 was concerned about the integration of human factors requirements after the framework was designed, but this is not really so, because human factors were kept in mind throughout the design process. Furthermore, E5 felt he need more detail in order to avoid misunderstandings and is able to respond.

5.3. Theoretical innovation

The theoretical innovation was successfully validated by four experts because the framework is taking into account human factors. As E2 put it more concisely “*Yes, the issue of human factor is well incorporated in the “classic” activities*”. However, E1 suggests “*...better highlights the delta [human factors]*” and E5 did answer this question.

5.4. Pertinence

In regard to pertinence, although none of the experts provided negative comments only three of them provided positive comments whereas the other two did not answer this question. Thus, E1 explains that “*... although it [the proposed framework] has not been implemented, it is reasonable to consider that the results can be applied to small companies...*”. Likewise, E3 point out that “*...it will be “easy” to implement the methodology because the actors know each other. The problem may be overloading the activities to be performed by roles...*”. This statement is also supported by E4, who added “*I believe that it [framework] can be potentially applicable if the micro and small and medium enterprises (MSMEs) have an coach and/or a support tool which assists them in the elaboration of the proposed diagrams [artifacts]...*”. On the contrary, E2 said that did understand the question and E5 did answer this question.

5.5. Areas for improvement

E1 and E5 were concerned about the implementation of the “Defying Doom” model in the small companies’ context but there is empirical evidence suggesting that its implementation is feasible. Moreover, E1 stated “*It could be interesting to identify which human factors influence or are more relevant*”. On one hand, E3 and E4 see the need to be aware of workload involved in each role. On the other hand, E1, E2 and E4 suggested some changes of the framework diagram which have been used to improve its content in order to gain a better understanding. For instance, E2 stated that “*in the diagram seem that the 7 phases are repeated*” and it is important to divide the process model and the implementation method, and consequently the diagram was reviewed. Furthermore, she was concerned about the amount of techniques –

information – they would have to first understand. It has seemed worthwhile to address these issues through further research.

5.6. Additional comments

There were comments from three participants. One of them (E4) was very enthusiastic and said that “*I found the proposed framework a very interesting work*” and “*...I would like to know more...* “. The comments made by E5 are presented in the objectives section. Finally, the recommended grammar/spelling/rewording changes, from E1, have been analyzed and the content was improved following the suggestions made.

5.7 Limitations

Regarding limitations, they are primarily linked to the sample size and quality. The group only consisted of five experts they have 82 person-years of practice and experience in this area. Indeed, it is very difficult to find experts with a great deal of experience and expertise in SPI and SMEs. It should be mentioned that one of them (E5) was the most critical of all and found difficult to answer the questions. For this reason an interview technique was used, it revealed that his viewpoint is quite different from the other viewpoints mentioned above. Therefore, four experts (62 person-years) took active part in this study. The current level of validation is limited. A case study is currently ongoing for an in-depth evaluation of Samay but its results are not available yet. Although a case study is an expensive method to conduct that takes a great deal of time to collect and analyze the data in a scientific way, the use of it is valuable because of its holistic approach. In this sense, one key concern regarding the case study is the lack of rigor but it is minimized following a protocol based on [59].

Despite the fact that every effort has been taken to ensure that the reference frameworks for the investigation are essential, fit-for-purpose and leading edge in their respective fields (and the expert feedback indicates that the methodology and objectives are appropriate), no framework can claim to be widely accepted. Therefore, the study is limited in that there is a possibility that some aspects of the software development process and human factors in small companies could have been overlooked. In fact, the set of techniques included in this framework could be modified or extended to fit the interests of a particular setting because it is neither exhaustive nor exclusive.

6. Conclusion and future work

The importance of this study is based on the importance of small software companies in the software industry. Human factors are important to achieve outcomes that are consistent and aligned with organizational strategies and values. Three main aims have been pursued in relation to the solution devised: i) to keep the lifecycle software process functionally agnostic to any methodology or method of software development, ii) to integrate human factors on a regular basis (daily) in order to enhance small companies’ ability to improve, and iii) to ensure the validity of the solution. Therefore, this paper presents an approach which integrates international standards in a comprehensive, yet practical, framework.

The proposed framework is designed as a core in order to allow the team members to adopt a comprehensive software development process that incorporates techniques that can be adapted to small companies’ needs. Samay is simple, flexible, and easy to implement so each element can be incorporated gradually and tailored according to the needs of small companies. In this sense, the techniques should be selected considering what every small company can do and project needs as there is more than one technique for the same purpose. The techniques are grouped in accordance with the seven phases of the software development process and provide concrete actions and a set of guidelines. An overview of this framework is presented in Figure 1 and 2.

The limitations lie in the difficulty of conducting research in small companies. In this study, the triangulation is a technique used to ensure the validity and credibility of the results. The methodological triangulation is based on theory of existing frameworks, expert review, and a case study. The results of the review by the four experts following the methodology explained in section 5 supported the view that Samay could support practitioners when small software companies want to start improving their ways of work. Furthermore, several minor changes, such as the wording and diagram, were made based on the suggestions by the experts. The case study is ongoing and their results are not yet available but are expected to be consistent with those achieved in this study. Future work will be to carry out more case studies and further research in order to address the improvement mentioned before.

Acknowledgements

The authors would like to express their thanks to the experts who participated in the validation of the framework with expertise and interest, and contributed to the improvement of the framework presented here.

References

- [1] R. Moll, Being prepared – A bird’s eye view of SMEs and risk management, *ISO Focus*. (2013).
- [2] Organization for Economic Co-operation and Development, *SME and Entrepreneurship Outlook 2005*, OECD PUBLISHING, Paris, 2005.
- [3] M.-L. Sánchez-Gordón, R.V. O’Connor, Understanding the Gap Between Software Process Practices and Actual Practice in Very Small Companies, *Softw. Qual. J.* 24 (2016) 549–570. doi:10.1007/s11219-015-9282-6.
- [4] UNCTAD, *Information economy report 2012: the software industry and developing countries*, United Nations Publications, New York, 2012.
- [5] M.C. Paulk, Using the software CMM in small organizations, in: *Proc. Pac. Northwest Softw. Qual. Conf. Eighth Int. Conf. Softw. Qual.*, Portland, Oregon, 1998: pp. 350–361.
- [6] G. Coleman, R.V. O’Connor, An investigation into software development process formation in software start-ups, *J. Enterp. Inf. Manag.* 21 (2008) 633–648.
- [7] N. Baddoo, T. Hall, De-motivators for software process improvement: an analysis of practitioners’ views, *J. Syst. Softw.* 66 (2003) 23–33. doi:10.1016/S0164-1212(02)00060-2.
- [8] S. Basri, R.V. O’Connor, Understanding the perception of very small software companies towards the adoption of process standards, in: A. Riel, R.V. O’Connor, S. Tichkiewitch, R. Messnarz (Eds.), *Syst. Softw. Serv. Process Improv.*, Springer, 2010: pp. 153–164.
- [9] C. Laporte, R. O’Connor, G. Fanmuy, International systems and software engineering standards for very small entities, *CrossTalk J. Def. Softw. Eng.* 26 (2013) 28–33.
- [10] L. Ibrahim, E. Wallmüller, W. Daschner, Using Enterprise SPICE in Very Small Entities, in: P.M. Clarke, R.V. O’Connor, T. Rout, A. Dorling (Eds.), *Softw. Process Improv. Capab. Determ.*, Springer International Publishing, 2016: pp. 423–430. http://link.springer.com/chapter/10.1007/978-3-319-38980-6_31.
- [11] S. Jeners, P. Clarke, R.V. O’Connor, L. Buglione, M. Lepmets, Harmonizing Software Development Processes with Software Development Settings – A Systematic Approach, in: F. McCaffery, R.V. O’Connor, R. Messnarz (Eds.), *Syst. Softw. Serv. Process Improv.*, Springer-Verlag, 2013: pp. 167–178. http://link.springer.com/chapter/10.1007/978-3-642-39179-8_15.
- [12] F. McCaffery, G. Coleman, Lightweight SPI assessments: what is the real cost?, *Softw. Process Improv. Pract.* 14 (2009) 271–278. doi:10.1002/spip.430.
- [13] P. Clarke, R.V. O’Connor, The situational factors that affect the software development process: Towards a comprehensive reference framework, *Inf. Softw. Technol.* 54 (2012) 433–447. doi:10.1016/j.infsof.2011.12.003.
- [14] L.F. Capretz, Bringing the Human Factor to Software Engineering, *IEEE Softw.* 31 (2014) 104–104. doi:10.1109/MS.2014.30.
- [15] A. Sampaio, I.B. Sampaio, E. Gray, The need of a person oriented approach to software process assessment, in: *2013 6th Int. Workshop Coop. Hum. Asp. Softw. Eng. CHASE*, 2013: pp. 145–148. doi:10.1109/CHASE.2013.6614752.
- [16] A. Fuggetta, E. Di Nitto, Software Process, in: *Proc. Future Softw. Eng.*, ACM, New York, NY, USA, 2014: pp. 1–12. doi:10.1145/2593882.2593883.
- [17] G. Coleman, R. O’Connor, Software Process in Practice: A Grounded Theory of the Irish Software Industry, in: I. Richardson, P. Runeson, R. Messnarz (Eds.), *EuroSPI 2006*, Springer-Verlag, 2006: pp. 28–39.
- [18] C.Y. Laporte, S. Alexandre, R.V. O’Connor, A Software Engineering Lifecycle Standard for Very Small Enterprises, in: R. O’Connor, N. Baddoo, K. Smolander, R. Messnarz (Eds.), *Proc. EuroSPI*, Springer, Heidelberg, 2008: pp. 129–141. doi:10.1007/978-3-540-85936-9_12.
- [19] P.G. Armour, The spiritual life of projects, *Commun. ACM.* 45 (2002). doi:10.1145/502269.502280.
- [20] X. Larrucea, R.V. O’Connor, R. Colomo-Palacios, C.Y. Laporte, Software Process Improvement in Very Small Organizations, *IEEE Softw.* 33 (2016) 85–89. doi:10.1109/MS.2016.42.
- [21] M.-L. Sanchez-Gordon, R.V. O’Connor, R. Colomo-Palacios, Evaluating VSEs Viewpoint and Sentiment Towards the ISO/IEC 29110 Standard: A Two Country Grounded Theory Study, in: *Softw. Process Improv. Capab. Determ.*, Springer-Verlag, 2015.
- [22] C. Valtierra, M. Muñoz, J. Mejia, Characterization of Software Processes Improvement Needs in SMEs, in: *IEEE*, 2013: pp. 223–228. doi:10.1109/ICMEAE.2013.33.
- [23] S. Bayona, J.A. Calvo-Manzano, T. San Feliu, Critical Success Factors in Software Process Improvement: A Systematic Review, in: *Softw. Process Improv. Capab. Determ.*, Springer, 2012: pp. 1–12. http://link.springer.com/chapter/10.1007/978-3-642-30439-2_1.
- [24] M.-L. Sánchez-Gordón, R.V. O’Connor, Understanding the gap between software process practices and actual practice in very small companies, *Softw. Qual. J.* (2015). doi:10.1007/s11219-015-9282-6.
- [25] X. Larrucea, I. Santamaría, R. Colomo-Palacios, Assessing ISO/IEC29110 by means of ITMark: results from an experience factory, *J. Softw. Evol. Process.* (2016) n/a-n/a. doi:10.1002/smr.1795.
- [26] P. Clarke, R.V. O’Connor, Business success in software SMEs: Recommendations for future SPI studies, in: D. Winkler, R.V. O’Connor, R. Messnarz (Eds.), *EuroSPI 2012*, Springer, Heidelberg, 2012: pp. 1–12.
- [27] P. Clarke, R.V. O’Connor, The influence of SPI on business success in software SMEs: An empirical study, *J. Syst. Softw.* 85 (2012) 2356–2367.
- [28] R.V. O’Connor, C.Y. Laporte, Software Project Management in Very Small Entities with ISO/IEC 29110, in: D. Winkler, R.V. O’Connor, R. Messnarz (Eds.), *Syst. Softw. Serv. Process Improv. EuroSPI 2012*, Springer-Verlag, 2012: pp. 330–341. http://dx.doi.org/10.1007/978-3-642-31199-4_29.

- [29] M. Takeuchi, N. Kohtake, S. Shirasaka, Y. Koishi, K. Shioya, Report on an assessment experience based on ISO/IEC 29110, *J. Softw. Evol. Process.* 26 (2014) 306–312. doi:10.1002/smr.1591.
- [30] ISO, Software engineering – Lifecycle profiles for Very Small Entities (VSEs) Part 5-1-2: Management and engineering guide: Generic profile group: Basic Profile, Geneva, 2011.
- [31] E. Moreno-Campos, M.-L. Sanchez-Gordon, R. Colomo-Palacios, A. Amescua Seco, Towards Measuring the Impact of the ISO/IEC 29110 Standard: A Systematic Review, in: *Proc. 21st EuroSPI 2014 Conf.*, Springer-Verlag, Luxembourg, 2014: pp. 1–12. doi:10.1007/978-3-662-43896-1_1.
- [32] H. Oktaba, F. Garcia, M. Piattini, F. Ruiz, F.J. Pino, C. Alquicira, Software Process Improvement: The Competisoft Project, *Computer.* 40 (2007) 21–28. doi:10.1109/MC.2007.361.
- [33] C.Y. Laporte, R.V. O’Connor, L.H.G. Paucar, The Implementation of ISO/IEC 29110 Software Engineering Standards and Guides in Very Small Entities, in: L.A. Maciaszek, J. Filipe (Eds.), *Eval. Nov. Approaches Softw. Eng.*, Springer International Publishing, 2015: pp. 162–179. http://link.springer.com/chapter/10.1007/978-3-319-30243-0_9.
- [34] ISO, ISO 10018 Quality Management - Guidelines on People Involvement and Competence, Geneva, 2012.
- [35] OMG, Essence – Kernel and Language for Software Engineering Methods, 2015.
- [36] M.E. Morales-Trujillo, H. Oktaba, M. Piattini, The making of an OMG standard, *Comput. Stand. Interfaces.* 42 (2015) 84–94. doi:10.1016/j.csi.2015.05.001.
- [37] I. Jacobson, P.-W. Ng, P.E. McMahon, I. Spence, S. Lidman, *The essence of software engineering: applying the SEMAT kernel*, Addison-Wesley, Upper Saddle River, NJ, 2013.
- [38] J. Pries-Heje, J. Johansen, eds., *ImprovAbility - success with process improvement*, 2. edition, DELTA, Hørsholm, 2013.
- [39] M. Kuhrmann, D. Méndez Fernández, From pragmatic to systematic software process improvement: an evaluated approach, *IET Softw.* 9 (2015) 157–165. doi:10.1049/iet-sen.2014.0190.
- [40] ISO, TECHNICAL REPORT ISO/IEC TR 33014:2013 Information technology — Process assessment — Guide for process improvement, Geneva, 2013.
- [41] M.-L. Sánchez-Gordón, R. Colomo-Palacios, A. de A. Seco, R.V. O’Connor, The Route to Software Process Improvement in Small- and Medium-Sized Enterprises, in: M. Kuhrmann, J. Münch, I. Richardson, A. Rausch, H. Zhang (Eds.), *Manag. Softw. Process Evol.*, Springer International Publishing, 2016: pp. 109–136. http://link.springer.com/chapter/10.1007/978-3-319-31545-4_7.
- [42] B. Kitchenham, S. Charters, *Guidelines for performing Systematic Literature Reviews in Software Engineering*, School of Computer Science and Mathematics, Keele University, 2007.
- [43] C.Y. Laporte, A. Renault, S. Alexandre, The Application of International Software Engineering Standards in Very Small Enterprises, in: H. Oktaba, M. Piattini (Eds.), *Softw. Process Improv. Small Medium Enterp. Tech. Case Stud.*, Information Science Reference, Hershey, New York, 2008: pp. 42–70.
- [44] D. Mishra, A. Mishra, Software process improvement in SMEs: A comparative view, *Comput. Sci. Inf. Syst.* 6 (2009) 111–140. doi:10.2298/CSIS0901111M.
- [45] M. Sulayman, E. Mendes, An extended systematic review of software process Improvement in small and medium web companies, in: *Eval. Assess. Softw. Eng. EASE 2011 15th Annu. Conf. On, IET, 2011*: pp. 134–143. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6083172.
- [46] F.J. Pino, F. García, M. Piattini, Software process improvement in small and medium software enterprises: a systematic review, *Softw. Qual. Control J.* 16 (2008) 237–261. doi:10.1007/s11219-007-9038-z.
- [47] M. Munoz, J. Mejia, J.A. Calvo-Manzano, G. Cuevas, T. San Feliu, A. De Amescua, Expected Requirements in Support Tools for Software Process Improvement in SMEs, in: *Electron. Robot. Automot. Mech. Conf. CERMA 2012 IEEE Ninth*, 2012: pp. 135–140. doi:10.1109/CERMA.2012.29.
- [48] M. Sulayman, E. Mendes, C. Urquhart, M. Riaz, E. Tempero, Towards a theoretical framework of SPI success factors for small and medium web companies, *Inf. Softw. Technol.* 56 (2014) 807–820. doi:10.1016/j.infsof.2014.02.006.
- [49] M. Yilmaz, R.V. Connor, P. Clarke, A Systematic Approach to the Comparison of Roles in the Software Development Processes, in: *7th Int. Workshop Soc. Softw. Eng. SSE 2015 Proc. Sept. 1 2015 Bergamo Italy*, 2015. <http://dl.acm.org/citation.cfm?id=2804381>.
- [50] J. Pries-Heje, J. Johansen, SPI Manifesto, (2010). http://www.iscn.com/Images/SPI_Manifesto_A.1.2.2010.pdf.
- [51] G. Matturro, J. Saavedra, Considering People CMM for managing factors that affect software process improvement, *Lat. Am. Trans. IEEE Rev. IEEE Am. Lat.* 10 (2012) 1603–1615. doi:10.1109/TLA.2012.6187605.
- [52] D. Viana, T. Conte, D. Vilela, C.R.B. de Souza, G. Santos, R. Prikładnicki, The influence of human aspects on software process improvement: Qualitative research findings and comparison to previous studies, in: *16th Int. Conf. Eval. Assess. Softw. Eng. EASE 2012*, 2012: pp. 121–125. doi:10.1049/ic.2012.0015.
- [53] P. Clarke, R.V. O’Connor, An empirical examination of the extent of software process improvement in software SMEs, *J. Softw. Evol. Process.* (2013). <http://onlinelibrary.wiley.com/doi/10.1002/smr.1580/full>.
- [54] M.-L. Sanchez-Gordon, Getting the Best out of People in Small Software Companies: ISO/IEC 29110 and ISO 10018 Standards, *Int. J. Inf. Technol. Syst. Approach.* 10 (2017) 45–60. doi:10.4018/IJITSA.2017010103.

- [55] M. Kuhrmann, S. Beecham, Artifact-based Software Process Improvement and Management: A Method Proposal, in: Proc. 2014 Int. Conf. Softw. Syst. Process, ACM, New York, NY, USA, 2014: pp. 119–123. doi:10.1145/2600821.2600839.
- [56] P. Lenberg, R. Feldt, L.G. Wallgren, Behavioral software engineering: A definition and systematic literature review, *J. Syst. Softw.* 107 (2015) 15–37. doi:10.1016/j.jss.2015.04.084.
- [57] C.Y. Laporte, R.V. O’Connor, Systems and Software Engineering Standards for Very Small Entities: Accomplishments and Overview, *Computer*. 49 (2016) 84–87. doi:10.1109/MC.2016.242.
- [58] B. Quinn, *Defying Doom: Leading Urgent Large-Scale Transformations*, LID Publishing, S.l., 2016.
- [59] P. Runeson, M. Host, A. Rainer, B. Regnell, *Case Study Research in Software Engineering: Guidelines and Examples*, John Wiley & Sons, 2012.