Software Quality Journal (2021) 29:239–273
https://doi.org/10.1007/s11219-021-09552-3

Reconciliation of scrum and the project management process of the ISO/IEC 29110 standard-Entry profile—an experimental evaluation through usability measures

Sergio Galvan-Cruz[1] · Manuel Mora[1] · Claude Y. Laporte[2] · Hector Duran-Limon[3]

Accepted: 10 March 2021 / Published online: 3 April 2021

Abstract

Software process standards and models are used in large- and medium-sized organizations to reach the *Iron Triangle*. In contrast, small and very small entities either ignore them or cannot apply them because these standards and models are technically and economically not affordable. Consequently, agile software development practices are usually used by small and very small organizations. The ISO/IEC 29110 series of standards and guides are now available for very small organizations, but their utilization with agile practices represents an agility-rigor reconciliation problem. In this research, we report the experimental evaluation of Scrum + EPG (a reconciled agile-rigorous software Project Management process from Scrum, and the Project Management process of the ISO/IEC 29110 series-Entry profile, documented in an Electronic Process Guide). Scrum + EPG was compared to Scrum EPG (a non-modified Scrum process also documented in an Electronic Process Guide). Thirty-two international academicians and practitioners, including experts and novices on agile practices, from Latin America, North America, and Asia–Pacific regions, evaluated six metrics of usability. A within-subjects design and Wilcoxon matched-pairs signed-rank tests were applied for collecting and analyzing the experimental data. The statistical results support the claim that the Scrum + EPG was considered a high-quality conciliated agile-rigorous software Project Management process for the Entry profile. Given the scarcity of similar studies and the need for reconciling agile-rigorous software development practices, this study contributes to a plausible solution for very small organizations. Finally, further empirical research is encouraged to confirm, update, and extend the results reported in this investigation.

Keywords Agility-rigor reconciliation problem · Scrum · ISO/IEC 29110 standard-Entry profile · Project Management process · Electronic Process Guide · Within-subjects experiment design

✉ Manuel Mora
jose.mora@edu.uaa.mx

[1] Autonomous University of Aguascalientes, Ave. Universidad, 940, 20131 Aguascalientes, Ags, Mexico

[2] École de Technologie Supérieure, 1100 Notre Dame Street, Montreal, QC H3C 1K3, Canada

[3] University of Guadalajara, Perif. Nte. 799, Villa de los Belenes, 45130 Zapopan, Jal, Mexico

1 Introduction

The permanent aim of software development organizations—of any size—is the delivering of software products/services on the agreed schedule, within the budget, and with the planned quality of functionalities and performance issues (i.e., the called *Iron Triangle* project management success criterion) (Agarwal & Rathod, 2006; Slaughter et al., 1998). Whereas additional project management success metrics have been proposed regarding achieving positive impacts on the business and organizational goals attained to the software project (Savolainen et al., 2012), the *Iron Triangle* success criterion is still valid and endorsed by software process standards and software process best practices (CMMI® Institute, 2019; ISO/IEC/IEEE, 2017).

However, to achieve the *Iron Triangle* success criterion is not considered a straightforward path to be followed by software development organizations. The academic (Marques et al., 2017; Savolainen et al., 2012) and professional literature (Bloch et al., 2012; Standish Group, 2015) reported the occurrence of failed software development projects in large- and medium-sized organizations. A failed software development project can be defined as a project that exceeded its cost, and/or had a significant schedule overrun, and/or was released with a quality shortfall (Agarwal & Rathod, 2006; Slaughter et al., 1998).

The discipline of software engineering, to counterattack these software development project failures, has developed software process standards and models (Niazi, 2015; Niazi et al., 2005; Unterkalmsteiner et al., 2011). These software process standards such as the ISO/IEC/IEEE 12207, 2017 (ISO/IEC/IEEE, 2017), and the ISO/IEC 33004, 2015 (ISO/IEC, 2015), are available for large- and medium-sized organizations. Similarly, software process models exist as the CMMI-DEV (CMMI, 2019) and the Team Software Process (TSP) (Humphrey, 2000). The correct implementation and utilization of these software process standards and models have generated relevant benefits for the organization, the customers and users, the development team and software development process, and lately for the released software product/service (Ebert, 2007; Humphrey, 2000; Pai et al., 2015; Unterkalmsteiner et al., 2011). Examples of the main benefits are project cost reduction, higher quality products, variance reduction on the project schedule, high user satisfaction, reduction of wasted organizational resources, and a more stable and predictable software development process (Ebert, 2007; Humphrey, 2000; Pai et al., 2015; Unterkalmsteiner et al., 2011). Consequently, the utilization of these software process standards and models is a common practice in large- and medium-sized organizations.

For very small organizations, this situation is a big challenge given the structural economic and organizational differences between the large (i.e., larger than 250 employees) and medium-sized (i.e., between 50 and 249 employees) organizations and the small (i.e., between 10 and 49 employees) and very small ones (i.e., up to 25 people) (Laporte et al., 2013a; Laporte et al., 2013b; Richardson & von Wangenheim, 2007). This kind of organization is characterized by having very limited budgets, less technical and managerial expertise, lack of interest in using heavy-process software standards and models, a highly dynamic and informal organizational culture, and pressures for fast delivery from customers (Clarke & O´Connor, 2013; Coleman & O'Connor, 2008; O'Connor & Coleman, 2009; Staples et al., 2007). Thus, these organizations either ignore or reject their utilization because they were designed by and for large- and medium-sized organizations, and because they demand a large project budget and introduce process bureaucracy issues and excessive documentation, large periods for releasing the product, and unnecessary project meetings (Clarke & O´Connor, 2013; Coleman & O'Connor, 2008; O'Connor &

Coleman, 2009; Staples et al., 2007). Consequently, these small and very small organizations fail to achieve the *Iron Triangle* project management success criterion, and lately, their software customers are also negatively impacted (Clarke & O´Connor, 2013; O'Connor & Coleman, 2009). Recently, a new series of software process standards and guides specifically designed for very small organizations has been released, i.e., the ISO/IEC 29110 series (ISO/IEC, 2012). This series of standards and guides are mainly composed of a Project Management process and a Software Implementation process. This standard defines the minimum but essential roles, activities, tasks, and work products that require very small organizations (Takeuchi et al., 2014) for pursuing the completion of software development projects on time, within budget, and releasing the expected quality. Initial studies (Laporte & O'Connor, 2017; Larrucea & Fernandez-Gauna, 2019; Majchrowski et al., 2016) have collected evidence of positive impacts for the organization, the customers and users, the development team and process, and lately for the software product/service released in very small organizations, similar to the received for large- and medium-sized ones by using the heavier process standards and models.

Furthermore, it has been reported that small and very small organizations prefer agile software development practices (Ahimbisibwe et al., 2017; Klotins et al., 2019; Pino et al., 2010) rather than rigor-oriented ones (i.e., plan-driven methods (Boehm & Turner, 2003, 2004)) as those defined by the ISO/IEC standards such as the ISO/IEC/IEEE 12,207. Pino et al. (2010), based on previous core literature on Software Process Improvement efforts for small and very small software development organizations, concluded that they promote inherently agile-alike practices (constant face-to-face verbal communication rather than written documents, flexible, dynamic, and lightweight managerial practices, and flat organizational structures). Consequently, they guided two small software development companies in Latin America to deploy a Scrum-based lightweight process and they reported positive improvements on several process/practice capability levels. Klotins et al. (2019; 71), similarly found in their research on 88 experience reports from software development startup organizations, that they inherently apply many agile practices as "iterative development, empowered small team, and ongoing planning." Both studies (Klotins et al., 2019; Pino et al., 2010) did not found the utilization of rigor-oriented methods. Ahimbisibwe et al. (2017) found in a survey of 471 agile projects that the project size is negatively associated with software process success, and thus small size projects, given that are usually sold by small and very small software development organizations, use agile practices.

This preference is motivated by the high flexibility and customization promoted by the agile software development practices, regarding the usual mandatory practices provided by the ISO/IEC standards. Hence, the very small organizations interested in combining the utilization of agile practices, mainly Scrum (Schwaber & Sutherland, 2017) and/or XP (Beck, 1999; Beck & Andres, 2004), and the best-disciplined ones from the ISO/IEC 29110 standards, face a reconciliation problem between two different software development approaches (Boehm & Turner, 2003, 2004; 2005; Magdaleno et al., 2012; Silva et al., 2015).

In this research, we address this problem and report the experimental evaluation of Scrum + EPG—a reconciled agile-rigorous software Project Management process from Scrum, and the Project Management process of the Entry profile of the ISO/IEC 29110 series, documented in an Electronic Process Guide (EPG)—versus the public available Scrum EPG designed by the EPF Eclipse Foundation (2019)—a non-modified Scrum process documented also in an EPG.

The remainder of this article continues as follows. Section 2 reports the background on the ISO/IEC 29110 series, the Scrum process, and the related studies on agile-rigorous software process reconciliations. Section 3 describes the planned experiment, while Sect. 4 presents the experimental results, their discussion, and their theoretical and practical implications. Section 5 explains the threats to internal, construct, external, and conclusion validity. Finally, Sect. 6 presents conclusions and recommendations for future work.

2 Background and related work

2.1 An overview of the ISO/IEC 29110 standard-Entry profile

The ISO/IEC 29110 series provides software and systems engineering processes to very small entities (VSEs) for improving their product quality as well as their process performance (ISO/IEC, 2012; Laporte & O'Connor, 2017). A VSE is defined as an enterprise, an organization, a department, or a project having up to 25 people (ISO/IEC, 2012). These series of standards and guides have been proposed to have several profile groups (Laporte et al., 2013b). A profile is a subset of selected standards necessary to accomplish a function. A profile group is a collection of profiles that are related to the composition of processes. At present, there is a Generic profile group targeted at "VSEs that do not develop critical products" (ISO/IEC, 2012; p. 7). This Generic profile group is not targeted for a domain of applications. The Generic profile group contains four software engineering and four systems engineering profiles: Entry, Basic, Intermediate, and Advanced. Systems are typically composed of hardware and software components, in the context of the ISO/IEC 29110 series. The Entry profile targets start-up VSEs on small 6-person-month effort projects. The Basic profile targets single project teams at VSEs. The Intermediate profile targets VSEs with multiple team projects. Finally, the Advanced profile targets VSEs seeking to be an independent business competitive organization (Laporte et al., 2013a). In this research, we used the Project Management process of the Entry profile of the ISO/IEC 29110 series.

This ISO/IEC 29110 series for software developers has, as its foundation, two processes for the four software profiles:

- Project Management
- Software Implementation

This ISO/IEC 29110 standard claims to provide a disciplined Project Management process to VSEs for pursuing essential project control and visibility issues, and a systematic Software Implementation process for pursuing the satisfaction of the consumer needs through the delivery of high-quality software products/services (ISO/IEC, 2012). The Project Management process supervises the planning and execution of the technical activities carried out in the Software Implementation process to comply with the project's objectives in the expected quality, time, and cost objectives (ISO/IEC, 2012). This process includes the following activities:

- Project Planning
- Project Plan Execution
- Project Assessment and Control
- Project Closure

The Software Implementation process defines the technical activities to be systematically followed for developing the software and documentation according to the agreed requirements (ISO/IEC, 2012). This process includes the following activities:

- Initiation
- Software Requirements Analysis
- Software Component Identification
- Software Construction
- Software Integration and Tests
- Product Delivery

Figure 1 portrays a diagram of the Project Management process of the ISO/IEC 29110 standard-Entry profile- (adapted from ISO/IEC, 2012). The left side of Fig. 1 shows its relationship with the Software Implementation process. The right side of Fig. 1 shows its four activities that are labeled as Project Planning, Project Plan Execution, Project Assessment and Control, and Project Closure. A summarized but substantial description of these four Project Management activities is reported below. Full descriptions of each of the four activities can be found in other sources (Buchalcevova, 2019; ISO/IEC, 2012).

The Project Management process of the ISO/IEC 29110 standard-Entry profile defines three roles (these are not shown in Fig. 1):

- Customer
- Project Manager
- Work Team

The Customer role refers to a person or group of persons who know the customer domain process and requirements, and the authority to make decisions on the requirements, and changes, and the delivered product. The Project Manager role refers to the administrative person responsible for the project who has attributes of leadership, supervision of personnel, and financial and software development knowledge and experience. The Work

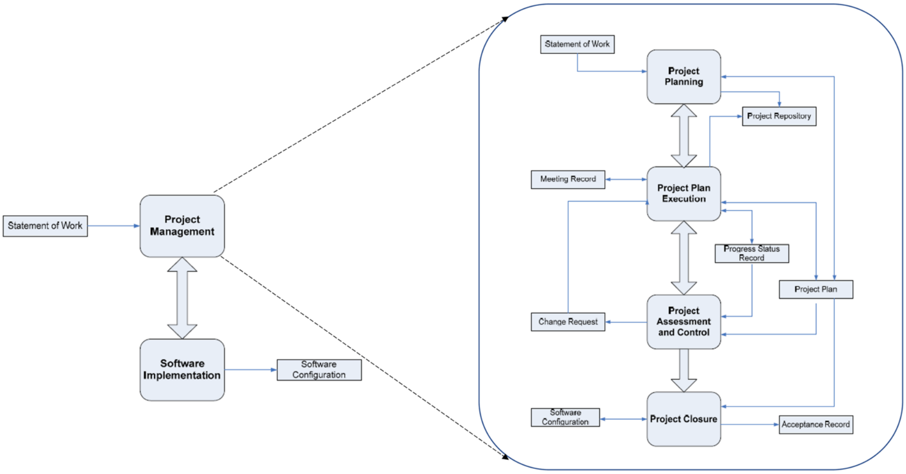


Fig. 1 Diagram of the Project Management process of the ISO/IEC 29110 standard-Entry profile

Team role refers to the software development technical people responsible for building the expected software product. The ISO/IEC 29110 standard does not define a set of specific Work Team sub-roles; instead, the following sub-roles can be derived from the activities of the Software Implementation process: Software Requirements Architect/Engineer, Software Developer, Software Tester, and Software Documentation Writer. In the ISO/IEC 29110 standard, a person can play more than one role.

The Project Management process of the Entry profile requires, for each task, at least one input Work Product and generates at least one output Work Product.

- The first input Work Product, usually provided by the Customer role, is a Statement of Work. The Work Products internally generated by the Work Team are Project Plan, Change Requests, Meeting Record, Progress Status Records, and Project Repository.
- The output work products are Acceptance Record and Software Configuration. Change Requests can also be generated by the Customer.

The first Project Planning activity "documents the planning details needed to manage the project" (ISO/IEC, 2012; p. 9). This first activity provides a reviewed Statement of Work that provides the description of work to be done, a project plan describing quality assurance actions, work team, and customer responsibilities, required project resources, effort, cost and schedule estimations, project risks, and the project repository. The second Project Plan Execution activity "implements the documented plan on the project" (ISO/IEC, 2012; p.11). This second activity provides the execution of control project monitoring, the status of the project plan, change requests tracking, and reviews/agreements with the customer. The third Project Assessment and Control activity "evaluates the performance of the plan" (ISO/IEC, 2012; p.12). This third activity provides current versus target plan performance/progress evaluation, change request tracking, and problem–solution tracking. Finally, the fourth Project Closure activity "provides the project's documentation and products in accordance with contract requirements" (ISO/IEC, 2012; p.12). This fourth activity provides support for the Customer's product acceptance, the Acceptance Record with the completion of the project, and the Software Configuration stored in the Project Repository.

The Project Management process of the ISO/IEC 29110 standard-Entry profile pursues seven objectives labeled from PM.O1 to PM.O7.

- The PM.O1 objective states that a project plan (including a Statement of the Work, efforts, tasks, and customer authorization) is developed.
- The PM.O2 objective states that the progress of the project is tracked until the project closure with a Customer's Acceptance Record.
- The PM.O3 objective states that the requests for changes are tracked.
- The PM.O4 objective states that review meetings and agreements between the Customer and the Work Team are documented.
- The PM.O5 objective states that risks are managed.
- The PM.O6 objective states that Software Configuration items are tracked.
- The PM.O7 objective states that software quality assurance actions are executed.

The Software Implementation process is related to the Project Management process because the former process is controlled by the latter. The Software Implementation process requires two inputs generated by the Project Management process which are Project Plan and Project Repository. The Software Implementation process produces a single output that is Software Configuration (which includes all software technical documentation and the executable software) and elaborates four internal products (Software Component Identification, Test Cases, and Test Procedures, Software Components, and a Test Report). The Software Implementation process includes six activities and uses the same three roles of the Project Management process:

- Customer
- Project Manager
- Work Team

Hence, the Project Management and Software Implementation processes of the ISO/IEC 29110 standard-Entry profile provide a set of roles, activities-tasks, and work products (i.e., artifacts) useful for implementing a disciplined software development process targeted for VSEs.

2.2 An overview of the Scrum process

Scrum is one of the first and most used agile software development processes that emerged at the beginning of the agile software engineering approach during the 1995–2001 period (Fowler & Highsmith, 2001; Hoda et al., 2018; VersionOne, 2018). Agile software development methodologies and practices contrast radically to rigor-oriented ones (Abrahamsson et al., 2010; Hong et al., 2011). According to Abrahamsson et al. (2010), agile methodologies share the attributes of fast, small, and frequent delivery of usable software (i.e., an incremental development); a co-responsible and cooperative development process including the three usual actors as a customer, project facilitator, and development team (i.e., a whole team); and a highly customized, flexible set of agile practices seeking the highest optimization of required resources avoiding non-added value tasks and items (i.e., a minimalist adaptive process open to unexpected changes). According to Hong et al. (2011), the rigor-oriented projects contrast to agile ones mainly in the following issues: stable vs. volatile user requirements, a single large development cycle vs. many short development cycles, mandatory and rigid methodology steps vs. optional/mandatory but flexible methodology steps, and useful functionality delivered until the end of the development cycle vs. useful functionality delivered every 2–4 weeks.

Figure 2 portrays a diagram of the Scrum process (reproduced with authorization from www.scrum.org).

Figure 2 shows five Project Management activities (called events) and one Software Implementation activity, which are listed in turn below:

- Product Backlog Articulation—a Project Management activity
- Sprint Planning—a Project Management activity
- Daily Scrum—a Project Management activity
- Sprint Review—a Project Management activity
- Sprint Retrospective—a Project Management activity
- Sprint—a Software Implementation activity

Scrum is known as a Project Management process, and its product development activity is realized through the Sprint activity. A detailed description of these activities is out of the scope of this article due to space limitations, and thus, we report here an essential

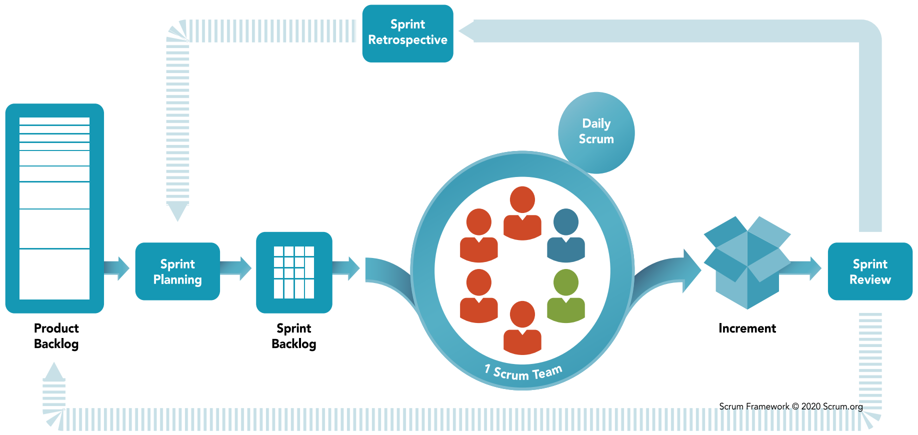


Fig. 2 Diagram of the Scrum process

description. Similarly, to the ISO/IEC 29110 standard analysis, a summarized but substantial description of these five Scrum Project Management activities is reported below. Full descriptions for each one of the five activities can be found in other sources (Schwaber & Sutherland, 2017; Sutherland, 2010).

Scrum is an iterative and incremental empirical process control that pursues transparency, inspection, and adaptation, and it is executed by a small team of people (Schwaber & Sutherland, 2017). A transparent control process makes any project item visible and accessible to all Scrum teams. Inspection is frequently exercised on the Scrum items, and if there are deviations from the targets, adaptations are performed as soon as possible. A Scrum Team fosters and lives the values of commitment, courage, focus, openness, and respect.

The Scrum Team is composed of three roles:

- Product Owner
- Scrum Master
- Developers

The Product Owner is a single person (i.e., not a group of people) who has the authority of defining and prioritizing the product requirements through the product backlog and defining the characteristics of the deliverables that need to be done. The Scrum Master is not considered a project manager in the traditional concept but a coach-evangelist of the Scrum process. The Scrum Master not only serves the Product Owner and Developers but also coaches them toward the Scrum project's goals accomplishment. Developers are a self-organized and cross-functional group involving from 3 to 9 people uniquely responsible for creating the increment artifact. All members of the Developers team are considered of similar rank and importance independently of the technical skill they provide to the Scrum project.

Essentially, a Scrum process requires one input and generates four internal products and three outputs. The input artifact is the Product Vision. The four internal generated products are Sprint Burndown Chart, Sprint Progress Review Record, Sprint Improvements Record, and Project Delivery Repository. The three outputs are Product Backlog, Sprint Backlog, and Increment. The last integrated increment is called the Done product.

The first Product Backlog Articulation activity defines the Product Backlog as a document that "lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases" (Schwaber & Sutherland, 2017; p. 15). This first activity defines the Product Backlog item attributes (i.e., feature description, feature test description, feature priority and value, feature effort estimation, and value). A Product Backlog emerges from the initial Product Vision that defines the user needs. A Project Delivery Repository is also implicitly built. The second Sprint Planning activity defines the Sprint Backlog which "is the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal" (Schwaber & Sutherland, 2017; p. 16). The third Daily Scrum activity is a very short-time meeting (usually a 15-min period) where the Developers use it "to inspect progress toward the Sprint Goal and to inspect how progress is trending toward completing the work in the Sprint Backlog" (Schwaber & Sutherland, 2017; p. 12). The fourth Sprint Review activity is organized at the end of each Sprint "to inspect the Increment and adapt the Product Backlog if needed" (Schwaber & Sutherland, 2017; p. 13). The fifth and last activity is the Sprint Retrospective activity, which is executed by all the Scrum Team (i.e., the three Scrum roles are included) to "inspect itself and create a plan for improvements to be enacted during the next Sprint" (Schwaber & Sutherland, 2017; p. 14). The product development activity, the Sprint, is "a time-box of one month or less during which a "Done," useable, and potentially releasable product Increment is created" (Schwaber & Sutherland, 2017; p. 8). Finally, an implicit Project Closure activity—presented in Fig. 2—"prepares the developed product for general release. Integration, system test, user documentation, training material preparation, and marketing material preparation are among closure tasks" (Schwaber, 1997; p. 14).

2.3 Overview of Electronic Process Guides

Electronic Process Guides (EPGs) are digital documents designed and built to improve the understanding, training, and execution of business processes (Kellner et al., 1998). EPGs are Knowledge Management documents rather than Business Process Management enablers for enacting an organizational process (Kellner et al., 1998; Moe & Dybå, 2006).

EPGs aim to overcome the natural limitations of printed documents or their digital versions (without a specific EPG design) such as deficient in form and content; difficulties to understand its content, to use it, and to access it; and scarcely used in the practice (Hauck et al., 2008; Kellner et al., 1998; Leuser et al., 2009). In particular, Dingsøyr et al. (2004) associated the printed process guides with negative effects such as few consulted, difficult to read, and just "dust collectors." In turn, benefits of the utilization of EPGs have been identified such as training and process guidance improvement (Kellner et al.,, 1998); gradual tailoring of process, reuse, process conformance, resulting in better process management (Becker-Kornstaedt, 2000); and process communication improvement (Koolmanojwong et al., 2008).

The usual content of an EPG includes (Becker-Kornstaedt, 2000; Kellner et al., 1998) the following sections: overview, phases, activities, tasks, roles, work products, and additional resources (i.e., list of terms, tasks guidelines, templates, examples, whitepapers, and tools). An overview is a concise textual description of the process guidance. Phases refer to texts and diagrams, which describe the integrated view of the process from an overall organization. Activities refer to the workflow of tasks, inputs, outputs, control metrics, and tools and roles participating in them. Tasks describe the specific steps realized in each workflow. Roles describe the types of human agents in the process. Work products describe the required and generated artifacts that are created through the process. Finally, additional resources describe complementary material. In summary, EPGs provide the following functional properties: to present an organized scheme of the knowledge chunks, to use a similar user interface in all digital pages; to provide user navigation flexibility, to present information in an adequate language, and to provide complementary information access tools (search engines, dictionaries, language translators, and so on).

However, EPG utilization to obtain the previous benefits is not a seamless implementation process (Moe & Dybå, 2006). To overcome implementation difficulties, several design recommendations have been reported. EPGs may be developed from a simple design and gradually evolve to avoid an "overdone" and badly designed EPG (Dingsøyr et al., 2004). General EPG design recommendations are (Kellner et al., 1998)

- minimal content structure
- inclusion of all core process information
- flexible page navigation for users
- ease of use
- keep a standardized format of pages
- minimal effect of windows juggling

A review on the utilization of EPGs in the domain of Software Engineering can be consulted in Mora et al. (2016).

2.4 Related work

The reconciliation problem, between the agile development methods (mainly Scrum and XP) and the software process standards and models, has been previously studied (Sutherland et al., 2008; Jakobsen & Johnson, 2008; Pikkarainen, 2009; Bass et al., 2013; Garzás & Paulk, 2013; Pasini et al., 2013; Silva et al., 2015; Torrecilla-Salinas et al., 2016; Suteeca & Ramingwong, 2016; Henriques & Tanner, 2017; Galvan-Cruz et al., 2017a; Muñoz et al., 2018; Palomino et al., 2016; Jirapanthong, 2019).

There are studies on the agility-rigor reconciliation problem mainly between Scrum and the CMMI-DEV version 1.3 maturity model (Sutherland et al., 2008; Jakobsen & Johnson, 2008; Pikkarainen, 2009; Bass et al., 2013; Garzás & Paulk, 2013; Silva et al., 2015; Torrecilla-Salinas et al., 2016; Henriques & Tanner, 2017; Palomino et al., 2016). However, given the novelty of the ISO/IEC 29110 standard series, there are scarce full studies on the reconciliation of this standard and Scrum (or other agile processes) (Galvan-Cruz et al., 2017a; Jirapanthong, 2019; Muñoz et al., 2018; Pasini et al., 2013; Suteeca & Ramingwong, 2016). Table 1 summarizes these findings.

The 14 studies reported in Table 1 were populated through a selective review process (Glass et al., 2004). A selective literature review qualifies as a descriptive research approach and literature analysis research method (Glass et al., 2004). A selective literature review differs from a systematic literature review (Brereton et al., 2007) in the magnitude of the sample of studies selected once fixed the selection criteria because it does not analyze all studies from an exhaustive search but from a reduced and selective sample of studies. A selective literature review differs also from a mapping study (Petersen et al., 2008) in the purpose of extracting core findings related to specific research questions rather than

Table 1 Summary of research findings on the agility-rigor reconciliation problem

Research	Main finding (positive (+) or negative (−) final impact)	New Scrum?/An EPG?
Sutherland et al. (2008)	CMMI-DEV® v.1.3 generic practices helped to institutionalize Scrum (+)	No/No
Jakobsen and Johnson (2008)	CMMI-DEV® v.1.3 practices enhanced Scrum's weaknesses in several activities (+)	No/No
Pikkarainen (2009)	Lack of well-defined agile reference models leads to the high variability of Scrum or XP utilization (−)	No/No
Bass et al. (2013)	Project Planning and Control and Risk Management must be enhanced in Scrum (−)	No/No
Garzás and Paulk (2013)	Scrum helped to reach a CMMI-DEV® v.1.3 level 2 (+)	No/No
Silva et al. (2015)	Scrum helped to reach a CMMI-DEV® v.1.3 level 2 (+). Alone Scrum practices are insufficient (−)	No/No
Torrecilla-Salinas et al. (2016)	Project Planning and Control and Risk Management must be enhanced in Scrum (−)	No/No
Henriques and Tanner (2017)	Scrum helps to already CMMI-DEV® v.1.3 level 5 organizations (+). Scrum is hard to be used in CMMI level 2 or 3 organizations (−)	No/No
Palomino et al. (2016)	Scrum helped to reach a CMMI-DEV® v.1.3 level 2 (+)	No/No
Pasini et al. (2013)	A new Q-Scrum process is proposed from the fusion of Scrum and the ISO/IEC 29110 Basic profile (+)	Yes/No
Galvan-Cruz et al. (2017a)	Scrum has an overall compliance moderate level (79%) regarding the ISO/IEC 29110 Entry profile (−)	No/No
Suteeca and Ramingwong (2016)	Scrum practices cover all of the ISO/IEC 29110 Basic profile Software Implementation activities (+)	No/No
Muñoz et al. (2018)	The ISO/IEC 29110 Basic profile implementation in four VSEs helped to better utilization of Scrum (+)	No/No
Jirapanthong (2019)	The ISO/IEC 29110 Basic profile implementation in eight VSEs helped to better utilization of Scrum (+)	No/No

elaborating a visual multidimensional classification of topics regarding the dimensions of interest for the researchers.

We applied the following steps: (1) to define general and specific knowledge inquiries, (2) to define the criteria for selecting sources of studies, (3) to define the generic document search statement as well as the document inclusion and exclusion criteria, (4) to execute knowledge search procedures, (5) to analyze the abstract of the located documents and decide its inclusion or exclusion for detailed analysis, (6) to elaborate a detailed analysis of included documents, and (7) to populate the table data.

In step 1, we defined the general knowledge inquiry as "to identify the recent studies (2008–2020 period) addressing the reconciliation problem between the agile development methods (mainly Scrum and XP) and the software process standards and models," and the specific knowledge inquiries as "to find the positive and negative impacts," "to identify whether a new Scrum or XP process was proposed," and "to identify whether an Electronic Process Guide was elaborated." In step 2, we defined the criterion for selecting sources of studies to the "journals, conferences, research-oriented books, and technical reports focused on Software Engineering area reported in the scholar Google search engine." In step 3, we defined the generic document search statement as "Title OR Abstract contains ((reconciliation OR harmonization OR integration OR combining OR enhancement OR study OR investigation) AND (agile OR scrum OR Extreme Programming OR XP OR maturity model OR ISO/IEC standard OR ISO/IEC 29110)),” and the inclusion and exclusion criteria respectively as “to include the study when it is focused on the general and specific knowledge inquiries from the step 1” and “to exclude the study when it addresses marginally the general and specific knowledge inquiries from the step 1.” In steps 4 and 5, 14 studies were approved for their analysis. Finally, in steps 6 and 7, we analyzed the full study and elaborated Table 1, respectively.

Table 1 reports the study in its first column. The main positive (represented with the plus symbol) or negative (represented with the negative symbol) impacts to combine or reconciliate a rigor-oriented model, process, or standard with an agile process are reported in the second column. Finally, the responses to the inquiry whether an enhanced or modified agile process and an Electronic Process Guide (EPG) were generated are reported in the third column. Hence, it was identified that only one study (Pasini et al., 2013) reported an enhanced or modified agile process, and none reported the elaboration of an EPG.

The studies on the reconciliation of Scrum and CMMI-DEV®version 1.3 revealed contrasting evidence on their combined use. Some studies reported benefits of using CMMI® practices to enhance Scrum ones and to accelerate a CMMI-DEV® level 2 achievement through Scrum practices (Sutherland et al., 2008; Jakobsen & Johnson, 2008; Garzás & Paulk, 2013; Palomino et al., 2016). Other studies identified that Scrum is insufficient to cover CMMI-DEV® practices and must be enhanced (Pikkarainen, 2009; Bass et al., 2013; Torrecilla-Salinas et al., 2016), and another one found mixed evidence (Henriques & Tanner, 2017). The positive integration of Scrum and CMMI-DEV® has been using the CMMI-DEV® Generic Practices to foster the Scrum institutionalization (Jakobsen & Johnson, 2008; Sutherland et al., 2008); using CMMI-DEV® for enhancing Scrum weaknesses on Risk Management, Project Estimation, Project Tracking, and Configuration Management activities (Jakobsen & Johnson, 2008); and helping to reach a CMMI-DEV® level 2 by using Scrum (Garzás & Paulk, 2013; Palomino et al., 2016). The negative integration of Scrum and CMMI-DEV® refers to lack of well-defined agile reference models because Scrum and XP foster high flexibility and freedom for using their agile practices (Pikkarainen, 2009), the need for relaxing some CMMI practices to fit the agile culture (Pikkarainen, 2009), and the need for enhancing some Scrum practices such as Project Planning and Control, Risk Management, and Organization Level of Standard Processes (Bass et al., 2013; Torrecilla-Salinas et al., 2016). Interesting findings on the contrasted evidence refer to the positive benefits of using Scrum to reach CMMI-DEV®level 2 or 3. Benefits were observed for using Scrum in already CMMI-DEV® level 5 organizations (Henriques & Tanner, 2017). However, it is also indicated that using Scrum and other agile practices alone is insufficient for this aim (Silva et al., 2015), given the difficulties of its utilization in the CMMI-DEV® level 2 or 3.

Regarding the reconciliation of Scrum and the ISO/IEC 29110, a few studies report the benefits of enhancing the Scrum practices with the ISO/IEC 29110 to fit the market competitive regulations, and that Scrum practices are insufficient for covering all ISO/IEC 29110 requirements of the Entry profile. Consequently, adjustments to the Scrum process have been proposed (Galvan-Cruz et al., 2017a; Jirapanthong, 2019; Muñoz et al., 2018; Pasini et al., 2013; Suteeca & Ramingwong, 2016). Pasini et al. (2013) proposed a new Q-Scrum process that adds roles, well-defined work products, and a fusion of the Scrum activities with the Project Management and Software Implementation ones with those of the ISO/IEC 29110 Basic profile. Suteeca and Ramingwong (2016) mapped the ISO/IEC 29110 Software Implementation process to the Scrum practices, and they estimated that they cover all of them. However, the mapping was only at a high-level and there is a lack of specific how-to details for this covering. Galvan-Cruz et al. (2017a) conducted a compliance analysis of the Project Management agile practices in Scrum (and XP and UPEDU methodologies), against the Project Management process of the Entry profile of ISO/IEC 29110. The authors identified an overall low (51%), moderate (79%), and very high (93%) compliance level for XP, Scrum, and UPEDU respectively. Scrum was assessed at a moderate level for roles (78%), work products (70%), and a very high level (90%) regarding activities-tasks. The authors concluded, thus, that UPEDU supports practically all the ISO/IEC 29110 statements of the Entry profile but Scrum and XP must be enhanced with complementary project management practices if better compliance with the Entry profile of ISO/IEC 29110 is required by market competitive regulations. Muñoz et al. (2018) reported a multi-case study with four VSEs that had experienced difficulties (i.e., weak project planning; very informal project tracking; lack of meeting records; null version control; and other issues related to the Software Implementation process) for correctly applying the Scrum practices. However, with the implementation of the Basic profile of the ISO/IEC 29110 and their certification, the four VSEs overcame most of such difficulties. Finally, Jirapanthong (2019) conducted an action research study with eight VSEs. In this study, the author trained software development teams from the eight already Scrum-trained VSEs to use the Basic profile of ISO/IEC 29110. The eight teams reported, via a questionnaire, benefits in several activities.

The overall findings from these 14 studies can be stated as follows: (1) Scrum practices and CMMI-DEV® or ISO/IEC 29110 integrations have been promoted for VSEs, (2) Scrum practices are useful, but alone they are insufficient for fitting the CMMI-DEV® or ISO/IEC 29110 statements, and (3) the results of the integration and evaluation of Scrum and CMMI-DEV® or ISO/IEC 29110 have not been reported in an Electronic Process Guide.

3 Description of the experiment

3.1 Experiment goal

To set the experiment goal, we used the structured experiment goal template reported by Wohlin et al. (2012). This template is as follows: *Analyze < Object(s) of study > for the purpose of < Purpose > with respect to their < Quality focus > from the point of view of the < Perspective > in the context of < Context >*. Object(s) of study refers to products, processes, resources, models, metrics, or theories to be investigated. Purpose is the intention of experimenting (i.e., evaluate, identify, or improve). Quality focus accounts for the main effect(s) of interest in the experiment. Perspective refers to the viewpoint of the groups of interest in the experimental results (i.e., from an academic vs. professional perspective; from software operational positions vs. software managerial positions). Context refers to the experiment setting characteristics (students vs. professional participants, novice vs. expert participants, academic vs. professional participants, concurrent vs. off-line experiment mode, and toy vs. real problems).

The experimental objects of study in this research corresponded to the Scrum processes (the original Scrum and the Scrum +) packaged for their experimental analysis in the Electronic Process Guidelines (EPGs) (i.e., the Scrum EGP and the Scrum + EPG). The purpose of this study was to evaluate the objects of study regarding the quality focus on six usability metrics from a dual academic-professional perspective. An academic perspective is of interest to the implications for teaching and researching conciliated agile-rigorous software development practices, as well as for the professional communities interested in taking advantage of conciliated agile-rigorous software development practices toward their *Iron Triangle* aim. The context of this study is of international academic-professional participants in an off-line experimental mode.

Thus, we can report the structured goal template as follows: *Analyze < two Scrum processes packaged in two EPGs > for the purpose of < evaluating them > with respect to their < six usability metrics > from the point of view of the < dual academic-professional perspective > in the context of < international academic-professional participants in an off-line experimental mode >.*

3.2 Experiment hypotheses

Based on the structured goal template, we formulated six statistical null hypotheses to evaluate whether the reconciled Scrum + process contained in the Scrum + EPG was identified with better score metrics than the original Scrum process contained in the Scrum EPG, from the experiment participants. These six null hypotheses were the following:

- H0.1 The usefulness metric for the Scrum + EPG is less or equal than the usefulness metric for the original Scrum EPG.
- H0.2 The ease of use metric for the Scrum + EPG is less or equal than the ease of use metric for the original Scrum EPG.
- H0.3 The compatibility metric for the Scrum + EPG is less or equal than the compatibility metric for the original Scrum EPG.
- H0.4 The value metric for the Scrum + EPG is less or equal than the value metric for the original Scrum EPG.
- H0.5 The normative beliefs metric for the Scrum + EPG is less or equal than the normative beliefs metric for the original Scrum EPG.
- H0.6 The attitude metric for the Scrum + EPG is less or equal than the attitude metric for the original Scrum EPG.

3.3 Experiment variables and instruments

The variables used in this study correspond to the six metrics of usability (Cayola & Macías, 2018; Karahanna et al., 1999; Mora et al., 2016; Riemenschneider et al., 2002) of Usefulness, Ease of Use, Compatibility, Value, Normative Beliefs, and Attitude. The Usefulness metric refers to the degree to which using the new tool is perceived as being better than using the currently used tool and it was measured with four statements. The Ease of Use metric refers to the degree to which using the new tool is perceived as being free of effort and it was measured with three statements. The Compatibility metric refers to the degree to which using a new tool is perceived as compatible with what people do and it was measured with three statements. The Value metric refers to the degree of value delivered to users by obtaining savings on money, time, and additional resources, as well as an overall value, and it was measured by four statements. The Normative Beliefs metric refers to the degree of external approbation of performing a given behavior as well as by the motivation for complying with it, and it was measured with six statements. The Attitude metric refers to the degree of the favorable or unfavorable evaluation of the expected behavior of using the evaluated artifact (i.e., in this study the two EPGs), and it was measured with three statements.

All the six metrics of usability were measured using a 5-point Likert scale except the attitude metric which was measured with a 7-point semantic differential scale from − 3 to + 3. The 5-point Likert scale varied from strong disagreement, disagreement, neutral, and agreement to strong agreement for the variables of Usefulness, Ease of Use, Compatibility, and Normative Beliefs. For the metric of Value, the 5-point Likert scale varied from very low, low, moderate, high, to very high. For the metric of Attitude, the 7-point semantic differential scale from − 3 to + 3 varied from extremely positive to extremely positive (statement 1), from extremely bad to extremely good (statement 2), and from extremely harmful to extremely beneficial (statement 3).

The instruments of this experiment correspond to the questionnaires elaborated to collect the six metrics of usability from the 32 international evaluators. These questionnaires have been previously used in other studies (Cayola & Macías, 2018; Karahanna et al. 1999; Mora et al. 2016; Riemenschneider et al. 2002). The six instruments for the six metrics of usability are reported in Appendix 1.

3.4 Experiment subjects

The subjects were contacted from an international list of 80 professional people from LinkedIn groups related to agile methods and the Entry profile of the ISO/IEC 29110, as well as from academic contacts of the research team. A group of 32 international evaluators accepted voluntary participation. Core demographic characteristics of the 32 international evaluators are as follows: (1) the evaluators were collected from 3 relevant international regions such as North America (9%), Europa/Asia (25%), and Latin America (66%); (2) there is a relevant number of experts (53%); (3) the sample has both academics (56%) and practitioners (44%) type of evaluators; and (4) a high percentage of the participants (81%) have sufficient working experience (at least 5 years).

Table 2 presents the main demographic attribute (i.e., the period of years using/knowing agile methods or the ISO/IEC 29110 standard). Novices were considered the evaluators in the range of 0–4 years of using/knowing agile methods and/or the ISO/IEC 29110, and experts were evaluators with 5 or more years of experience with agile methods and/or the ISO/IEC 29110. Whereas the using and knowing capabilities of the subjects can be assessed differently, in this research, we used this construct as a conceptual unit. We did not apply statistical tests with blocks grouping academics vs. professionals, and novices vs. experts, but upon the total sample size of 32 international evaluators. Thus, this conceptual consideration does not affect the validity of the statistical conclusions. Table 2 reports the size of the blocks for informative purposes on the four types of blocks belonging to the total sample size of 32 subjects.

Table 2 Period of years using/knowing agile methods or the Entry profile of ISO/IEC 29110

Period of years using/knowing Scrum or the ISO/IEC 29110	Academics	Professionals	Sub-totals
Novices: 0–4 years	6	9	15
Experts: (5 or more years)	12	5	17
Sub-totals	18	14	32

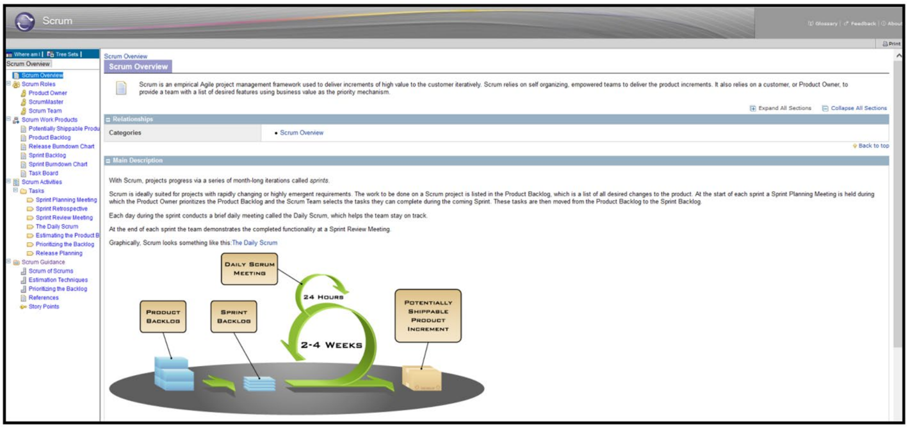
3.5 Experiment materials

The materials of the experiment correspond to two Electronic Process Guides (EPGs). An experiment material was the public available Scrum EPG designed by the EPF Eclipse Foundation (2019), and the other one was the new Scrum+EPG. Scrum+EPG was designed with the application of the Means-Ends Analysis (Anderson, 1993; Simon & Newell, 1971). The Means-Ends Analysis is a heuristic design method for systematically applying rule-fixed planned changes to an original Scrum process, with the design goal of producing a reconciled Scrum+process. A summary of the applied design process with Means-Ends Analysis can be consulted in Galvan-Cruz et al. (2017b) and Appendix 2.

The Scrum+EPG was built with the open-source tool EPF (Eclipse Process Framework) (Haumer, 2007). EPF is a powerful tool that provides support for structured documentation and automatically generates HTML-based EPGs. EPF has been used for creating EPGs in multiple types of research (Garcia et al., 2011; Abad et al. 2012; Ghanadbashi & Ramsin, 2016; Mora et al., 2016), and it has been highly suggested by the editors of the ISO/IEC 29110 series (O'Connor & Laporte, 2017) for developing Deploying Packages (i.e., digital guides for using the ISO/IEC 29110) (Laporte, 2019). Figures 3 and 4 display the main page of the original Scrum EPG (available online at: http://vm21-labdc.uaa.mx/scrum) and the Scrum+EPG (available online at http://x3620a-labdc.uaa.mx/scrum_plus). The original Scrum EPG can be also downloaded (Eclipse Foundation, 2019).

3.6 Experiment type

The type of experiment design selected was a within-subjects experimental design (Berger & Maurer, 2018). This experimental method applies all treatments to all subjects (i.e., the same subject works as his/her control group). However, the application of each treatment to each subject is randomly selected to avoid the negative order carryover effect (Berger & Maurer, 2018). This experimental method has been used commonly in software engineering research (Basili et al. 1996; Briand et al., 2001; Johanson & Hasselbring, 2017; Vegas et al., 2015). The two treatments were the original Scrum EPG and the new elaborated Scrum+EPG. Thus, a total of 32 subjects received both treatments but each one in random order.



Scrum is an empirical Agile project management framework used to deliver increments of high value to the customer iteratively. Scrum relies on self organizing, empowered teams to deliver the product increments. It also relies on a customer, or Product Owner, to provide a team with a list of desired features using business value as the priority mechanism.

Categories: • Scrum Overview

With Scrum, projects progress via a series of month-long iterations called *sprints*.

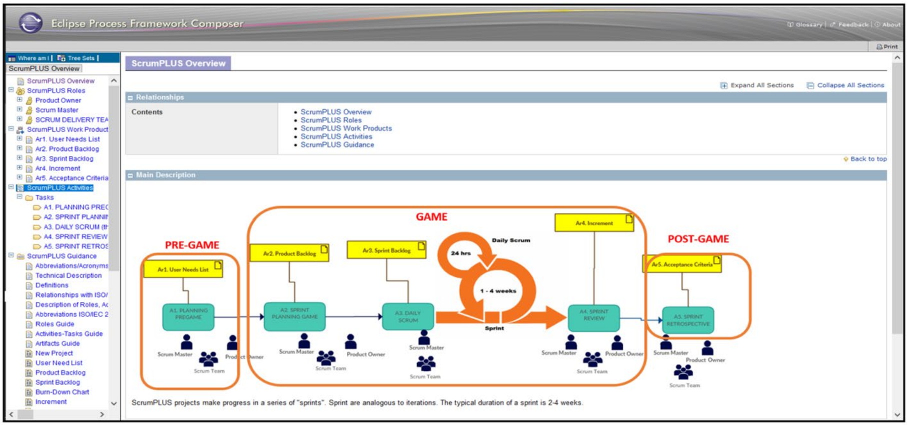
Scrum is ideally suited for projects with rapidly changing or highly emergent requirements. The work to be done on a Scrum project is listed in the Product Backlog, which is a list of all desired changes to the product. At the start of each sprint a Sprint Planning Meeting is held during which the Product Owner prioritizes the Product Backlog and the Scrum Team selects the tasks they can complete during the coming Sprint. These tasks are then moved from the Product Backlog to the Sprint Backlog.

Each day during the sprint conducts a brief daily meeting called the Daily Scrum, which helps the team stay on track.

At the end of each sprint the team demonstrates the completed functionality at a Sprint Review Meeting.

Graphically, Scrum looks something like this:The Daily Scrum

Fig. 3 Scrum EPG main display



ScrumPLUS projects make progress in a series of "sprints". Sprint are analogous to iterations. The typical duration of a sprint is 2-4 weeks.

Fig. 4 Scrum + EPG main display

4 Experiment execution, results, and discussion

In this section, we present details on how the experiment was performed. We also present the results, a discussion, and implications of the results.

4.1 Experiment instructions and tasks

The experiment was conducted off-line for 1 month—July 2017. All 32 international evaluators received an email with a document of instructions to participate in the experiment. Appendix 3 reports the content of this document. As a summary, the 32 international evaluators were asked to execute the following tasks: (1) randomly select one of the two EPGs and navigate and review it during a period of 20–30 min, (2) navigate and review another EPG for a similar period of 20–30 min, and (3) answer the demographic data and usability metric questionnaires.

4.2 Experiment results

Table 3 reports the descriptive statistics—median, mean, and standard deviation—for the six usability metrics for the Scrum + EPG and Scrum EPG. These descriptive results show—before any statistical test—that the 32 international evaluators assigned higher scores, summarized in the median and mean values for the six usability metrics, for the Scrum + EPG than for the Scrum EPG. As it was reported previously in Sect. 3.3, the first five metrics were assessed with a 5-point Likert scale and the sixth metric with a 7-point semantic differential scale from − 3 to + 3.

Table 4 reports 12 one-sample Wilcoxon signed-rank tests for the six usability metrics for the Scrum + EPG and Scrum EPG to provide statistical evidence that the 10 medians can be considered greater than 3.00 for the Usefulness, Ease of Use, Compatibility, Value, and Normative Beliefs metrics or greater than 0.00 for the Attitude metric. These 12 tests are required before applying the six Wilcoxon matched-pairs signed-rank tests to establish the relevance of the comparison of medians between the two EPGs. Without these initial

Table 3 Descriptive statistics for the six usability metrics in the two EPGs

Metric	Scrum EPG			Scrum + EPG		
	Median	Mean	Std.Dev	Median	Mean	Std.Dev
Usefulness	3.75	3.63	0.47	4.00	4.16	0.61
Ease of use	3.67	3.68	0.73	4.00	4.16	0.65
Compatibility	3.50	3.63	0.72	4.00	4.18	0.71
Value	4.00	3.67	0.70	4.38	4.29	0.54
Normative beliefs	3.67	3.64	0.62	4.00	4.07	0.71
Attitude	1.00	1.11	0.96	2.00	2.08	0.88

12 tests, the comparison of the medians would not be relevant if both medians were scored with low values. For instance, if the medians for the Usefulness metric for Scrum + EPG and Scrum EPG were respectively 2.00 and 1.00, then the statistical test could provide support to consider the Usefulness of the Scrum + EGP better than that of the Scrum EGP, but both would be practically low scored, and thus, this result would be not relevant.

Finally, a Wilcoxon matched-pairs signed-rank test was used for the six hypotheses. This test is non-parametric, and it avoids assumptions and additional pre-tests on normality distributions in data, and on the need for the equality of variances in the two datasets to be compared (Sheskin, 2000). Non-parametric statistical tests also perform well with small samples (i.e., among 15–20 subjects) (Sheskin, 2000; Gresse von Wangenheim et al., 2009; Vegas et al., 2015). Table 5 reports the results of the statistical tests.

This study collected data from a sample size of 32 subjects, and thus, we consider it does not constitute a threat to the statistical validity of results. This test compares medians

Table 4 One-sample Wilcoxon signed rank tests for the six usability metrics in the two EPGs

Metric	Scrum EPG		Reject H0 with an α-value = 0.01	Scrum + EPG		Reject H0 with an α-value = 0.01
	Median	P-value		Median	P-value	
H0.1 The usefulness metric for the EPG is less or equal than 3.00	3.75	0.0001	Yes	4.00	0.0001	Yes
H0.2 The ease of use metric for the EPG is less or equal than 3.00	3.67	0.0001	Yes	4.00	0.0001	Yes
H0.3 The compatibility metric for the EPG is less or equal than 3.00	3.50	0.0001	Yes	4.00	0.0001	Yes
H0.4 The value metric for the EPG is less or equal than 3.00	4.00	0.0001	Yes	4.38	0.0001	Yes
H0.5 The normative beliefs metric for the EPG is less or equal than 3.00	3.67	0.0001	Yes	4.00	0.0001	Yes
H0.6 The attitude metric for the EPG is less or equal than 0.00	1.00	0.0001	Yes	2.00	0.0001	Yes

Table 5 Results for the six null hypotheses

Null hypotheses	Medians		*T*-value	*P*-value	Reject H0 with an α-value = 0.01
	Scrum + EPG	Scrum EPG			
H0.1 The usefulness metric for the Scrum + EPG is less or equal than the usefulness metric for the original Scrum EPG	4.00	3.75	43.00	0.0001	Yes
H0.2 The ease of use metric for the Scrum + EPG is less or equal than the ease of use metric for the original Scrum EPG	4.00	3.67	23.00	0.0006	Yes
H0.3 The compatibility metric for the Scrum + EPG is less or equal than the compatibility metric for the original Scrum EPG	4.00	3.50	16.00	0.0001	Yes
H0.4 The value metric for the Scrum + EPG is less or equal than the value metric for the original Scrum EPG	4.38	4.00	10.50	0.0001	Yes
H0.5 The normative beliefs metric for the Scrum + EPG is less or equal than the normative beliefs metric for the original Scrum EPG	4.00	3.67	10.00	0.0001	Yes
H0.6 The attitude metric for the Scrum + EPG is less or equal than the attitude metric for the original Scrum EPG	2.00	1.00	11.00	0.0001	Yes

instead of means, and it has been frequently used in Software Engineering research (Basili et al., 1996; Briand et al., 2001; Johanson & Hasselbring, 2017). The free statistical tool used was MaxStatLite (MaxStatLite, 2019).

The null hypothesis for each one of the six metrics of usability is reported in the first column. The medians for the Scrum + EPG and the Scrum EPG related to each usability metric are reported in the second and third columns. The value of the statistic *T* obtained from the (directional one-tailed) Wilcoxon matched-pairs signed-rank test is reported in the fourth column. The *P*-value (i.e., the maximum probability of having the type-I error by rejecting the null hypothesis when it is true) is reported in the fifth column. Finally, the decision on rejecting (or failure on rejecting) the null hypothesis is reported in the sixth column. An α-value significance test level of 0.01 was fixed. The six null hypotheses were rejected (i.e., the *P*-values obtained were less than the fixed α-value), and thus, the scores received for the Scrum + EPG usability metrics can be considered better ones than the received for the Scrum EPG for this group of 32 international evaluators.

5 Discussion

We do not pursue the research goal of comparing the usability metrics from academics vs. professionals, and novices vs. experts. Whereas this research goal is worthy and has been studied in other domains (Mora et al., 2016), we consider that this study can be pursued in the next research stage. However, we report that the group of 32 international was composed of 18 academics and 14 professionals. This group was also classified as experts versus novices in agile practices and/or the Project Management process of ISO/IEC 29110 standard-Entry profile. The group of 18 academics had 11 experts and 7 novices. The group of 14 professionals had 4 experts and 10 novices. Thus, the inclusion of these four types of groups of evaluators helped to have a representative and varied sample of evaluators.

In this study, we focused on investigating the effectiveness (measured on the six usability metrics) of a recent designed Scrum + EPG aligned to the Project Management process of the ISO/IEC 29110 standard-Entry profile-. Agile software development practices are being frequently used by small and very small software development organizations, but they are also asked to fit regulatory standards to increase their competitive market status (Laporte et al., 2013a; O'Connor & Laporte, 2017). The ISO/IEC 29110 standard series was released to target very small organizations, from 1 to 25 people, but its inherent design despite it is lightweight it cannot be considered agile. Consequently, very small software development organizations using an agile process like Scrum or XP that are interested in complying with regulatory standards face what is known as the "agility-rigor reconciliation problem" between two different software process approaches (Boehm & Turner, 2003, 2004; 2005; Magdaleno et al. 2012; Silva et al., 2015). At present, no official reconciliated solution between the agile process and the ISO/IEC 29110 standard has been released, but it is currently being developed (Galvan-Cruz et al., 2021).

Table 5 provides empirical statistical-tested evidence that the group of 32 international evaluators—composed of academics and professionals, as well as experts and novices—perceived the reconciliated Scrum + EPG to the Project Management process of the ISO/IEC 29110 standard-Entry profile with a better Usefulness, Ease of Use, Compatibility, and Value, as well as more recommended to be used from the Normative Beliefs and the final Attitude toward the utilization of the Scrum + EPG. Although the sample size is small, the size of 32 subjects is adequate for non-parametric tests, as such tests perform well with small samples in the range of 15–20 subjects (Sheskin, 2000; Gresse von Wangenheim et al., 2009; Vegas et al., 2015).

The Usefulness metric was measured with three statements related to performing agile Project Management tasks faster, with better quality and better effectiveness. The Ease of Use metric was also measured with three statements related to easiness for learning and operating the EPG, as well as being it not considered difficult regarding agile Project Management tasks. Compatibility metric was measured with three statements related to work compatibility aspects, alignment to work style, and linkage to like the way of work on agile Project Management tasks. The Value metric was measured with four statements related to saving money, time, and providing data, information, and knowledge, and overall value on agile Project Management tasks. The Normative Beliefs was measured with four statements related to the external recommendations for using the evaluated EPG from peers, supervisors, IT area, and IT staff for agile Project Management tasks. Finally, the Attitude metric was measured with three statements related to the positive–negative, good-bad, and harmful-beneficial semantic ranges for using the evaluated EPG for agile Project Management tasks in the next 6 months.

These results can be interpreted as initially satisfactory on the research stream efforts toward the elaboration of a commonly accepted agile solution that is aligned to the recommendations of a Software Engineering ISO/IEC standard. Furthermore, from the specific design recommendations reported in Galvan-Cruz et al. (2017b), new designers of similar enhanced Scrum or XP processes are alerted on the no obvious and inherently design difficulty to apply a controlled-balanced approach of changes to the agile process to achieve an acceptable level of alignment without the loss of agility perception from the evaluators and potential users of this reconciliated solution. The non-controlled addition of issues from the ISO/IEC 29110 standard to the agile process can derive in a non-agile one, as well as in a slightly aligned solution to the standard.

5.1 Implications for the knowledge of reconciled agile-rigorous software development methods

The relevance of counting with reconciled agile-rigorous software processes can be argued on the permanent aim of any software process as the achievement of the expected *Iron Triangle* on quality, cost, and schedule (Agarwal & Rathod, 2006; Slaughter et al., 1998). Medium- and large-sized organizations usually count on economic, human, and organizational resources to implement software process standards and models (Laporte, et al., 2013a; Laporte et al., 2013b). However, the population of small and very small ones has been unserved populations. According to the ISO/IEC 29110 standard (ISO/IEC, 2012), small and very small business organizations, including software development providers, account for—at least—90% of the total of business in the OECD countries and these kinds of organizations are usually sub-contracted for specific software needs by medium- and large-sized organizations facing a software process maturity process gap between both types of organizations. It is also reported that all software process standards and models, except the rigor-oriented PSP model (Humphrey, 1999), have been designed for medium- and large-sized organizations, and thus, the small and very small software development business cannot fit the expected software process improvement efforts (O'Connor & Coleman, 2009; Staples et al., 2007). These small and very small organizations, alternatively, have accepted the agile practices, which have generated partial benefits. However, these agile practices alone are insufficient for that these organizations can reach high-maturity process levels (Pikkarainen, 2009; Bass et al., 2013; Torrecilla-Salinas et al. 2016). Additionally, the market competition needs of providing international software process certification (Laporte et al., 2013a) place more pressure on the small and very small organizations for improving their agile and informal software development process. With this real international context, the need for scientifically knowing how to design and provide reconciled agile-rigorous software development process to small and very small software development business is worth to be studied.

Hence, thus, we consider this research contributes to the Software Engineering domain as follows: (1) It opens a research avenue on potential solutions for the agility-rigor reconciliation problem through the utilization of EPGs, (2) it introduces the Means-Ends design process in the Software Engineering domain which provides a systematic and objective method for elaborating an expected software process (i.e., the goal state) from a software process base (i.e., the initial state) employing a set of transformation actions (Galvan-Cruz et al., 2017b; Appendix 2, and 3) it remarks the simple but relevant theoretical problem of finding a balanced rigor-agile software process to avoid the failed design of a solution that has lost its agility practices.

5.2 Implications for the practice of reconciled agile-rigorous software development methods

To the best of our knowledge, and given the literature review evidence conducted until 2019, no similar full Scrum EPG reconciled to the Project Management process of the

ISO/IEC 29110 standard-Entry profile – has been already reported, although initial partial efforts are recently available (Galvan-Cruz et al., 2021). In contrast, EPGs for supporting rigor-oriented and heavy software processes from standards and models have been reported (Abad et al., 2012; Garcia et al.,, 2011; Ghanadbashi & Ramsin, 2016; Mora et al., 2016), but none for very small organizations. The Scrum + EPG is a free-access tool available online (at http://x3620a-labdc.uaa.mx/scrum_plus), so any very small software development organization can be benefited by using it.

Hence, thus, we consider this research can also contribute to very small software development organizations using agile practices that are interested in combining them with the best practices derived from rigor-oriented software process from standards and models, with the following issues: (1) to have available an online and free-access EPG of a Scrum process reconciled with a relevant ISO/IEC software process standard designed specifically for very small organizations, (2) to provide digital access to the reconciled Scrum process through an EPG for very small organizations, and (3) to help to disseminate the best practices provided by the ISO/IEC 29110 standard in very small organizations to reduce training costs.

6 Threats to validity

According to Wohlin et al. (2012; p. 102), four types of threats to the validity of experimental research must be analyzed. These are the construct, internal, conclusion, and external validity. Table 6 summarizes the type of validity, threat statement, and countermeasure that was applied.

Construct validity (Jedlitschka et al., 2008) refers to the extent to which the instruments used for measuring the observed variables correspond really to the theoretical variables (called constructs). Internal validity (Wohlin et al., 2012) refers to the extent to which the studied causal relationship between the treatments (independent) and the outcome (dependent) variables are not spurious (i.e., modified by another variable not considered in the experiment). Conclusion validity (Jedlitschka et al., 2008; Wohlin et al., 2012) refers to the extent to which the results can be supported with the correct statistical tests applied to the experimental data. Finally, external validity ((Jedlitschka et al., 2008; Wohlin et al. 2012) refers to the extent to which the conclusions can be generalized to other similar populations. In this investigation, we applied countermeasures to reduce the threats to these four types of validity (Appendix 4).

7 Conclusion and future work

This research concludes that the agility-rigor reconciliation problem is an issue that is currently faced by very small software development organizations. These organizations have preferred the utilization of agile practices, but they are also pushed by competitive market issues to reach satisfactory software process maturity levels. Thus, these organizations are interested in combining the utilization of agile practices with adequate software process standards and models. For this kind of organization, the ISO/IEC 29110 series of standards and guides are available. However, to the best of our knowledge, no previous reconciliation

Table 6 Countermeasures applied to threats to validity

Type of validity	Threat	Countermeasure
Construct	• Content validity of instruments	• The six instruments were selected from the literature. No new item was added.
	• Convergent validity of instruments	• PLS (Chin, 2010) statistical method was applied. Three statements (one for *Usefulness* and two for *Normative Beliefs*) were dropped by insufficient factor loadings. The remainder ones achieved satisfactory scores (i.e., factor loadings at least 0.50). Appendix 4 reports PLS results for the two EPGs.
	• Discriminant validity of instruments	• PLS (Chin, 2010) statistical method was applied. The square root of each AVE (average variance extracted) for each construct was at least 0.71 and greater than the correlations among constructs. Appendix 4 reports PLS results for the two EPGs.
Internal	• History	• Each participant was asked to participate in a single session between 55–75 min without interruptions and null communication with other participants.
	• Testing	• No pre-test was applied, and no one of the participants was informed previously on the experimental materials.
	• Statistical regression	• Participants were not selected by a particular score on agile methods. Participation was voluntary based on inclusion criteria.
	• Selection	• The within-subjects experiment design eliminates differences between subjects. Each participant acts as his/her control group.
Conclusion	• Reliability of measures	• PLS (Chin, 2010) statistical method was applied. The composite reliability index was at least 0.90 for 5 of the six variables and 0.75 and 0.87 for the remaining one in the Scrum EPG and Scrum + EPG, respectively. Appendix 4 reports PLS results for the two EPGs.
	• Random heterogeneity of subjects	• The within-subjects experiment design eliminates differences between subjects. Each participant acts as his/her control group.
	• Inadequate statistical tests	• A Wilcoxon matched-pairs signed-rank test was applied as it is recommended for repeated measures (also known as within-subjects) experimental design.
	• Small sample size	• The non-parametric tests perform well with small samples in the range of 15–20 subjects.
External	• Interaction of testing-treatments	• No pre-test was applied.
	• Interaction of selection-treatments	• The order of application of treatments was randomly selected for each participant.
	• Interaction of setting-treatments	• Each experimental task was conducted with similar agreed conditions. No rewards. Voluntary participation. Same experimental materials. No training on materials.

affords using this standard, and the most used agile practice (i.e., Scrum) had been pursued and released through an Electronic Process Guide (EPG).

This research has produced a reconciled Scrum + EPG, and it was evaluated from its validity of content and usability metrics respectively by a panel of eight experts, and a group of 32 international evaluators (including experts and novice academics, and experts and novice professionals). The two evaluations produced satisfactory statistical results, implying that the newly elaborated Scrum + EPG was assessed with adequate theoretical content validity, as well as with adequate metrics of usability (i.e., usefulness, ease of use, compatibility, value, normative beliefs, and attitude). Consequently, we consider that this research provides contributions to the software engineering domain and practice and provides a high-quality reconciliation of Scrum and the Project Management process of the ISO/IEC 29110 standard-Entry profile. Overall, this research introduces an objective process for systematically elaborating reconciled practices, and it provides an online and free-access EPG of the new Scrum + EPG.

Agility has helped small and very small organizations to have more predictable product delivery, schedule, and costs. However, there are several weaknesses of agility practices such as poor resource planning, limited documentation, and no explicit support for continuous improvement. Moreover, agility does not promote software architecture design, which helps to take better architecture decisions and as a consequence obtain higher-quality software. We, therefore, hope that this research help to make agile practices more robust.

Regarding the use of EGPs, we do not suggest that an organization needs to use an EGP to successfully reconcile Scrum and the ISO/IEC 29110 standard. Rather, we see an EGP as a tool that can be useful for an organization to adopt such a reconciliation although other means for achieving this purpose could be employed (e.g., other ways of documenting the organization processes).

Hence, we can report that the pursued research goal of *Analyze < two Scrum processes packaged in two EPGs > for the purpose of < evaluating them > with respect to their < six usability metrics > from the point of view of the < dual academic-professional perspective > in the context of < international academic-professional participants in an off-line experimental mode >* was successfully achieved.

As future work, we recommend the following research avenues: (1) to replicate the experimental comparative evaluation of the two EPGs for obtaining more definitory results; (2) to conduct a comparative analysis blocking the four groups (i.e., expert-academic, novice-academic, expert-professional, and novice-professional) to identify characteristics associated with their usability metrics; (3) to refine and polish the Scrum + EPG for producing an ISO/IEC 29110 technical report and a complimentary guide document for very small organizations; (4) to investigate the utilization of the Means-Ends Analysis for producing customized new software process; and (5) to investigate the automatization of the Means-Ends Analysis for assisting software process design teams in the elaboration of a customized reconciled agile-rigorous software process.

Finally, we encourage the research community to advance on these research avenues for helping very small software development organizations to achieve their expected *Iron Triangle* aim.

Appendix 1. Instruments

Usefulness metric statements:

- Item USEF#1. If I were to use the (Scrum + EPG|Scrum EPG), it would enable me to accomplish my agile project management tasks more quickly.
- Item USEF#2. If I were to use the (Scrum + EPG|Scrum EPG), the quality of my work of agile project management would improve.
- Item USEF#3. If I were to use the (Scrum + EPG|Scrum EPG), it would enhance my effectiveness on the job related to agile project management.

Ease of use metric statements:

- Item EOU#1. Learning to use the (Scrum + EPG|Scrum EPG) would be easy for me.
- Item EOU#2. If I were to use the (Scrum + EPG|Scrum EPG), it would be easy to operate.
- Item EOU#3. If I were to use the (Scrum + EPG|Scrum EPG), it would be difficult to use. (This statement is codified intentionally in reverse scale).

Compatibility metric statements:

- Item COMP#1. If I were to use the (Scrum + EPG|Scrum EPG), it would be compatible with most aspects of my work related to agile project management.
- Item COMP#2. If I were to use the (Scrum + EPG|Scrum EPG), it would fit my work style related to agile project management.
- Item COMP#3. If I were to use the (Scrum + EPG|Scrum EPG), it would fit well with the way I like to work related to agile project management.

Value metric statements:

- Item VAL#1. The value for saving money by using the (Scrum + EPG|Scrum EPG) for agile project management tasks is:
- Item VAL#2. The value for saving valuable time by using the (Scrum + EPG|Scrum EPG) for agile project management tasks is:
- Item VAL#3. The value for being able to locate a wide variety of agile project management data, information, and knowledge by using the (Scrum + EPG|Scrum EPG) is:
- Item VAL#4. In overall, the value of using the (Scrum + EPG|Scrum EPG) for agile project management tasks is:

Normative beliefs metric statements:

- Item NBEF#2. My close friends think I should use (Scrum + EPG|Scrum EPG) for agile project management tasks.
- Item NBEF#3. My immediate supervisor thinks I should use (Scrum + EPG|Scrum EPG) for agile project management tasks.
- Item NBEF#5. My IT Department thinks I should use (Scrum + EPG|Scrum EPG) for agile project management tasks.
- Item NBEF#6. Other IT specialists in my work organization think I should use (Scrum + EPG|Scrum EPG) for agile project management tasks.

Attitude metric statements:

- Item ATT#1. All considered things, using (Scrum + EPG|Scrum EPG) in my job within the next 6 months would be [extremely negative … extremely positive]
- Item ATT#2. All considered things, using (Scrum + EPG|Scrum EPG) in my job within the next 6 months would be [extremely bad … extremely good]
- Item ATT#3. All considered things, using (Scrum + EPG|Scrum EPG) in my job within the next 6 months would be [extremely harmful … extremely beneficial]

Appendix 2. Means-ends analysis design of the scrum + EPG

A conceptual design research method from Gregor and Hevner (2013), Hevner et al. (2004), and Mora et al. (2008) and augmented with the Means-Ends Analysis method (Anderson, 1993; Simon & Newell, 1971) was applied for the design of the Scrum + EPG. The applied conceptual design (CD) research method consisted of the following activities:

CD.1 Knowledge Gap Identification, CD.2 Design Work Planning, CD.3 Conceptual Design, CD.4 Design Evaluation, and CD.5 Analysis and Synthesis of Design Results. Seven design research guidelines reported in Hevner et al. (2004) and fitted the Design Science Research Knowledge Contribution Framework (DSRKCF) as an exaptation type of contribution (Gregor & Hevner, 2013) were also applied. A knowledge contribution of type exaptation refers to the adaptation of known solutions (i.e., the EPG artifact) to new problems (i.e., the agility-rigor reconciliation problem).

The design research problem formulation was stated as follows: "is it feasible to design and build the artifact [X] = {Scrum + EPG} what fulfill the set of design objectives [DO] = {DO.1 = Scrum + EPG is theoretically valid; DO.2 = Scrum + EPG must reach at least a high compliance level with the Project Management process of the ISO/IEC 29110 standard-Entry profile -; DO.3 = Scrum + EPG must be still perceived as an agile method by practitioners (i.e., Scrum + EPG must not lose its agile essence); and DO.4 = Scrum + EPG is perceived with adequate levels of usefulness, ease of use, compatibility, value, normative beliefs, and attitude} and design restrictions [DRs] = {DR.1 = Scrum + EPG is designed, built and evaluated in a 3-year period; DR.2 = Scrum + EPG must be designed, built and evaluated with the assigned research budget)} by using a design process [DP] = {design approach DA = heuristic; design method DM = Means-Ends Analysis; design knowledge DK = {DK.1 = Scrum practices; DK.2 = Project Management process of the ISO/IEC 29110 standard-Entry profile; DK.3 = EPG design recommendations}; design parameters DPs = {DP.1 = roles; DP.2 = activities-tasks; DP.3 = artifacts}} ?"

The design process (DP) applied a heuristic design approach (DA). Neither analytic nor axiomatic design approaches could be used because no design equations or logical rules were available. Thus, a heuristic design approach (practical design recommendations based on the joint research team expertise) was applied. The design method DM was the Means-Ends Analysis (Anderson, 1993; Simon & Newell, 1971). Means-Ends Analysis is an iterative problem-solving heuristic method that represents a design problem as a non-desired initial set of attributes (called *initial state*), and its solution as a desired final set of attributes (called *goal state*). Means-Ends Analysis, thus, requires defining a set of valid actions (called *operators*) to be applied for transforming (called *transformation sequence path*) the *initial state* to the *goal state*. The set of all possible states including the initial and goal ones is called the *state space*. The set of all *operators* is called the *action space*. A solution can be satisfactory or optimal. The first case implies that is not known the best one *path* and it is not possible to search for all plausible paths by time or cost restrictions and a near state to the *goal state* was reached. This last case implies also that the designer is aware of the practical unfeasibility to find an optimal solution and he/she accepts a sub-optimal one but that satisfies also the minimal expected set of pre-established criteria. The second case implies that was found the best *path* regarding a set of pre-established criteria on economic costs, duration of actions, and other related relevant metrics. The application of a heuristic design approach and the Means-Ends Analysis consisted essentially in the application of *operators* that reduce the difference between the *current state* and the *goal state*.

Further details of the application of the Means-Ends Analysis method to the design of the Scrum + EPG are available upon request to the corresponding author.

Appendix 3. Instructions to international evaluators

PRESENTATION

You have been contacted because you qualify as a practitioner or academic interested and skilled in using/knowing agile software development methods (in particular Scrum) and the ISO/IEC 29110 or a related software engineering standard. We thank you for the asked evaluation on Scrum +, which is an agile project management methodology enhanced from the original Scrum for reaching an estimated overall compliance level of 90% with the project management process of the Entry profile of ISO/IEC 29110. The goal of 90% is for keeping its agility essence but at the same time to cover some identified gaps. It must be remarked that a full compliance level (100%) was not pursued by Scrum + EPG because it would cause the loss of its agility approach. Thanks very much in advance for conducting the asked evaluations for Scrum + EPG. We also thank the complimentary evaluation (for comparative research purposes) of the official Scrum EPG elaborated by EPF Consortium

INSTRUCTIONS

1. Please download the attached file containing the questionnaires:

- QUESTIONNAIRES.DOCX

2. Please, you must evaluate the two Scrum process-EPGs (original Scrum EPG and Scrum + EPG). For doing it, we need you to select the order randomly (i.e., some people will evaluate first the Scrum EPG and after it the Scrum + EPG, and other ones in the opposite order; it is required for statistical validation purposes). For it, we ask you the favor of flipping a coin and:

2.1 If the result is HEAD then please navigate firstly in the Scrum + EPG application for 20 min at least and 30 min at most by using any internet browser (Explorer, Firefox, Chrome, etc.) and please register the number of minutes finally spent in this task. You can access it at the next URL: http://x3620a-labdc.uaa.mx/scrum_plus/

After it, please close the browser and take a 15-min break. Then, please navigate also 20 min at least and 30 min at most (and please also register the exact number of minutes spent in this task) in the other Scrum EPG at: http://epf.eclipse.org/wikis/scrum/

Once completed these 2 navigations on Scrum + EPG and Scrum EPG, please continue with step #3

2.2 If the result is TAIL, please navigate firstly in the original Scrum EPG application for 20 min at least and 30 min at most by using any internet browser (Explorer, Firefox, Chrome, etc.) and please register the number of minutes finally spent in this task. You can access it at the next URL: http://epf.eclipse.org/wikis/scrum/.

After it, please close the browser and take a 15-min break. Then, please navigate also 20 min at least and 30 min at most (and please also register the exact number of minutes spent in this task) in the other Scrum + EPG at: http://x3620a-labdc.uaa.mx/scrum_plus/

Once completed these 2 navigations on Scrum + EPG and Scrum EPG, please continue with step #3

3. Now, we ask you to fill the USABILITY and DEMOGRAPHIC questionnaires. Both are in the same word document entitled QUESTIONNAIRES.DOCX. This task is estimated at 30 min. We thank you answer all questions for both Scrum + EPG and Scrum EPG.

4. Thanks very much for returning to us the filled questionnaires to the email (X) on or before (date).

Appendix 4. PLS results for the threats to construct validity

The validity and reliability of the six metrics of usability were assessed respectively with the convergent validity of the factor loadings, the discriminant validity, and the composite reliability index (CRI) of the PLS statistical method (Chin, 2010). Three statements (one for usefulness and two for normative beliefs) were dropped by insufficient factor loadings. The remainder ones achieved satisfactory scores (i.e., factor loadings at least 0.50). For the discriminant validity, the square root of each AVE (average variance extracted) of each construct was greater than the correlations among constructs. It is verified in the correlation matrix where the values in the diagonal (i.e., the square roots of the AVEs) must be at least 0.71 and greater than the other values in the off-diagonal line. Regarding the CRI, all of the six metrics of usability achieved also satisfactory scores (i.e., at least 0.70). Tables 7 and 8 report the results for Scrum EPG and Scrum + EPG, respectively. These values were obtained with the free-100-data license of the software tool SmartPLSv3 (https://www.smartpls.com).

Table 7 Validity and reliability of usability metrics for Scrum EPG

	Item statement code	Usability metrics					
		Usefulness	Ease of use	Compatibility	Value	Normative beliefs	Attitude
CRI ≥ 0.70		0.75	0.91	0.93	0.92	0.92	0.98
Convergent validity of concepts (Factor loadings ≥ 0.50)	USEF#1	0.50	0.30	0.29	0.09	0.24	0.09
	USEF#2	0.88	0.14	0.42	0.42	0.46	0.42
	USEF#3	0.71	-0.08	-0.02	0.25	0.29	0.25
	EOU#1	0.19	0.96	0.55	0.31	0.27	0.47
	EOU#2	0.12	0.96	0.63	0.19	0.23	0.34
	EOU#3	-0.15	0.71	0.22	-0.09	0.02	0.34
	COMP#1	0.26	0.53	0.85	0.40	0.20	0.33
	COMP#2	0.43	0.44	0.92	0.35	0.42	0.36
	COMP#3	0.23	0.58	0.93	0.39	0.42	0.37
	VAL#1	0.48	0.06	0.31	0.83	0.39	0.30
	VAL#2	0.29	0.35	0.31	0.85	0.58	0.34
	VAL#3	0.23	-0.05	0.18	0.82	0.51	0.44
	VAL#4	0.37	0.28	0.54	0.93	0.67	0.54
	NBEF#2	0.39	0.12	0.13	0.54	0.81	0.37
	NBEF#3	0.38	0.10	0.13	0.62	0.84	0.54
	NBEF#5	0.45	0.30	0.45	0.46	0.92	0.59
	NBEF#6	0.45	0.24	0.60	0.55	0.86	0.54
	ATT#1	0.27	0.29	0.33	0.52	0.62	0.97
	ATT#2	0.27	0.51	0.43	0.39	0.64	0.97
	ATT#3	0.31	0.44	0.40	0.46	0.50	0.95
Discriminant validity (square root of AVE ≥ 0.71)	Usefulness	0.71	0.12	0.33	0.42	0.49	0.29
	Ease of use	0.12	0.89	0.58	0.21	0.23	0.43
	Compatibility	0.33	0.58	0.90	0.42	0.39	0.40
	Value	0.42	0.21	0.42	0.86	0.63	0.48
	Normative beliefs	0.49	0.23	0.39	0.63	0.86	0.58
	Attitude	0.29	0.43	0.40	0.48	0.58	0.96

Table 8 Validity and reliability of usability metrics for Scrum + EPG

	Item statement code	Usability metrics					
		Usefulness	Ease of use	Compatibility	Value	Normative beliefs	Attitude
CRI ≥ 0.70		0.87	0.92	0.93	0.91	0.93	0.97
Convergent validity of concepts (Factor loadings ≥ 0.50)	USEF#1	0.89	0.54	0.44	0.65	0.46	0.26
	USEF#2	0.80	0.10	0.23	0.43	0.27	0.17
	USEF#3	0.82	0.45	0.51	0.44	0.21	0.42
	EOU#1	0.45	0.83	0.46	0.47	0.30	0.35
	EOU#2	0.42	0.92	0.66	0.49	0.47	0.59
	EOU#3	0.39	0.91	0.74	0.66	0.49	0.64
	COMP#1	0.37	0.56	0.88	0.67	0.30	0.36
	COMP#2	0.46	0.65	0.91	0.58	0.45	0.46
	COMP#3	0.45	0.71	0.91	0.70	0.37	0.64
	VAL#1	0.56	0.48	0.57	0.85	0.40	0.28
	VAL#2	0.55	0.66	0.69	0.92	0.46	0.39
	VAL#3	0.40	0.27	0.51	0.72	0.45	0.41
	VAL#4	0.60	0.62	0.67	0.88	0.63	0.56
	NBEF#2	0.31	0.45	0.20	0.41	0.72	0.35
	NBEF#3	0.21	0.31	0.22	0.40	0.91	0.64
	NBEF#5	0.44	0.52	0.50	0.63	0.94	0.69
	NBEF#6	0.44	0.46	0.48	0.56	0.90	0.57
	ATT#1	0.23	0.49	0.43	0.38	0.71	0.94
	ATT#2	0.30	0.62	0.56	0.47	0.66	0.99
	ATT#3	0.43	0.67	0.62	0.55	0.55	0.95
Discriminant validity (square root of AVE ≥ 0.71)	Usefulness	0.83	0.46	0.48	0.63	0.40	0.33
	Ease of use	0.46	0.89	0.72	0.62	0.49	0.62
	Compatibility	0.48	0.72	0.90	0.73	0.42	0.56
	Value	0.63	0.62	0.73	0.85	0.58	0.49
	Normative beliefs	0.40	0.49	0.42	0.58	0.87	0.67
	Attitude	0.33	0.62	0.56	0.49	0.67	0.96

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11219-021-09552-3.

References

Abad, Z., Alipour, A., & Ramsin, R. (2012). Enhancing tool support for situational engineering of agile methodologies in eclipse. In R. Lee (Ed.), *Software Engineering Research, Management and Applications* (pp. 141–152). Springer.

Abrahamsson, P., Oza, N., & Siponen, M. T. (2010). Agile software development methods: A comparative review. In T. Dingsøyr, T. Dybå, & N. Moe (Eds.), *Agile software development* (pp. 31–59). Springer.

Agarwal, N., & Rathod, U. (2006). Defining 'success' for software projects: An exploratory revelation. *International Journal of Project Management, 24*(4), 358–370.

Ahimbisibwe, A., Daellenbach, U., & Cavana, R. (2017). Empirical comparison of traditional plan-based and agile methodologies—Critical success factors for outsourced software development projects from vendors' perspective. *Journal of Enterprise Information Management, 30*(3), 400–453.

Anderson, J. (1993). Problem solving and learning. *American Psychologist, 48*(1), 35–44.

Basili, V. R., Briand, L. C., & Melo, W. L. (1996). How reuse influences productivity in object-oriented systems. *Communications of the ACM, 39*(10), 104–116.

Bass, J. M., Allison, I. K., & Banerjee, U. (2013). Agile method tailoring in a CMMI level 5 organization. *Journal of International Technology and Information Management, 22*(4), 78–93.

Beck, K. (1999). Embracing change with extreme programming. *Computer, 32*(10), 70–77.

Beck, K., & Andres, C. (2004). *Extreme programming explained: Embrace change*. Addison Wesley Professional.

Becker-Kornstaedt, U. (2000). A strategy for the integration of software process support technology into organizations. In Proceedings of ICSE workshop 'SE overthe Internet', Limerick, Ireland, pp. 1–6.

Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software, 80*(4), 571–583.

Berger, P., & Maurer, G. (2018). *Experimental Design with Applications in Management, Engineering, and the Sciences*. Springer.

Bloch, M., Blumberg, S., & Laartz, J. (2012). Delivering large-scale IT projects on time, on budget, and on value. *McKinsey on Business Technology, 27,* 1–7.

Boehm, B., & Turner, R. (2003). Using risk to balance agile and plan-driven methods. *Computer, 36*(6), 57–66.

Boehm, B., & Turner, R. (2004). Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods. In Proceedings of the 26th International Conference on Software Engineering, Edinburgh, Scotland, pp. 718–719.

Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, *22*(5), 30–39.

Briand, L. C., Bunse, C., & Daly, J. W. (2001). A controlled experiment for evaluating quality guidelines on the maintainability of object-oriented designs. *IEEE Transactions on Software Engineering, 27*(6), 513–530.

Buchalcevova, A. (2019). Using ArchiMate to model ISO/IEC 29110 standard for very small entities. *Computer Standards & Interfaces, 65,* 103–121.

Cayola, L., & Macías, J. A. (2018). Systematic guidance on usability methods in user-centered software development. *Information and Software Technology, 97,* 163–175.

Chin, W. (2010). How to write up and report PLS analyses. In V. Esposito-Vinzi, W. Chin, J. Henseler, & H. Wang (Eds.), *Handbook of Partial Least Squares* (pp. 655–690). Springer.

Clarke, P., & O'Connor, R. V. (2013). An empirical examination of the extent of software process improvement in software SMEs. *Journal of Software: Evolution and Process, 25*(9), 981–998.

CMMI® Institute. (2019). CMMI® for Development v2.0. https://cmmiinstitute.com/products/cmmi/cmmi-v2-products. Accessed 1st March 2019.

Coleman, G., & O'Connor, R. (2008). Investigating software process in practice: A grounded theory perspective. *Journal of Systems and Software, 81*(5), 772–784.

Dingsøyr, T., Moe, N., Dybå, T., & Conradi, R. (2004). A workshop-oriented approach for defining electronic process guides—a case study. In Proceedings of the 11th Norwegian Conference on Information Systems, Stavanger, Norway, pp. 10–25.

Ebert, C. (2007). The impacts of software product management. *Journal of Systems and Software, 80*(6), 850–861.

Eclipse Foundation. (2019). Scrum EPG version 1.5 https://www.eclipse.org/downloads/download.php?file=/technology/epf/Scrum/library/scrum_library_1.5_20080820.zip. Accessed 1st March 2019.

Fowler, M., & Highsmith, J. (2001). *The agile manifesto. Software Development, 9*(8), 28–35.

Galvan-Cruz, S., Mora, M., O'Connor, R. V., Acosta, F., & Álvarez, F. (2017a). An objective compliance analysis of project management process in main agile methodologies with the ISO/IEC 29110 entry profile. *International Journal of Information Technologies and Systems Approach, 10*(1), 75–106.

Galvan-Cruz, S., Mora, M., & O'Connor, R. (2017b). A Means-Ends Design of Scrum+: An agile-disciplined balanced Scrum enhanced with the ISO/IEC 29110 Standard. In J. Mejia, M. Muñoz, A. Rocha, Y. Quiñonez, & J. Calvo-Manzano (Eds.), *Trends and Applications in Software Engineering (CIMPS 2017)* (pp. 13–23). Springer.

Galván-Cruz, S., Muñoz, M., Mejía, J., Laporte, C.Y., & Negrete, M. (2021). Building a Guideline to Reinforce Agile Software Development with the Basic Profile of ISO/IEC 29110 in Very Small Entities. In J. Mejia, M. Muñoz, A. Rocha, and Y. Quiñonez (Eds.) New Perspectives in Software Engineering. CIMPS 2020. Advances in Intelligent Systems and Computing, vol 1297. Springer, Cham.

Garcia, F., Vizcaino, A., & Ebert, C. (2011). Process management tools. *IEEE Software, 28*(2), 15–18.

Garzás, J., & Paulk, M. C. (2013). A case study of software process improvement with CMMI-DEV and Scrum in Spanish companies. *Journal of Software: Evolution and Process, 25*(12), 1325–1333.

Ghanadbashi, S., & Ramsin, R. (2016). Towards a method engineering approach for business process reengineering. *IET Software, 10*(2), 27–44.

Glass, R. L., Ramesh, V., & Vessey, I. (2004). An analysis of research in computing disciplines. *Communications of the ACM, 47*(6), 89–94.

Gregor, S., & Hevner, A. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly, 37*(2), 337–355.

Hauck, J., von Wangenheim, C., de Souza, R., & Thiry, M. (2008). Process reference guides—Support for improving software processes in alignment with reference models and standards. In Proceedings of the 15th EuroSPI conference on European Systems and Software Process Improvement and Innovation, Dublin, Ireland, pp. 1–12.

Haumer, P. (2007). Eclipse process framework composer (parts I and II). Eclipse Foundation. https://www.eclipse.org/epf/general/EPFComposerOverviewPart1-2.pdf. Accessed 1st March 2019.

Henriques, V., & Tanner, M. (2017). A systematic literature review of agile and maturity model research. *Interdisciplinary Journal of Information, Knowledge, and Management, 12,* 53–73.

Hevner, A., March, S. T., Park, J., & Ram, S. (2004). Design science research in information systems. *MIS Quarterly, 28*(1), 75–105.

Hoda, R., Salleh, N., & Grundy, J. (2018). The rise and evolution of agile software development. *IEEE Software, 35*(5), 58–63.

Hong, W., Thong, J. Y., Chasalow, L. C., & Dhillon, G. (2011). User acceptance of agile information systems: A model and empirical test. *Journal of Management Information Systems, 28*(1), 235–272.

Humphrey, W. S. (1999). Pathways to process maturity: The personal software process and team software process. *SEI Interactive, 2*(2), 1–17.

Humphrey, W. (2000). The Team Software Process (TSP). Software Engineering Institute. https://resources.sei.cmu.edu/asset_files/TechnicalReport/2000_005_001_13754.pdf. Accessed 1st March 2019.

ISO, IEC. . (2012). *ISO/IEC TR 29110-5-1-1:2012 Software engineering—Lifecycle profiles for very small entities (VSEs) Part 5-1-1: Management and engineering guide: generic profile group: Entry profile*. International Organization for Standardization.

ISO, IEC. . (2015). *ISO/IEC 33004:2015 Information technology—Process assessment—Requirements for process reference, process assessment and maturity models*. International Organization for Standardization.

ISO, IEC. . (2017). *ISO/IEC/IEEE 12207:2017 Systems and software engineering—Software life cycle processes*. International Organization for Standardization.

Jakobsen, C. R., & Johnson, K. A. (2008). Mature Agile with a Twist of CMMI. In Proceedings of the Agile 2008 Conference, Toronto, Canada, pp. 212–217.

Jedlitschka, A., Ciolkowski, M., & Pfahl, D. (2008). Reporting controlled experiments in software engineering. In F. Shull, J. Singer, & D. Sjøberg (Eds.), *Guide to advanced empirical software engineering* (pp. 201–228). Springer.

Jirapanthong, W. (2019). Experience in applying of ISO 29110 to agile software development. *Journal of Information Science and Technology, 9*(1), 63–70.

Johanson, A. N., & Hasselbring, W. (2017). Effectiveness and efficiency of a domain-specific language for high-performance marine ecosystem simulation: A controlled experiment. *Empirical Software Engineering, 22*(4), 2206–2236.

Karahanna, E., Straub, D. W., & Chervany, N. L. (1999). Information technology adoption across time: A cross-sectional comparison of pre-adoption and post-adoption beliefs. *MIS Quarterly, 23*(2), 183–213.

Kellner, M. I., Becker-Kornstaedt, U., Riddle, W. E., Tomal, J., & Verlage, M. (1998). Process guides: Effective guidance for process participants. In Proceedings of the Fifth International Conference on the Software Process, Illinois, USA, pp. 11–25.

Klotins, E., Unterkalmsteiner, M., & Gorschek, T. (2019). Software engineering in start-up companies: An analysis of 88 experience reports. *Empirical Software Engineering, 24*(1), 68–102.

Koolmanojwong, S., Aroonvatanaporn, P., & Charoenthongtrakul, I. (2008). Incremental commitment model process guidelines for software engineering class. *USC CSSE Technical Report, 2008–832,* 1–12.

Laporte, C., O'Connor, R., & Fanmuy, G. (2013a). International systems and software engineering standards for very small entities. *CrossTalk, The Journal of Defense Software Engineering, 26*(3), 28–33.

Laporte, C. Y., Séguin, N., Villas Boas, G., & Buasung, S. (2013b). Small tech firms: Seizing the benefits of software and systems engineering standards. ISO Focus+, 4(2), 32–36.

Laporte, C., & O'Connor, R. (2017). Software process improvement standards and guides for very small organization: An overview of eight implementations. *CrossTalk, The Journal of Defense Software Engineering, 30*(3), 23–27.

Laporte, C. (2019). Deployment packages guidelines. http://profs.etsmtl.ca/claporte/english/VSE/VSE-packages.html. Accessed 1st March 2019.

Larrucea, X., & Fernandez-Gauna, B. (2019). A mapping study about the standard ISO/IEC29110. *Computer Standards & Interfaces, 65,* 159–166.

Leuser, J., Porta, N., Bolz, A., & Raschke, A. (2009). Empirical validation of a requirements engineering process guide. In Proceedings of the 13th International Conference on Evaluation and Assessment in Software Engineering, Durham University, UK, pp. 1–10.

Magdaleno, A. M., Werner, C. M. L., & De Araujo, R. M. (2012). Reconciling software development models: A quasi-systematic review. *Journal of Systems and Software, 85*(2), 351–369.

Majchrowski, A., Ponsard, C., Saadaoui, S., Flamand, J., & Deprez, J. C. (2016). Software development practices in small entities: An ISO29110-based survey. *Journal of Software: Evolution and Process, 28*(11), 990–999.

Marques, R., Costa, G., Silva, M., & Gonçalves, P. (2017). A survey of failures in the software development process. In Proceedings of the 25th European Conference in Information Systems, Guimarães, Portugal, pp. 2445–2459.

MaxStatLite. (2019). MaxStatLite tool. http://www.maxstatlite.com. Accessed 1st March 2019.

Moe, N. B., & Dybå, T. (2006). The use of an electronic process guide in a medium-sized software development company. *Software Process: Improvement and Practice, 11*(1), 21–34.

Mora, M., Gelman, O., Paradice, D., & Cervantes, F. (2008). The case for conceptual research in information systems. In Proceedings of the CONF-IRM 2008, Niagara Falls, Canada, pp. 1–10.

Mora, M., Rory, V. O., Rainsinghani, M., & Gelman, O. (2016). Impacts of electronic process guides by types of user: An experimental study. *International Journal of Information Management, 36*(1), 73–88.

Muñoz, M., Mejia, J., & Laporte, C. Y. (2018). Reinforcing very small entities using agile methodologies with the ISO/IEC 29110. In Proceedings of the International Conference on Software Process Improvement, Guadalajara, Mexico, pp. 88–98.

Niazi, M. (2015). A comparative study of software process improvement implementation success factors. *Journal of Software: Evolution and Process, 27*(9), 700–722.

Niazi, M., Wilson, D., & Zowghi, D. (2005). A maturity model for the implementation of software process improvement: An empirical study. *Journal of Systems and Software, 74*(2), 155–172.

O'Connor, R., & Coleman, G. (2009). Ignoring "Best Practice": Why Irish software SMEs are rejecting CMMI and ISO 9000. *Australasian Journal of Information Systems, 16*(1), 7–30.

O'Connor, R. V., & Laporte, C. Y. (2017). The evolution of the ISO/IEC 29110 set of standards and guides. *International Journal of Information Technologies and Systems Approach, 10*(1), 1–21.

Pai, D. R., Subramanian, G. H., & Pendharkar, P. C. (2015). Benchmarking software development productivity of CMMI level 5 projects. *Information Technology and Management, 16*(3), 235–251.

Palomino, M., Dávila, A., Melendez, K., & Pessoa, M. (2016). Agile practices adoption in CMMI organizations: A systematic literature review. In Proceedings of the International Conference on Software Process Improvement, Aguascalientes, Mexico, pp. 57–67.

Pasini, A., Esponda, S., Boracchia, M., & Pesado, P. (2013). Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110. In Proceedings of the XVIII Congreso Argentino de Ciencias de la Computación, Bahía Blanca, Argentina, pp. 898–909.

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2008). Systematic mapping studies in software engineering. In Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE), University of Bari, Italy, pp. 1–10.

Pikkarainen, M. (2009). Towards a better understanding of CMMI and agile integration-multiple case study of four companies. In Proceedings of the International Conference on Product-Focused Software Process Improvement, Oulu, Finland, pp. 401–415.

Pino, F. J., Pedreira, O., García, F., Luaces, M. R., & Piattini, M. (2010). Using Scrum to guide the execution of software process improvement in small organizations. *Journal of Systems and Software, 83*(10), 1662–1677.

Richardson, I., & Von Wangenheim, C. G. (2007). Guest editors' introduction: Why are small software organizations different? *IEEE Software, 24*(1), 18–22.

Riemenschneider, C. K., Hardgrave, B. C., & Davis, F. D. (2002). Explaining software developer acceptance of methodologies: A comparison of five theoretical models. *IEEE Transactions on Software Engineering, 28*(12), 1135–1145.

Savolainen, P., Ahonen, J. J., & Richardson, I. (2012). Software development project success and failure from the supplier's perspective: A systematic literature review. *International Journal of Project Management, 30*(4), 458–469.

Schwaber, K. (1997). Scrum development process. In the Proceedings of the OOPSLA'95, Austin, USA, pp. 117–134.

Schwaber, K., & Sutherland, J. (2017) The definitive guide to scrum: The rules of the game. https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf. Accessed 1st March 2019.

Sheskin, D. (2000). Handbook of parametric and nonparametric statistical procedures. Boca Raton, Florida, USA, Chapman & Hall/CRC.

Silva, F. S., Soares, F. S. F., Peres, A. L., de Azevedo, I. M., Vasconcelos, A. P. L., Kamei, F. K., & de Lemos Meira, S. R. (2015). Using CMMI together with agile software development: A systematic review. *Information and Software Technology, 58,* 20–43.

Simon, H., & Newell, A. (1971). Human problem solving: the state of the theory in 1970. *American Psychology, 26*(2), 145–159.

Slaughter, S. A., Harter, D. E., & Krishnan, M. S. (1998). Evaluating the cost of software quality. *Communications of the ACM, 41*(8), 67–73.

Standish Group. (2015). Chaos Report 2015. https://www.infoq.com/articles/standish-chaos-2015. Accessed 1st March 2019.

Staples, M., Niazi, M., Jeffery, R., Abrahams, A., Byatt, P., & Murphy, R. (2007). An exploratory study of why organizations do not adopt CMMI. *Journal of Systems and Software, 80*(6), 883–895.

Suteeca, K., & Ramingwon, S. (2016). A framework to apply ISO/IEC29110 on Scrum. In Proceedings of the Computer Science and Engineering Conference, Chiang Mai, Thailand, pp. 1–5.

Sutherland, J. (2010). Scrum handbook. Scrum Training Institute.

Sutherland, J., Jakobsen, C., & Johnson, K. (2008). Scrum and CMMI level 5: The magic potion for code warriors. In Proceedings of the 41st Annual Hawaii International Conference on Systems Sciences (HICSS), Waikoloa, USA, pp. 466–466.

Takeuchi, M., Kohtake, N., Shirasaka, S., Koishi, Y., & Shioya, K. (2014). Report on an assessment experience based on ISO/IEC 29110. *Journal of Software: Evolution and Process, 26*(3), 306–312.

Torrecilla-Salinas, C. J., Sedeño, J., Escalona, M. J., & Mejías, M. (2016). Agile, web engineering and capability maturity model integration: A systematic literature review. *Information and Software Technology, 71,* 92–107.

Unterkalmsteiner, M., Gorschek, T., Islam, A. M., Cheng, C. K., Permadi, R. B., & Feldt, R. (2011). Evaluation and measurement of software process improvement—A systematic literature review. *IEEE Transactions on Software Engineering, 38*(2), 398–424.

Vegas, S., Apa, C., & Juristo, N. (2015). Crossover designs in software engineering experiments: Benefits and perils. *IEEE Transactions on Software Engineering, 42*(2), 120–135.

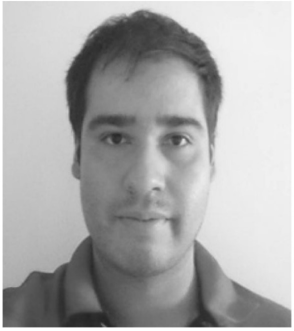
VersionOne. (2018). The 12th Annual State of Agile Survey Report. http://stateofagile.versionone.com. Accessed 1st March 2019.

Von Wangenheim, C. G., Thiry, M., & Kochanski, D. (2009). Empirical evaluation of an educational game on software measurement. *Empirical Software Engineering, 14*(4), 418–452.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). Experimentation in Software Engineering. Springer.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Sergio Galvan-Cruz is the Chief Executive Officer in TECOM since 2007, a Mexican software company focused on small businesses in different Mexican states. He holds an MSc and a Ph.D. in Software Engineering since 2009 and 2017 respectively. Dr. Galvan-Cruz is also a part-time Professor in the Electronic Systems Department at the Autonomous University of Aguascalientes, Mexico.



Manuel Mora is a full-time Professor in the Information Systems Department at the Autonomous University of Aguascalientes (UAA), Mexico. Dr. Mora holds an M.Sc. in Computer Sciences (Artificial Intelligence area, 1989) from Monterrey Tech (ITESM), and an Eng.D. in Engineering (Systems Engineering area, 2003) from the National Autonomous University of Mexico (UNAM). He has published over 90 research papers in international top conferences, research books, and journals such as IEEE-TSMC, European Journal of Operational Research, Int. Journal of Information Management, Engineering Management, Int. J. of Information Technology and Decision Making, Information Technology for Development, Int. J. in Software Engineering and Knowledge Engineering, and Computer Standards & Interface. Dr. Mora is a senior member of ACM (since 2008), an SNI at Level II, and serves in the ERB of several international journals indexed by Emergent Source Citation Index focused on decision-making support systems (DMSS) and IT services systems.

Claude Y. Laporte is a Professor at the École de technologie supérieure (ETS) an engineering school of the Université du Québec network of institutions. He teaches graduate and undergraduate software engineering courses. He did his undergraduate studies at the Collège Militaire Royal de St-Jean (Canadian Military College). He received his Master in Applied Sciences from the Electrical and Computer Engineering department of École Polytechnique de Montréal in 1986. In 1980, he received his Master in Physics (computerized instrumentation) from the Université de Montréal. He completed a Ph.D., under the supervision of Dr. J. Tisseau, Director of the National Engineering School of Brest (ENIB) of the Université de Bretagne Occidentale (UBO). He received a Doctor Honoris Causa from the Faculty of Engineering and Architecture of the Universidad de San Martin de Porres (Lima, Peru). In 1989, he instigated the development of a software engineering center modeled on the Software Engineering Institute. This lead to the establishment of the Applied Software Engineering Center (ASEC) of the Computer Research Institute of Montreal (CRIM). Between 1978 and 1991, he was a professor at the Royal Military College (Saint-Jean), a Military university of the Department of National Defense. He was tasked in 1988, as a project manager, to lead the development of a graduate program in software engineering for the Department of National Defense. From 1992 to

1999, he worked at Oerlikon Aerospace (now Rheinmetall Defense) as a senior analyst responsible to coordinate the development and deployment of software and systems engineering processes and management processes. He participates, as an Affiliate of the Software Engineering Institute, to the project Improving Processes for Small Settings. He co-founded and chaired the Montréal Software Process Improvement Network (SPIN). SPINs are sponsored by the Software Engineering Institute. His interests include the development and deployment of software and systems engineering processes, process assessment, software quality, and the management of technological changes. He is a life member of the IEEE Computer Society (IEEE), the International Council on Systems Engineering (INCOSE) and the Project Management Institute (PMI). He has been an invited professor at the French University of Brest (Université de Bretagne Occidentale) and the Chiang Mai University of Thailand. Since 2005, he is the Project Editor of an ISO Working Group, Working Group 24, of ISO/SC7/JTC1. The working group has been mandated to develop the ISO/IEC 29110 series of Systems engineering and Software engineering standards for Very Small Entities (organizations with up to 25 people). He also represented the Standards Council of Canada (SCC) at Working Group 20 of JTC1/ISO/SC7. Working Group 20 has developed a standard for the international certification of software engineering professionals. He has been the ISO co-project editor of the Guide to the Software Engineering Body of Knowledge (SWEBOK Guide).

Hector Duran-Limon is a Full Professor at the Information Systems Department, University of Guadalajara. He completed a Ph.D. at Lancaster University in 2002. Following this, he was a post-doctoral researcher until December 2003. He obtained an IBM Faculty award in 2008. His research interests include Grid Computing, Adaptive Middleware, and Mobile Computing. He is also interested in software architectures and component-based development. In 2006, He was invited to create a Ph.D. program in Information Technologies for the University of Guadalajara, becoming a member of the Academic-Council. Contact him at the Information Systems Department, University of Guadalajara, Mexico.