A Means-Ends Design of SCRUM+: an agile-disciplined balanced SCRUM enhanced with the ISO/IEC 29110 Standard

Sergio Galván-Cruz[3] , Manuel Mora[2], Rory O'Connor[3]

[1] Autonomous University of Aguascalientes, Electronic Systems, Av. Universidad, 940, Aguascalientes, Mexico, checogc23@gmail.com
[2] Autonomous University of Aguascalientes, Information Systems, Av. Universidad, 940, Aguascalientes, Mexico, dr.manuel.mora.uaa@gmail.com
[3] Dublin City University, School of Computing, Glasnevin, Dublin 9, Ireland, rory.oconnor@dcu.ie

Abstract. Agile systems development methodologies (ASDMs) have gained high acceptance in very small entities (VSEs) of software development seeking quality at minimal effort. SCRUM and XP in industrial settings and UPEDU in academic ones are main of them. Similarly, Software Process Improvement (SPI) initiatives promote the utilization of process frameworks and standards. However, despite both worlds (i.e. ASDMs and SPI) pursue a shared end of high-quality software, both are separated by different underlying approaches. We consider that ASDMs can get benefits from SPI through controlled enhancements (i.e. an agile-discipline balance) without elimination of agility. Thus, in this research, we report the design of SCRUM+, an enhanced SCRUM with recommendations on roles, activities-tasks and artifacts from the SPI standard ISO/IEC 29110. SCRUM+ was designed by using a Means-Ends analysis. Our final aim is to provide such an enhanced SCRUM methodology via an Electronic Process Guide (EPG) to help practitioners for a better use of agile approaches with SPI added recommendations that be found theoretically robust and potentially useful regarding SCRUM from a panel of experts and SCRUM practitioners.

Keywords: SCRUM+, SCRUM, ISO/IEC 29110, SPI, enhanced methodology

1 Introduction

Software process models and standards (SPMSs) such as CMMI-DEV and the ISO/IEC 12207 [1, 2], have been developed by international associations for helping to software development organizations to meet the current demands for quality process and product improvements [3, 4]. SPMSs are important for software

development organizations because their correct implementation has generated relevant benefits such as: process cost reduction, critical software failure reduction, quality product increment, team productivity increment, and customer satisfaction increment among others [5, 6]. Such SPMS are core initiatives (i.e. process frameworks besides best practices and tools [7]) into the Software Process Improvement (SPI) approach. According to [7] SPI refers to "*a systematic approach to increase the efficiency and effectiveness of a software development organization and to enhance software products*". However, according to [4] these SPI initiatives "*were not written for small projects, small development organizations, or companies with between 1 and 25 employees, and are consequently difficult to apply in such settings*". Thus, VSEs (whole business or project teams) while represent a high percentage of software business in the world [8], are unserved potential users by such normal software process standards and models [9, 10]. Given this problematic situation, a new ISO/IEC 29110 software process lifecycle standard [11] was elaborated specifically to VSEs.

In this same context of VSEs, it has been also identified the preference for using Agile Software Development Methodologies (ASDMs) such as: SCRUM and XP in industrial settings [12, 13] and UPEDU in academic ones [14]. Furthermore, ASDMs literature claims [12, 13] similar overall benefits achieved for opposite software development methodologies (i.e. rigor-disciplined ones framed on SPI process frameworks) with the additional advantages provided only by the agile approach [15]. Thus, the ASDMs should be successfully used by VSEs. However, while there are evidences of a high rate of utilization of these agile methodologies [12, 13] by VSEs, it has been also reported that there are some contextual prerequisites for a successful utilization [16, 17]. We consider that these contextual prerequisites for successful adoption of ASDMs refers to the adherence to best scholastic practices provided by SPMS as the SPI approach promotes [16, 17]. Thus, the practitioners of ASDMs need to enhance their ASDM with some recommendations from rigor-disciplined development methods instead of using ASDMs directly [18, 19].

In this research, we report the design of SCRUM+: an enhanced SCRUM [20, 21] with some recommendations from the ISO/IEC 29110 standard. This new standard [11] has been released for VSEs but it was designed independently of the ASDM approach. Hence, thus we pursue to present a more robust balanced agility-disciplined SCRUM method aligned to the recommendations from several literatures [18, 19]. In practical perspective, it provides a dual overall benefit: for SCRUM practitioners to count with a more robust disciplined process enhanced with some critical recommendations from the ISO/IEC 29110 standards, and for the ISO/IEC 29110 community to count with a specific adaptation of SCRUM with a greater coverage of the expected practices to be conducted in the ISO/IEC 29110 standard [11, 22] than SCRUM actually covers [20, 21].

The remainder of this paper continues as follows: the research process is reported in the section 2; the theoretical bases on SCRUM, the ISO/IEC 29110 standard, and the Agility-Discipline debate are reported in the section 3; the application of the Means-Ends method for designing SCRUM+ is presented in section 4; finally, limitations, recommendations and conclusions of this research are reported in section 5.

2 Research Process

2.1 Research Goals and Design Restrictions.

The main overall research goal is: to design an enhanced and agile-disciplined balanced SCRUM+ methodology based on the original SCRUM methodology and best practices provided by the ISO/IEC 29110 (Entry Profile, Project Management process) [11, 22]. Two specific and critical design restrictions are: 1) SCRUM+ must be still perceived as an agile method by practitioners (i.e. it means that SCRUM+ must not lose its agile essence); and 2) SCRUM+ must reach at least a high coverage level with the ISO/IEC 29110 standard (Entry Profile, Project Management process).

2.2 Research Methods and Materials.

The Means-Ends analysis technique [23] was initially elaborated in the early Artificial Intelligence research stream in the 50's [24] as a Problem-Solving technique. According to Newell and Simon [25]: "*Problem solving can be viewed, then, as finding one of the few paths that leads from a problem's initial state to its goal state through some space of possible intermediate states*".

In Means-Ends Analysis technique [25] a problem is a situation faced by a person to reach a desired state (named *Goal State*) from an initial departure point (named *Initial State*) and it is not known in advance the set of actions (named *Operators*) and the sequence of application (named *Path*) on objects (named *Operands*) that must be applied. Thus, to find a *Solution* means to find a sequence of *Operators* applied to *Operands* to transform the *Initial State* in the *Goal State*. A *Solution* can be *Optimal* or *Satisfactory*. For many problems, there are not known or practical feasible algorithms (i.e. a predefined set of actions to be followed for transforming an *Initial State* in a *Goal State*) to be applied for reaching to the *Goal State*. In these cases, the concept of Heuristics is applied [24]. Heuristics are recommendations and clues gained through the experience in similar or related problems that are suggested to be applied (i.e. Heuristics on what *Operators* apply on the *Operands* given a current *State* and the expected *Goal State*). This process, according to [23] has two principal features: 1) Reduction of Differences which is preference of problem solvers to use the *Operators* that produce *States* more similar to the *Goal State*; and 2) Sub-Goaling which happens when a problem can be divided in sub-Problems and thus its final *Solution* can be reached when the *Solution* for all of the Sub-Goals is reached under necessity of all Sub-Goals (i.e. connected by AND logical operator) or at least one *Solution* is reached (i.e. connected by OR logical operator). Hence, thus, we consider that the Means-Ends Analysis technique provides a systematic well-tested method that can be applied for the systematic design of SCRUM+.

In this research, we used the following materials: 1) the ISO/IEC 29110 Entry Profile document [22]; 2) the official guides for SCRUM [20, 21]; 3) a SCRUM book [26]; 4) a SCRUM EPG (Electronic Process Guide) [27]; and 5) a coverage analysis

of SCRUM regarding the ISO/IEC 29110 Entry Profile Project Management process [28].

3 Theoretical Background

3.1 SCRUM Methodology.

SCRUM has been reported as the most used agile methodology [13]. According to [29] "*SCRUM has a project management emphasis. SCRUM has been applied mainly to software projects, but a number of non-software projects have also been managed with SCRUM---the principles are applicable to any projec*t". Moreover, according to [21] "*SCRUM is a framework for developing and sustaining complex products*". The model of SCRUM was designed for optimize the flexibility, creativity and productivity of well-trained teams.

SCRUM can be structured in three Roles, seven Activities (with 24 tasks) and five Artifacts. The three roles (reported in an IDEF0 diagram [30] as Mechanisms) are: 1) "Product Owner", 2) "Scrum Master", and 3) "Development Team". The "Product Owner" is responsible of the product backlog (its content, availability and ordering). The "Scrum Master" can be considered the SCRUM expert and project leader that will interact with the other roles for leading and guiding them toward the end goal. The "Development Team" "consists of professionals who do the work of delivering a potentially releasable Increment of DONE product at the end of each Sprint" [21].

The seven SCRUM Activities are: 1) "Planning Pre-Game", 2) "Systems Architecture Pre-Game", 3) "Sprint Planning Game", 4) "Daily SCRUM Game", 5) "Sprint Increment Development Game", 6) "Sprint Review Game", and 7) "Sprint Retrospective Game". Some literature [31, 26] adds an explicit final activity of 8) "Project Closure Post-Game". In this research, we have focused on the Project Management (PM) activities (i.e. the Activities 1, 3, 4, 6, 7 and 8). The activities 2 and 5 corresponds to Software Implementation (SwI) process.

The five SCRUM Artifacts are: 1) "User Need List", 2) "Product Backlog" (it includes the "User Stories"), 3) "Sprint Backlog" (it includes the "Sprint Burndown Chart"), 4) "Increment", and 5) "Acceptance Criteria". The specific Artifacts related to Project Management Process are 1, 2, 3, and 5. The 1) "User Need List" is the open list of needs and requirements expressed by the Customer. The 2) "Product Backlog" is "*an ordered list of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product*" [21]. An "User Story" is "*a card that describes an increment of value to the customer. The user story is written for the developer in order to express the increment of value*" [26]. The 3) "Sprint Backlog" is "*the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal*" [21]. This Artifact includes the "Sprint Burndown Chart" which is a chart which "*shows the amount of work remaining across time*" and permits to visualize "*the correlation between the amount of work remaining at any point in time and the progress of the project team(s) in reducing this work*" [32]. The 4) "Increment" is "the sum of all the Product Backlog items completed during a Sprint and the value of the increments of

all previous Sprints" [21]. The "Acceptance Criteria" is "*essentially a clarification of the story. It gives the developer a set of steps that must be completed before the story can be considered done. The acceptance criteria are created by the product owner with the help of the customer. It sets the expectation of the user story*" [26].

3.2 The ISO/IEC 29110 Standard – Entry Profile

The ISO/IEC 29110 standard (Entry Profile) [9] provides a lightweight process model developed for organizations classified as very small entities (VSEs employs from 1 to 25 people). According to [31] standard emerged for the needs identified in VSEs on: 1) clear and detailed guidance with templates and examples; 2) a lightweight and easy-to-understand standards; and 3) standards with minimum cost, time, and resources for their implementation. This ISO/IEC 29110 standard has three Roles, two Process Categories, and fourteen Artifacts. The three roles are: "Customer", "Project Manager", and "Work Team". The two Process Categories are: Project Management (PM) and Software Implementation (SI).

Project Management aims to establish and carry out the tasks of the software implementation, which will fulfill the objectives of the project according to quality, time and expected costs. PM includes four activities: ***Planning, Control, Execution and Closure***. **Software Implementation** aims to systematically analyze, design, construction, integration and testing of software products processed according to specified requirements. SI includes six activities: ***Initiation, Analysis, Design, Construction, Tests and Delivery.***

The fourteen Artifacts are: 1) Acceptance Record, 2) Change Request, 3) Meeting Record, 4) Progress Status Record, 5) Project Plan, 6) Project Repository, 7) Requirements Specifications, 8) Software, 9) Software Component, 10) Software Configuration, 11) Software Component Identification, 12) Statement of the Work, 13) Test Cases and Test Procedures, and 14) Test Report. The Activities and Artifacts of interest for this research are the corresponding to PM Process Category. These activities are: ***Planning, Control, Execution and Closure***; and these Artifacts are: 1) Acceptance Record, 2) Change Request, 3) Meeting Record, 4) Progress Status Record, 5) Project Plan, 6) Project Repository, 10) Software Configuration, and 12) Statement of the Work.

3.3 The Agility-Discipline Debate

According to several relevant literatures [16, 17], the direct application of ASDMs does not guarantee the proffered benefits of agility Project Management. Furthermore, from a disciplined Project Management approach [18, 19] there had been logical arguments on the need to robust the agile methods with some disciplined-oriented best practices. A summary of recommendations for having a balanced agility-disciplined Project Management approach (called also *ambidextrous* approach [34]) is as follows: 1) risks are not managed explicitly in agile methods; 2) a particular organizational culture is required for agile methods while that disciplined is less contingent to this factor; 3) agile methods can be considered chaotic by excessive

flexibility and customization for teams trained in disciplined methods; 4) control and monitoring of project must be still exercised; 5) agile methods are more focused on small teams and small project (that can be large by evolution but not for an initial planned scope as a large project) and thus their scale up suffer of drawbacks; 6) new current software projects are more complex than past ones so both approaches (disciplined and agile one) are required. Thus, a call for elaborating balanced agility-disciplined Project Management methodologies is currently reported in the literature.

4. A Means-Ends Design of SCRUM+

4.1 The SCRUM Initial Status as the Core Input for the Means-Ends Analysis

To design SCRUM+ as a balanced agility-disciplined enhanced SCRUM methodology, we start from the results reported in [28] regarding a thoroughly analysis of the coverage of SCRUM, XP and UPEDU Project Management processes regarding the ISO/IEC 29110 (Entry Profile, Project Management process). According to [35] SCRUM, XP and UPEDU had respectively an overall coverage of moderate (79%), low (51%) and high level (93%) respectively. Hence, while the obvious selection of reporting UPEDU as a ready-to-use balanced agility-disciplined methodology and highly in congruence with the ISO/IEC 29110 Entry Profile concerning to the Project Management process, UPEDU is not a well-known agile methodology in industrial settings and it supports an agile approach based on a simplified rigor and discipline from its derivation from RUP (i.e. a strong disciplined development methodology). In the opposite case, to try to enhance XP which has a low compliance level implies the addition of many missed issues, and thus, the enhanced XP can be perceived theoretically far away of the agile approach by practitioners. Thus, in this research it has been selected SCRUM that reached a moderate level (79%) and its enhancement toward next level (i.e. high) can produce a less conceptual disruption perception than the change required in XP from low to high coverage level (i.e. a suitable balanced agility-disciplined methodology).

4.2 Application of the Means-Ends Analysis Technique for Designing SCRUM+.

We propose six heuristic strategies to perform systematically transformations from SCRUM to SCRUM+. These six heuristic strategies are considering the design limits reported. The strategies 1 to 3 tried to eliminate only the items (i.e. a Role, an Activity, or an Artifact) that are evaluated in overall as NULL level, and the strategies 4 to 6 tried to eliminate both the overall of NULL and LOW levels regarding their compliance level with the ISO/IEC 29110 Entry Profile. These items (i.e. a Role, an Activity, or an Artifact) are not mentioned or are weakly reported in SCRUM regarding the ISO/IEC 29110 Entry Profile. The items were analyzed on the three components (Roles, Activities and Artifacts).

For example, the Strategy 1 was based on doing soft (minimal) changes from SCRUM to SCRUM+ by moving the NULL (✖+) and LOW (⊙) status found in individual feature (i.e. cells) of each item (i.e. a specific Role, Activity or Artifact) whose status level is NULL (✖+). Thus, NULL (✖+) to LOW (⊙) and from LOW (⊙) to MODERATE (◒) are the changes to be applied. The other two status level of

MODERATE (◒) and HIGH (●) found in the cells were kept in the same status level. These changes were soft (minimal), and it pursued a soft evolution from the SCRUM to SCRUM+ with the minimum change as possible for every specific item in the three components (Roles, Activities and Artifacts). The Strategies 4, 5 and 6 were almost the same of 1, 2 and 3 ones respectively, with the unique difference that the changes done in strategies 1, 2 and 3 were applied only on the items (i.e. a specific Role, Activity or Artifact) whose status level is NULL (✖+), while that in the strategies 4, 5 and 6 the changes are applied on items whose status level is NULL (✖+) and LOW (⊙).

All these qualitative assessments were finally mapped to a numerical scale from 0 to 3, and their average value multiplied by their correspond weight assigned to the specific Role, Activity-Task or Artifact. Thus, the final scores can be from 0 to 100 points. The Table 1 reports the final levels reached by each strategy. The value of 100 points for the strategy 6, for instance, implies to add to the original SCRUM all identified missing attributes for Roles, Activities-Tasks and Artifacts from the ISO/IEC 29110 standard, but it naturally will produce a loss of the agile essence of SCRUM. Thus, the selected strategy to produce SCRUM+ faces a trade-off situation between getting an improved methodology and keeping its agility status.

Table 1. Results of the Six Means-Ends Strategies

	Strategy 1	**Strategy 2**	**Strategy 3**	**Strategy 4**	**Strategy 5**	**Strategy 6**
Roles	78	89	100	78	89	100
Activities	93	97	100	93	97	100
Artifacts	73	80	80	83	90	100
Total	81	89	93	85	92	100

3.3 Solution: Selected Means-Ends Strategy

In order to select the final solution we defined three criteria: 1) the solution must have an overall level that it is between 80 and 89 points (i.e. a high coverage level); 2) the solution must have the minimal overall Euclidean Distance M3, which measures the distance to the origin (0,0) from a Solution Strategy mapped in a 2D plane of M1xM2, where the point (M1,M2) corresponds to M1 and M2 as the Euclidian Distance between the Solution Strategy, and the SCRUM solution and the ISO/IEC solution respectively; and 3) the Solution Strategy must have a Face Validity approbation though the visualization of the 3D-scatter graph (see Fig. 1). The Face Validity test means that the sphere of the solution in the 3D-scatter graph be perceived as suitable for being not so far to both SCRUM and ISO/IEC 29110, and thus very near to the theoretically IDEAL solution elaborated.

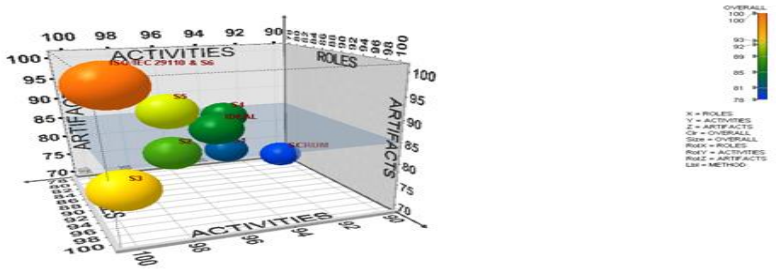

Fig. 1. Face Validity Test of Strategies with a 3D-scatter Graph.

By space limitations, we do not report the specific metrics calculated for the three criteria. However, in the Table 2 is reported the summarization of the results. Thus, we found the Strategy 2 as the unique solution that fitted the three criteria.

Table 2. Final Results from the Three Criteria

	Strategy 1	**Strategy 2**	**Strategy 3**	**Strategy 4**	**Strategy 5**	**Strategy 6**
Criterion 1	OK	OK		OK		
Criterion 2		OK			OK	
Criterion 3		OK			OK	

Again, by space limitations, we do not report the whole transformations executed on SCRUM on Roles, Activities-Tasks and Artifacts, but in the Table 3, we illustrate representative examples of enhancements taken from the ISO/IEC 29110 Entry Profile.

Table 3. Example of the differences between SCRUM y SCRUM+ Roles, Activities-Tasks and Artifacts

SCRUM	**SCRUM+**
	Roles (Product Owner)
Represents Customer's interests	**Represents Customer's interests** --- *Accomplish a mandatory formalized project start and closure.*
Authorize and review project outcomes	**Authorize and review project outcomes** --- Verify that the main interests of the customer needs be specified in the project start and closure in formalized way.
Accept or reject final product	**Accept or reject final product** --- Accomplish mandatory signed closure document.
	Activities-Tasks (Sprint Review)
Review of Increment	**Review of Increment** --- It is important to ensure the project closure with a document as a contract. In this case, it is very important to write the evidence of the present increment. --- It is very important to update the repository because it can be consulted in the future for auditing purposes. In this case, the last increment should be considered.
Review of Project Plan	**Review of Project Plan** --- It is important ensure the closure project with a document as a contract. In this case, is very important to write the evidence according with

	the projections to be done in the initial project plan. --- It is very important to update the repository because it can be consulted in the future for auditing purposes. In this case, the project plan should be considered.
Update Product Backlog	**Update Product Backlog** --- It is important to ensure the project closure with a document as a contract. In this case, is very important to write the evidence according with the projections to be done. --- It is very important to update the repository because it can be consulted in the future for auditing purposes. In this case, the product backlog should be considered.
Artifacts (Increment)	
Set of User Stories implemented (DONE) in the last Sprint.	**Set of User Stories implemented (DONE) in the last Sprint.** ---Determine how much was the cost of possible changes on the user stories done. Check against the initial plan.
Value provided to previous Increments from previous Sprints.	**Value provided to previous Increments from previous Sprints.** ---Determine the cost of possible changes on the previous increment from the previous sprint.
Review of potential adjustments to the Sprint Backlog	**Review of potential adjustments to the Sprint Backlog** ---Determine how much was the cost of possible changes on the Sprint Backlog done. Check against the initial plan.

5. Conclusions

This research was pursued with the objective to design systematically an enhanced balanced agility-disciplined SCRUM methodology from the original SCRUM and the ISO/IEC 29110 Entry Profile. Motivation is based on the recurrent literature on the need to strengthen the agile methods (in particular, the Project Management processes) with a balanced approach. A Mean-Ends Analysis technique was used for systematically produce six solutions. Three criteria were fixed for selecting the best one from these six solutions. A final solution SCRUM+ with an overall coverage of 89% was reached as a Means-Ends transformation from the original SCRUM with a 79% level of coverage.

We consider that as any conceptual research, limitations on the reproducibility and internal and external validity of these results can be reported. On reproducibility of results while we use the most original source materials the variability of expertise and self-interpretations of the designers can introduce variations. On internal validity, we consider that the utilization of different source material can introduce variations. On external validity, we conducted an initial conceptual validation from a panel of experts (which is not reported here by space limitations) with suitable initial results but a more robust empirical validation with SCRUM practitioners is planned for next step of this research. As main recommendations, we can report: 1) to conduct the empirical validation with SCRUM practitioners through a survey study; 2) to conduct experiments to compare the utilization of SCRUM vs SCRUM+; 3) to elaborate an Electronic Process Guide of SCRUM+ and promote their utilization in a VSE to study empirically via a Case Study their impacts and limitations.

Finally, we can conclude that balanced agility-disciplined Project Management methodologies are required and that the Means-Ends Analysis technique provides a robust method for systematically produce potential solutions such as SCRUM+ which was transformed from the original SCRUM.

References

1. SEI. 2010. CMMI for Development, Version 1.3. Technical Report CMU/SEI-2010-TR-033. Software Engineering Institute (SEI): Pittsburgh, PA.
2. ISO/IEC. ISO/IEC 12207:2008, Systems and Software Engineering – Software Life Cycle Processes, 2008, ISO organization, Geneva, Switzerland.
3. Succi, G., Valerio, A., Vernazza, T., Succi, G.: Compatibility, standards, and software production, vol. 6, no. 4, pp. 140-146. Standard View (1998)
4. Laporte, C., Alexandre, S., O'Connor, R.: A Software Engineering Lifecycle Standard for very Small Enterprises, pp. 129--141. Springer-Verlag Berlin Heidelberg (2008)
5. SEI: CMMI for development, version 1.2: CMMI-DEV, V1.2. Software engineering institute. CMU/SEI-2006-TR-008, ESC-TR-2006-008. Retrieved from www.sei.cmu.edu (2006)
6. Clarke, P., O'Connor, R.: The situational factors that affect the software development process: Towards a comprehensive reference framework, vol. 54, no. 5, pp. 433--447. Information and Software Technology (2012)
7. Unterkalmsteiner, M., Gorschek, T., Islam, A. M., Cheng, C. K., Permadi, R. B., Feldt, R.: Evaluation and measurement of software process improvement—a systematic literature review. IEEE Transactions on Software Engineering, vol. 38, no. 2, pp. 398--424. IEEE (2012)
8. Organization for Economic Co-Operation and Development (OECD).: SME and Entrepreneurship. (Outlook ed.) (2005).
9. O'Connor, R., Laporte, C.: Towards the Provision of Assistance for Very Small Entities in Deploying Software Lifecycle Standards. In: Proceedings of the 11th International Conference on Product Focused Software Development and Process Improvement (2010).
10. O'Connor, R. V., Laporte, C. Y.: Software Project Management with ISO/IEC 29110. Proceedings of the 19th European Conference on Systems, Software and Services Process Improvement (EuroSPI 2012), vol. 301. Springer-Verlag (2012)
11. ISO.: Software Engineering-Lifecycle profiles for Very Small Entities (VSEs)-. ISO/IEC TR 29110-5-1-1. ISO, Geneva, Switzerland (2012)
12. Dybå, T., Dingsøyr, T.: Empirical studies of agile software development: A systematic review. Information and software technology, vol. 50, no. 9, pp. 833--859 (2008)
13. West, D., Grant, T., Gerush, M., D'silva, D.: Agile development: Mainstream adoption has changed agility, vol. 2, no. 1, pp. 41. Forrester Research (2010).
14. Robillard, P. N., Kruchten, P., d'Astous, P. Yoopeedoo (UPEDU).: A process for teaching software process. In: Proceedings of the 14th Conference on Software Engineering Education and Training, pp. 18--26. IEEE (2001)
15. Sidky, A., Arthur, J., Bohner, S.: A disciplined approach to adopting agile practices: the agile adoption framework. Innovations in systems and software engineering, vol. 3, no. 3, pp. 203--216 (2007)
16. Misra, S. C., Kumar, V., Kumar, U.: Identifying some important success factors in adopting agile software development practices. In: Journal of Systems and Software, vol. 82, no. 11, pp.1869-1890 (2009)
17. Chow, T., Cao, D. B.: A survey study of critical success factors in agile software projects. In: Journal of systems and software, vol. 81, no. 6, pp. 961-971 (2008)
18. Boehm, B.: Get ready for agile methods, with care. In: Computer, vol. 35, no. 1, pp. 64--69 (2002)
19. Boehm, B., Turner, R.: Using risk to balance agile and plan-driven methods. In: Computer, vol. 36, no. 6, pp. 57--66 (2003)
20. Schwaber, K.: Scrum development process. In: Business Object Design and Implementation, pp. 117--134. Springer London (1997)

21. Sutherland, J., Schwaber, K.: The scrum guide. The definitive guide to scrum: The rules of the game. Scrum.org (2013)
22. ISO Software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 5-1-1: Management and engineering guide: Generic profile group: Entry profile. ISO/IEC TR29110-5-1-1. ISO, Geneva, Switzerland (2012).
23. Anderson, J. R.: Problem solving and learning. American Psychologist, vol. 48, no. 1, pp. 35--44 (1993)
24. Newell, A., Shaw, J. C., Simon, H. A.. Chess-playing programs and the problem of complexity. In: IBM Journal of Research and Development, vol. 2, no. 4, pp. 320-335 (1958)
25. Newell, A., Simon, H. A.: Human problem solving, vol. 104, no. 9. Englewood Cliffs, NJ: Prentice-Hall (1972)
26. Blankenship, J., Bussa, M., Millett, S., Lewis, R., Foggon, D.: Pro Agile. NET Development with Scrum. Apress (2011)
27. Eclipse Process Framework Project, http://epf.eclipse.org/wikis/scrum/
28. Galvan-Cruz, S., Mora, M., O'Connor, R. V., Acosta, F., Álvarez, F.: An Objective Compliance Analysis of Project Management Process in Main Agile Methodologies with the ISO/IEC 29110 Entry Profile. In: International Journal of Information Technologies and Systems Approach (IJITSA), vol. 10, no. 11, pp. 75--106 (2017)
29. Highsmith, J. A.: Agile software development ecosystems, vol. 13. Addison-Wesley Professional (2002)
30. IDEF: IDEF Method Report - The Standard for Integration Definition for Function Modeling (IDEF0). Federal Information Processing Standards Publication 183. Retrieved from http://www.idef.com/pdf/idef0.pdf (1993).
31. Galal-Edeen, G.H., Riad, A.M., Seyam, M.S.: Agility Versus Discipline: Is Reconciliation Possible?, pp. 331--337 (2007)
32. Schwaber, K: What is Scrum?, Scrum: Its about common sense. Internet document (2007)
33. Laporte, C. Y, Fanmuy, G., Ptack, K.: The Development of Systems Engineering International Standards and Support Tools for Very Small Enterprises. In: 22nd Annual International Symposium of the International Council on Systems Engineering, pp. 9--12 (2012)
34. Vinekar, V., Slinkman, C. W., Nerur, S.: Can agile and traditional systems development approaches coexist? An ambidextrous view. Information systems management, vol. 23, no. 3, pp. 31--42 (2006)
35. Public Site of the ISO Working Group Mandated to Develop ISO/IEC 29110 Standards and Guides for Very Small Entities involved in the Development or Maintenance of Systems and/or Software, http://profs.etsmtl.ca/claporte/english/VSE/ (2016)