Investigating Vincenti Engineering Principles in Support to the Auditing of Measurement Processes in Agile Organizations

Malik Qasaimeh[1*], Alain Abran[2], Ammar Abdallah[2], Raad S. Al-Qassas[1]

[1] King Hussein Faculty of Computing Sciences, Princess Sumaya University for Technology, Al-Jubaiha, Amman 11941, P.O. Box 1438, Jordan.
[2] École de Technologie Supérieure, University of Québec, 1100 Notre–Dame Ouest, Montréal, Québec H3C 1K3, Canada.

* Corresponding author. Email: m.qasaimeh@psut.edu.jo

Abstract: Software organizations that develop their software products using agile software processes, such as Extreme Programming (XP), face a number of challenges in their effort to demonstrate that their process activities conform to ISO 9001 requirements, including product measurement requirements. This paper proposes an auditing model for ISO 9001 measurements requirements that is applicable in agile XP environments. The design of this auditing model is based on some of the Vincenti's principles on engineering design. The proposed auditing model includes the use of several auditing "yardsticks" derived from ISO 15939, the SWEBOK Guide and the CMMI-DEV guidelines for requirements measurement. This auditing model has been assessed based on case studies selected from the literature.

Key words: ISO 9001, agile software development, vincenti engineering principles, certification process, auditing.

---

1. Introduction

ISO 9001 impacts the entire range of software life cycle activities, including software planning, software requirements gathering and analysis, software construction, the software life cycle traceability process, and the measurement process [1]. To address all these activities, software organizations that need to become ISO 9001 certified find themselves in a position where they need to develop a myriad of tools and techniques to demonstrate that their software processes are in conformity with this ISO quality standard. A common methodology is to have in place a certification team (i.e. software analysts) responsible for understanding which ISO 9001 clauses impact the organization's business processes, including software process activities. This team must also assess the development team process to demonstrate that the software products are being developed according to ISO 9001 requirements: in essence, this means providing documented evidence that clarifies how and when a particular design decision has been implemented. The collection of evidences constitutes a very important foundation on which the auditors base their audit results and conclusion.

The measurement activities in agile software processes create challenges for both auditors and software

developers. These challenges are inherited from the structure of iterative development, including the ability of handling requirements changes and a decreased emphasis on documentation in agile software processes. For example, the literature proposes techniques for agile software processes such as XP and Scrum, including the term "Velocity" to represent the effort invested to produce software [2]. Calculation of "Velocity" is based on data from several previous iterations, then averaging the estimates of the features that will be delivered per iteration. Project managers can use several measurement units, such as User Story Points (USP), ideal days, or hours of work. This calculated project velocity is used for estimating new sets of backlog User Stories. These historical data are also a valuable resource for ISO 9001 auditors. However, it has been noted in [3] that not all agile teams collect or have access to historical data, and that they often rely on an estimate of their upcoming velocity to perform their preliminary project estimation. This practice will provide fewer evidences for ISO 9001 to assure that the estimation has been performed based on an engineering approach. Moreover, the literature reports on fewer measurements approaches specifically defined for agile software processes [4].

This paper analyzes the ISO 9001 requirements of measurement and introduces two types of relations (i.e. implicit relations and explicit relations) to highlight the impact of ISO 9001 measurement requirements on the development activities of a software system. This paper proposes a set of auditing criteria and auditing yardsticks that can help, for example, the agile-XP development teams to comply with ISO 9001 measurement requirements. The proposed auditing criteria and auditing yardsticks are also intended to help the auditors to assure that the XP activities have been implemented in conformance with ISO 9001 measurement requirements. Four approaches for agile-XP measurement approaches are audited and evaluated based on the proposed audit model.

This paper is organized as follows. Section 2 presents an analysis of the ISO 9001 measurement requirements. Section 3 presents the methodology for the design of the proposed auditing model. Section 4 presents the structure of the proposed auditing model, including criteria and yardsticks. Section 5 presents an assessment of the proposed models using four case studies from the literature. Finally, section 6 presents a summary and some future work.

2. ISO 9001 Measurement Requirements

The requirements for measurement in ISO 9001:2008 cover multiple activities of the quality management system, such as customer satisfaction, process quality attributes and product quality attributes. These requirements are presented in four different subsections: customer satisfaction in ISO 8.2.1 section, internal auditing in ISO 8.2.2 section, measurement of processes in ISO 8.2.3 section and measurement of products in ISO 8.2.4 section.

The ISO 9001 recognizes customer satisfaction as one of the important measurement factors for the performance of a quality management system. Thus software organizations are required to provide methods and techniques for the gathering and analysis of the customer feedback data. Similarly Customer Satisfaction data is essential to project managers for continuous improvement, corrective actions, and management reviews. ISO 9001 also emphasizes the importance of monitoring and measurement techniques at the process level during the development of a product, but does not clearly defines the differences between the activities of process measurement and process monitoring in section 8.2.2. The SWEBOK Guide [7] defines the term "process measurement" as the collection, analysis and interpretation of quantitative information about the process. Measurement at this level is performed to identify the strengths and weaknesses of processes and to evaluate processes after they have been implemented and/or changed. On the other hand, "process monitoring" is meant to cover all activities that steer the implementation of projects by continuous assessment of their possibility to achieve expected goals.

ISO 9001 section 8.2.4 requires the organization to monitor and measure the product characteristics. These characteristic should be measured to verify that the product requirements have been met. The term "verify" has been used clearly in the requirement sentence in this section: it should not be confused with the meaning of other terms such as "verification" used frequently in software engineering to determine specific activities that are performed at the process level of software development.

The role of internal auditing is defined in section 8.2.2 in the ISO 9001. The standard characterizes the internal auditing process in terms of organization resources, scope and planning activities. The set of requirements in ISO 9001 section 8.2.2 specifies that the internal auditing should be performed by the organization itself at planned intervals to determine whether the organization processes and activities are in conformity to the ISO standard.

3. Design of the Auditing Model

The design process for the auditing model is clarified here using the concepts of engineering design process in Vincenti [5], CMMI [6], SWEBOK [7] and ISO 15939 [8]. While in Vincenti the principles of engineering design have been inspired from the domain of aeronautical engineering, Vincenti also mentioned that this classification was not specific to the aeronautical engineering domain and that a transformation could be made for design and analysis purposes to any other engineering domain. For example, the work presented in Meridji [9] utilizes Vincenti's engineering domain areas as constituting criteria for investigating software engineering from an engineering perspective. Also, Zarour [10] uses these Vincenti's engineering principles for the development of evaluation methods to evaluate the software assessment methods from an engineering perspective.
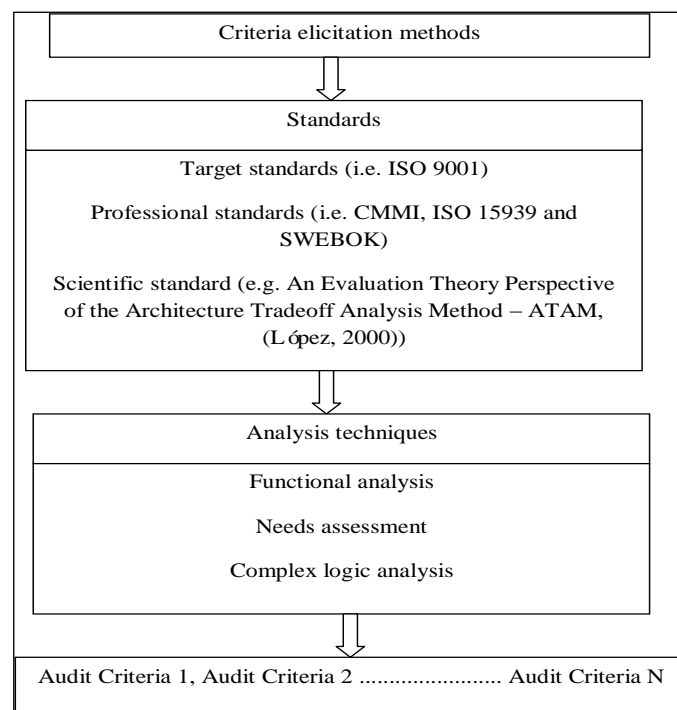

Fig. 1. Auditing criteria elicitation [12].

The main components for the proposed model are the auditing criteria of the ISO 9001 requirements. These auditing criteria are defined as the major area of interest for the software process auditor: there are expected to reveal audit evidences to assure that certain activities have been planned, executed and/or evaluated. Based on Lopez [11], criteria elicitation can be made using a mandatory standard that contains

implicitly the criteria to be applied. For the identification of measurement auditing criteria, the mandatory standard is ISO 9001, and more precisely the measurement requirements of ISO 9001. However, deriving auditing criteria from measurement requirements in Sections 8.2.1, 8.2.3, and 8.2.4 of ISO 9001 is difficult without a support from other related software engineering-based models, such as CMMI, SWEBOK and ISO 15939. For this purpose, Lopez [11] mentions that several criteria elicitation techniques can be used such as functional analysis, needs assessment and complex logical analysis.

Fig. 1 shows the main source documents for criteria elicitation and the techniques that define the auditing criteria for ISO 9001 measurement requirements. For example, ISO 15939 details the following phases during the measurement process: establishing and sustaining measurement commitment, plan the measurement process, performing the measurement process and measurement evaluation.

This standard is used here for deriving the main measurement auditing criteria, and to define the process and product measurement information that are useful for the software organization that implements an agile process (i.e. XP), in order to achieve the ISO 9001 certification. Those auditing criteria will be useful to an auditor to extract the audit evidences that assure the process conformance. Agile software organizations can also implement their measurement process to address specific technical and/or project measurement techniques related to the auditing criteria.

4. Model Structure

This section provides a description of the structure of the proposed auditing model, including its criteria and their related yardsticks.

4.1. Auditing Criterion for Measurement Plans

In the context of a measurement process, the SWEBOK Guide and ISO 15939 standards stress the importance of planning at the beginning of the measurement process. The same is present in the CMMI model where the planning for measurement is classified as a support process at maturity level 2 to improve the corrective and preventive actions. These standards highlight the importance of early planning for a software process, including for specifying the objectives of the measurement program and aligning the identified information needs within the organizational objectives, i.e. improving quality, reduce cost, enhance the infrastructure security etc.

An auditing criterion at this phase focuses on investigation of evidences to assure that activities of planning have been carried during the measurement process. The key point of this auditing criterion is to provide a link between the development team of agile-XP software processes and the auditors to understand what evidences they need at this phase, and how evidences can provide them with facts.

The following are the audit yardsticks identified for this criterion:

Yardstick #1:

Identification of the measurement context: the measurement plan should identify the context of the measurement process. This can be accomplished by defining the focus of the measurement process to the intended goal of the organizational improvement. For example, the SWEBOK mentions several improvement goals at the organizational level, such as the organizational processes, application domains, technology and organizational interfaces. The SEWBOK mentions that "it is important to make this context explicit and to articulate the assumptions that it embodies and the constraints that it imposes".

Yardstick #2:

Role Assignment: this yardstick is focused on identifying the measurement team roles and the responsibilities of each team member. This yardstick has been derived from ISO 15939 that mentions "Individuals shall be assigned responsibility for the measurement process within the organization". On the other hand, the CMMI model mentions that when the organization establishes a measurement program, the

measurement plan should include a specific role for each team member, as to ensure that analysis will properly address the identified information needs.

Yardstick #3:

Resources and budget constraints: the plan for the measurement process should be integrated with the main business improvement plan. For example, the CMMI model suggests several activities in the business improvement plan, such as:

- Identify major milestones: milestones should be reviewed once the measurement plan is defined as to keep the measurement process aligned with the project scope and time constraints.

- Identify schedule assumptions: assumptions are judgments and forecasting of several development activities' duration and required resources. Without documented assumptions, little data is available for estimations based on engineering methodologies. CMMI mentions that: "When schedules are initially developed, it is common to make assumptions about the duration of certain activities".

- Identify constraints: the CMMI defines the constraints as factors that limit the flexibility of development and design. For agile-XP, this can include team expertise, customer involvement, project size, project criticality (e.g. safety critical systems), requirement changes, etc.

- Identify task dependencies: the dependencies between project tasks and/or services should be defined. CMMI mentions that an ordered sequence of task dependencies for the software project can minimize the project duration. For agile-XP, several dependencies can be defined at this level, such as: logical dependency, data dependency, temporal dependency and resource dependency [13].

4.2. Auditing Criterion for Measurement Development

The focus of this auditing criterion is to help the auditors of agile-XP in finding evidences of the development activities for the measurement process. The auditing yardsticks associated with this auditing criterion are presented next.

Yardstick #4:

Identification of measures: software measures are classified into base measures and derived measures which are used to construct the software quality measure [14]. This auditing yardstick is focused on investigating the classification of measures that have been specified for the agile-XP project. As explained by the in ISO 15939, data for the base measures are obtained by direct measurement, whereas data for derived measures come from other measures, typically by combining two or more base measures. In other words, the base measures are obtained based on the attribute itself and without using any other measures, while derived measures are calculated based only on other measures.

Yardstick #5:

Measurement data and models: the CMMI mentions that "Existing sources of data may have been identified when specifying the measures. Appropriate collection mechanisms may exist whether or not pertinent data have already been collected". This yardstick investigates the existence of mechanisms for the data collection used during the measurement development. This can provide the auditors for agile-XP with evidences that a specific mechanism/method has been developed to collect the measurement data. Several measurement data collection methods can be inspected in this yardstick such as: qualitative data collection methods, web based questionnaires and paper based questionnaires. Modeling artifacts of measurement data is also a valuable resource of evidence, especially for the measurement design based on modeling measurement methods such as COSMIC – ISO 19761 [15].

Yardstick #6:

Measurement results: the focus of this auditing yardstick is the investigation of the existence of appropriate data analysis and presentation techniques for better understanding of measurement results.

The SEWBOK Guide mentions that measurement users can also participate in reviewing the data to ensure its accuracy.

Yardstick #7:

Traceability of measurement data: this yardstick focuses on the investigation of the existence of techniques for tracing the measurement data throughout the measurement process. The importance of measurement traceability techniques has been highlighted by the CMMI model as a tool for identification of the measures that have been already addressed in the measurement plan.

Yardstick #8:

Prioritization of measurement data: the focus of this auditing yardstick is the investigation of the existence of techniques for prioritizing the measurement data throughout the measurement process. CMMI mentions that it is important to develop a prioritization technique throughout the measurement development to improve the accuracy of measurement data and to enhance the measurement evaluation.

Yardstick #9:

Identification of a measurement design: this yardstick focuses on investigating the classification of measurement design that has been used by the measurement team. Vincenti [5] has classified the engineering design into:

- Normal design: the main features that characterize the normal design is "evolutionary rather than revolutionary" and "The designer knows at the outset of how the device works and what its customary features are". The measurement design will be considered to be a normal design if it is based on known measurement models such as COSMIC.
- Radical design: the main features which characterize the radical design are "The designer has never seen such a device before and cannot presume that it will succeed" and "the designer is not familiar with the device itself". The measurement design will be considered to be a radical design if it does not follow the guidelines of any measurement model and the measurement methods are designed based on team experiences.

4.3. Auditing Criterion for Measurement Management

The measurement management auditing criterion focus is to identify the auditing yardsticks that can reveal evidences at the measurement management level. The measurement management auditing criterion is composed of the following yardsticks.

Yardstick #10:

Evaluation of the measurement process and results: The measurement result is the output of the analysis and manipulation that has been made on measurement data to achieve results that are useful for project stakeholders. This yardstick is also important to investigate evaluation methodologies (i.e. criteria) for the measurement process and results. The SWEBOK Guide and the CMMI model have not specified any evaluation criteria. The recommendations for evaluation criteria have been found in ISO 15393, such as: timelines, efficiency and defect containment. For auditors of agile-XP, it is important to find evidences that the measurement process and results have been evaluated. Existence of such evaluation criteria can provide more evidences for important measurement activities, such as: customer involvement, team expertise and/or tools selection process.

Yardstick #11:

Result communication: this auditing yardstick is focused on investigating the existence of an appropriate scheduling and presentation procedures to discuss the measurement results with the project stakeholders. This will include any procedure that assists the project stakeholders to understand the measurement results and keep the project stakeholders informed about the measurement results periodically.

5. Assessment of the Auditing Model

5.1. Case Studies Selection

To study the applicability of the auditing model proposed in this paper, four case studies have been selected as compatible with the scope defined in this paper - see Table 1. The selection of these case studies was based on a search in IEEE Xplore and ScienceDirect-Elsevier for any paper proposing a methodology, technique, or framework to enhance the measurements of an agile software process. The following terms were used in browsing the content of IEEE Xplore and Science Direct-Elsevier:

- Agile AND measurement program
- XP AND measurement program
- Agile AND measurement techniques OR method.
- XP AND measurement techniques OR method.

The title and abstract analysis was performed to select the papers that discuss agile measurement. Note that some authors discuss different agile processes such as XP and Scum, but refer to them as agile process (i.e., they do not specify which agile process they are improving, and selected XP as a candidate process without referring to it by name).

Table 1. Selected Agile Measurement Case Studies

	Authors	Case title	Case objective	Case Date
Case A	Ktata, Lévesque	Designing and implementing a measurement program for Scrum teams: what do agile developers really need and want?	To propose a measurement program for agile software process in software organization. The goal of this project is to allow teams and individuals to improve their development process and to provide better product quality and control over the project.	May 2010
Case B	Mahnic, Zabkar	Introducing CMMI Measurement and Practices into Scrum-based Software Development Process	To develop a measurement program for agile software process based on CMMI measurement and analysis key process area (KPA).	June, 2007
Case C	Gustafsson	Model of Agile Software Measurement: A Case Study	To propose a measurement model for performance measurement and process optimization measurement in agile software process environment.	June, 2011
Case D	Fehlmann	Six Sigma for Agile Teams	To propose a measurement technique for agile software processes based on the methods used in Six Sigma and COSMIC for Software Functional Size Measurement (FSM).	March, 2011

All the papers that discuss agile measurement in general, without proposing a methodology, technique, or framework for managing the measurement process, were discarded. For the four cases described in Table 1, the evidences were gathered using the information system audit procedure described by the standards, guidelines and procedures for information system auditing ISACA [16], as follows:

- Observation of measurement processes and the existence of the components of the measurement method.
- Documentary audit evidence, such as results of the measurement method execution, and records of the method evaluation.
- Representations of the measurement results, such as written analyses, and descriptions of the measurement method flowcharts.

5.2. Audit Summary

Case A [17] (Ktata, Lévesque): Case A supports the auditing criteria for measurement plans by identifying the context of the measurement program using the Goal Question Metrics approach (GQM). The GQM has been used to state explicit measurement goals and to analyze the measurement data. Role Assignment is not fully supported since there is no evidence on how the measurement team has been managed (i.e., Measurement user, Measurement analyst and/or Measurement librarian) but it has been noted that the product owner has assigned a responsible for integrating the measurement activities with regular project activities. No evidence has been identified on how the resources and budget for the measurement team has been managed.

Table 2. Illustrates the Auditing Results for the Case Studies Based on Proposed Auditing Model

	Case A (Ktata, Lévesque)	Case B (Mahnic, Zabkar)	Case C (Gustafsson)	Case D (Fehlmann)
Auditing criteria for measurement plan				
Yardstick # 1 Identification of the measurement context	Evidence exists	Evidence exists	Evidence exists	Evidence does not exist
Yardstick # 2 Role assignment	Evidence partially exists	Evidence exists	Evidence partially exists	Evidence does not exist
Yardstick #3 Resources and budget constraints	Evidence does not exist	Evidence exists	Evidence does not exist	Evidence does not exist
Auditing criteria for measurement development				
Yardstick #4 Identification of measures	Evidence exists	Evidence exists	Evidence exists	Evidence exists
Yardstick #5 Measurement data and models	Evidence exists	Evidence exists	Evidence exists	Evidence exists
Yardstick #6 Measurement results	Evidence does not exist	Evidence partially Exists	Evidence exists	Evidence exists
Yardstick #7 Traceability of measurement data	Evidence does not exist	Evidence exists	Evidence does not exist	Evidence exists
Yardstick #8 Prioritization of measurement data	Evidence exists	Evidence does not exist	Evidence does not exist	Evidence exists
Yardstick MR #9 Identification of measurement design	Evidence exists	Evidence exists	Evidence exists	Evidence exists
Auditing criteria for measurement management				
Yardstick #10 Evaluation of measurement process and results	Evidence does not exist	Evidence does not exist	Evidence exists	Evidence does not exist
Yardstick #11 Results communication	Evidence does not exist	Evidence exists	Evidence does not exist	Evidence exists

The auditing criteria for measurement development have been supported mainly by identifying a set of measures which has been derived from the GQM. The measures are next classified into team dynamic indicators, process and project related indicators, customer related improvement indicators and internal quality indicators. No evidence has been found on traceability of the measurement data.

Case B [18] (Mahnic, Zabkar): Case B supports the auditing criteria for measurement plans by identifying the goal and context of the measurement program using the guidelines of the CMMI model. The measurement program context is set into: timely information on project performance, quality improvement,

job satisfaction, and efficient impediments resolution and customer satisfaction. The role assignment of the team was managed using relational tables describing the relationships among projects, teams and employees: each team has an ID and each employee has an ID and an employee description. Evidence has been found on how the measurement data was prioritized. No evidence has been found on how the measurement result was evaluated - See table 2.

Case C [19] (Gustafsson): Case C supports the auditing criteria for measurement plans by identifying the goal and context of the measurement program using a set of identified key process indicator areas (KPIA). The context of Case C is to improve the following organizational practices: product quality, product delivery, development cost, service level and product planning. The measurement user for Case C is Bwin Games, a web game software company based in Sweden. No evidence has been found on how the measurement responsibilities were assigned to the measurement team. No evidence has been found on how the resources and budget of the measurement cost were managed. No evidence has been found of traceability of the measurement data and no evidence has been found for management data prioritization. No evidence has been found on how the measurement results were evaluated or the presentation procedures to the project stakeholders. Evidence of evaluation of measurement results and communication has been found as a part of the design of the measurement model.

Case D [20] (Fehlmann): Case D presents measurement methods for "making agile development processes lean and measurable". The objective of Case D was not to propose a measurement program for agile software process but to focus on a measurement improvement method for the agile software process. The objectives of Case D make it less convenient to support auditing criteria for measurement plans. The auditing criteria for measurement development have been supported mainly by identifying the data movements based on the guidelines of COSMIC model. The goal of the identified data movements is to estimate effort needed for implementing user stories. The measurement data was modeled using data movement diagrams and prioritized using Quality Function Deployment (QFD). The measurement result was presented using the "Buglione-Trudel Matrix". No evidence has been found for traceability of measurement data. Table 2 summarizes the auditing results for the case studies based on proposed auditing model.

6. Summary and Future Work

This paper has proposed an auditing model for ISO measurement requirements. This model can help software organizations in their effort to achieve ISO 9001 certification and help software auditors to extract auditing evidences that demonstrate the ability of a software organization to implement the ISO 9001 measurement requirements. The design process for the auditing model extension is based on Vincenti's engineering design, evaluation theory, the CMMI model and the SWEBOK Guide. Four different case studies have been audited based on the proposed model to investigate whether or not they conform to the ISO 9001 measurement requirements. The evidences gathered show partial support for the requirements in each case study; however no case study has been demonstrated as supporting fully the auditing yardsticks.

The future work will focus on enhancing the assessment process of the proposed model. A limitation of the current assessment is that the case studies that have been selected to evaluate the proposed auditing model have been carried out by the same researchers who designed the auditing models: this might introduce biases in the evaluation process of the proposed auditing models. To cope with this problem, the validation of the auditing models could be performed by external experts to assess the usability of the proposed auditing models in the area of ISO 9001 auditing and agile software processes.

References

[1] Malik, Q., & Alain, A. (2015). Design and assessment for agile auditing model: The case of ISO 9001 traceability requirements. *Lecture Notes on Software Engineering*, *3(3)*, 189-194.

[2] Jeeva, P. K. V., & Dilum, B. P. (2015). Use of software metrics in agile software development process. *Proceedings of the Moratuwa Engineering Research Conference* (pp. 312-317).

[3] COSMIC (2011). Guideline for the use of COSMIC FSM to manage agile projects, V 1.0. Retrieved May 22, 2011, from http://www.cosmicon.com/portal/public/COSMIC_Agile_Projects_Guideline_v10.pdf.

[4] Popli, R., & Chauhan, N. (2014). Cost and effort estimation in agile software development. *Optimization, Reliabilty, and Information Technology*, 57-61.

[5] Vincenti W. G. (1990). *What Engineers Know and How They Know It*. The John Hopkins University Press.

[6] Capability Maturity Model Integration (CMMI), Version 2.1 - Institute – 2012.

[7] Abran, A., & J. Moore. (2004). *Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society Press, 1-228

[8] Software engineering — Software measurement process. (2012). *International Organization for Standardization*, 1-37.

[9] Meridji, K. (2010). Analysis of software engineering principles from an engineering perspective. Ph.D. dissertation, École de technologie supérieure, Montréal (Canada), 2010.

[10] Zarour, M. (2009). Methods to evaluate lightweight software process assessment methods based on evaluation theory and engineering design principles. PhD thesis, École de Technologie Supérieure, Université du Québec, Montréal, Canada.

[11] Lopez, M. (2000). An evaluation theory perspective of the architecture tradeoff analysis method (ATAM).

[12] Qasaimeh, M., & Abran, A. (2012). Extending extreme programming user stories to meet ISO 9001 formality requirements. *Journal of Software Engineering and Applications*, *4(11)*, 626-638.

[13] Qasaimeh. (2012). Auditing for ISO 9001 requirements in the contextof agile software processes. PhD thesis, École de Technologie Supérieure, Université du Québec,, Montréal, Canada.

[14] Al-Qutaish, R. (2007). SPQMM: A software product quality maturity model using ISO/IEEE standards, metrology, and sigma concepts. PhD thesis, École de Technologie Supérieure, Université du Québec, Montréal, Canada.

[15] ISO-19761. (2011). Software engineering — COSMIC v 3.0 — A functional size measurement method. *International Organization for Standardization*.

[16] ISACA. (2010). IT standards, guidelines, and tools and techniques for audit and assurance and control professionals. *Information Systems Audit and Control Association*.

[17] Ktata. O., & Lévesque, G. (2010). Designing and implementing a measurement program for Scrum teams: what do agile developers really need and want? *Proceedings of the Third C* Conference on Computer Science and Software Engineering* (pp. 101-107).

[18] Mahnic, V., & Zabkar, N. (2007). Introducing CMMI measurement and analysis practices into scrum-based software development process. *International Journal of Mathematics and Computers in Simulation. 1(1),* 65-72.

[19] Johan, G. (2011). Model of agile software measurement: A case study. Master Thesis, Chalmers University of Technology University of Gothenburg, Sweden.

[20] Fehlmann, T. H. (2011). Six sigma for agile teams. *Proceedings of the International Conference on the Modern Art of Software* (pp. 1-11).

Malik Qasaimeh is an assistant professor of software engineering at Princess Sumaya University for Technology (Jordan). He received his PhD degree in software engineering form University of Quebec (Canada, 2012). He has a master degree in information systems security from Concordia University in Canada, in 2007, and a bachelor degree in computer science from Jordan University of Science and Technology in Jordan, in 2003. He has published several papers in a reputed journals and international conferences. His research interests include agile software processes certification and compliance, software process and product improvement, software engineering ISO standards, software security and software engineering principles.

Alain Abran is a professor at Ecole de technologie superieure (ETS) — University of Quebec (Montréal, Canada). He holds a Ph.D. in electrical and computer engineering (1994) from École Polytechnique de Montréal (Canada) and the master's degrees in management sciences in 1974 and the electrical engineering in 1975 from the University of Ottawa (Canada). He has published over 400 peer-reviewed publications and he is the author of the books *Estimation of Software Projects*, *Software Metrics and Software Metrology* and a co-author of the book *Software Maintenance Management* (Wiley Inter science & IEEE-CS Press). Dr. Abran is co-editor of the 2004 Guide to the Software Engineering Body of Knowledge – SWEBOK, and the chairman of the Common Software Measurement International Consortium (COSMIC).

Ammar Abdallah is a research assistant in software engineering and IT at Quebec University in Montreal, Canada. He received his master degree in quality systems engineering from Concordia University in Montreal, Canada in 2014 and his undergrad in computer science from Al-Balqa Applied University - Jordan in 2011. His research interests include enterprise architecture, software engineering measurement and standards, and agile methodologies.

Raad S. Al-Qassas received his PhD in computer science from University of Glasgow/UK in 2007. Dr. Al-Qassas research interests include computer networks, mobile computing, wireless communication systems, security, software engineering and cloud computing. He has served on the technical program committees of many well-known international conferences and as a reviewer of well-known international academic journals. He has also served as supervisor and examiner for undergraduate and postgraduate students. Dr. Al-Qassas has been a member of academic staff at Princess Sumaya University for Technology since 2007.