Software Qual J (2006) 14: 7–23
DOI 10.1007/s11219-006-5997-8

PEM: The small company-dedicated software process quality evaluation method combining CMMI[SM] and ISO/IEC 14598

Sylvie Trudel · Jean-Marc Lavoie · Marie-Claude Paré · Witold Suryn

Abstract Many small software organizations have recognized the need to improve their software product. Evaluating the software product alone seems insufficient since it is known that its quality is largely dependant on the process that is used to create it. Thus, small organizations are asking for evaluation of their software processes and products. The ISO/IEC 14598-5 standard is already used as a methodology basis for evaluating software products. This article explores how it can be combined with the CMMI to produce a methodology that can be tailored for process evaluation in order to improve their software processes.

1. Introduction

Many small software organizations have recognized the need to improve their software product. Evaluating the software product alone seems insufficient since it is known that its quality is largely dependant on the process that is used to create it. Thus, small organizations are asking for evaluation of their software processes and products. The ISO/IEC 14598-5

[SM]: CMMI is a service mark of Carnegie-Mellon University.

S. Trudel
Software Test Center, CRIM, 550, rue Sherbrooke Ouest, Bureau 100, Montréal (Québec), Canada H3A 1B9
e-mail: strudel@crim.ca

J.-M. Lavoie
Consultations Jean-Marc Lavoie Inc., 5815 Parnasse, Brossard (Québec), Canada J4W 2C7
e-mail: jeanmarc.lavoie@videotron.ca

M.-C. Paré
Motorola Canada Software Center, 700 Wellington Street, Montréal (Québec), Canada H3C 3S4
e-mail: Marie-Claude.Pare@motorola.com

W. Suryn
Département de génie logiciel et TI, École de technologie supérieure, 1100, rue Notre-Dame Ouest, Montréal (Québec), Canada H3C 1K3
e-mail: wsuryn@ele.etsmtl.ca

(ISO/IEC 1459805, 1998) standard is already used as a methodology basis for evaluating software products, combined with quality measures that are described in ISO/IEC 9126 (ISO/IEC 9126-1, 2000). Can the same method be used to evaluate software processes against the CMMI (CMMI Team, 2001), leading to its qualitative measurement? This article describes the design activities to define a software process evaluation based on ISO/IEC 14598-5 and the CMMI. It also describes the resulting method and field trials.

1.1. ISO/IEC 14598-5

The ISO/IEC 14598 series is concerned with the process of evaluation, seen from different viewpoints. The standard is separated in six parts and provides guidance for 3 different perspectives: developer, acquirer and evaluator. As we are interested in an independent evaluation of a software process, part 5 of 14598 was used. The ISO/IEC 14598-5 is a standard that describes the evaluation process and the activities needed to perform an independent software evaluation in terms of quality characteristic as defined in ISO/IEC 9126. The evaluation process comprises the following five activities:

- The analysis of the evaluation requirements, where the evaluation objectives are described.
- The specification of the evaluation, where the scope of the evaluation and the measurements to be performed on the product are defined.
- The design of the evaluation, where the procedures to be performed during the evaluation are specified.
- The execution of the evaluation, where the defined procedures are executed and the results are obtained.
- The conclusion of the evaluation, where the final evaluation report is completed and delivered to the requester of the evaluation.

Figure 1 summarizes the ISO/IEC 14598-5.

1.2. The capability maturity model integration (CMMI)

The Capability Maturity Model Integration (CMMI^{SM}) is known in the industry as a best practices model. It combines practices of Systems Engineering (SE), Software Engineering (SW), Integrated Process and Product Development (IPPD), and Supplier Sourcing (SS) disciplines. The CMMI is mostly used to "provide guidance for an organization to improve its processes and ability to manage development, acquisition, and maintenance of products and services". The CMMI was conceived to allow organizations to rely on a single model to evaluate their maturity and process capability, establish priorities for improvements, and help them improve their practices.

The CMMI is available for various combinations of disciplines in two representations: "Staged" and "Continuous". The model is divided into Process Areas (PA), each of which containing a set of generic and specific practices. In the Staged representation, PA are grouped into maturity levels, as shown in Table 1. Version 1.1 of the CMMI was published in January 2001.

1.3. Motivation for combining ISO/IEC 14598-5 and the CMMI

Several small organizations have expressed the need to improve their products quality. Many of those organizations are equally concerned with quality attributes of their software products and the quality of their software engineering process. Combining 14598-5 and 9126 is a

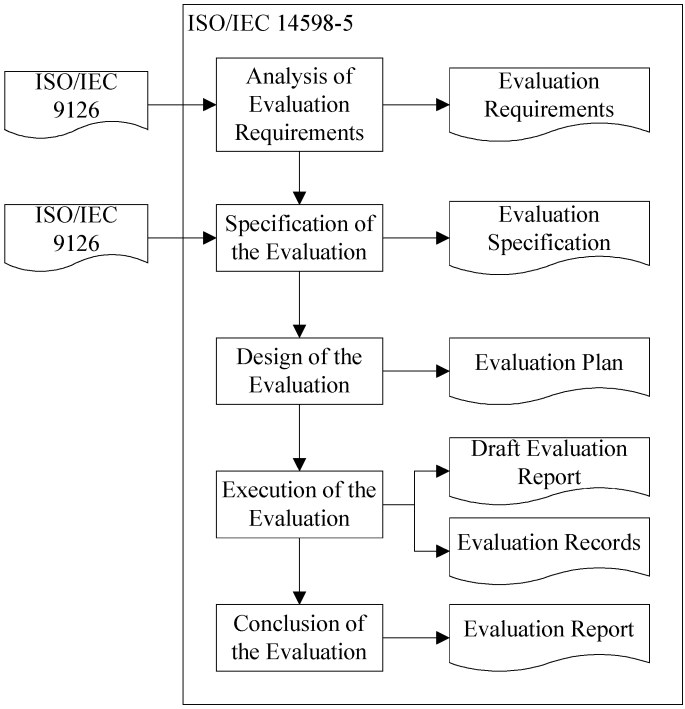

Fig. 1 The ISO/IEC 14598-5 evaluation process

natural choice to evaluate internal quality, external quality, and quality in use of a software product but there is no quality measures to apply to the software engineering process in 9126.

Maturity models such as the CMMI are used as a basis for process capability evaluation in the software engineering industry. At first, the question was raised as to whether it was appropriate to combine 14598-5 and CMMI to eventually come up with an evaluation method that will look at both the process and the product. A first glance at 14598-5 provided indications that combining CMMI and 14598-5 was appropriate. A well designed CMMI-based evaluation method driven by 14598-5 recommended activities and deliverables should provide expected results at a reasonable cost for small organizations: areas of software process improvement with a positive return on investments within a week, with a level of effort between 70 and 90 hours (team of 2 evaluators).

2. The design approach of the evaluation method

2.1. Expected usage context of the evaluation method

The designed method aims to evaluate project processes of small organizations, typically with team size of 2 to 10 developers. Those small organizations have limited budget and usually cannot afford a formal CMMI appraisal. It is expected that 2 evaluators execute the method within a week. The evaluation can certainly be applied to bigger organization. However, it is expected that the required level of effort would be higher.

Table 1 CMMI-SE/SW/IPPD/SS process areas by maturity level

Maturity levels	Process areas
5. Optimizing	Causal analysis and resolution
	Organizational innovation and deployment
4. Quantitatively managed	Quantitative project management
	Organizational process performance
3. Defined	Organizational environment for integration
	Decision analysis and resolution
	Integrated supplier management
	Integrated teaming
	Risk management
	Integrated project management for IPPD
	Organizational training
	Organizational process definition
	Organizational process focus
	Validation
	Verification
	Product integration
	Technical solution
	Requirements development
2. Managed	configuration management
	Process and product quality assurance
	Measurement and analysis
	Supplier agreement management
	Project monitoring and control
	Project planning
	Requirements management
1. Initial	None

2.2. Scope of the evaluation method

Considering available resources to design the evaluation method, a decision was made to limit the scope to software processes, leaving the evaluation of the product for a next iteration of the design. Given that, appropriate Process Areas (PA) of the CMMI were selected, typically all level 2 PA and those level 3 PA that are usually performed in small businesses that may not have organizational defined software processes.

2.3. Influences from known assessment methods

The authors had previous experiences or knowledge with known process assessment/appraisal methods, such as the CMM-Based Assessment for Internal Process Improvement (CBA IPI) (Donaway and Masters, 2001), Standard CMMI Appraisal Method for Process Improvement

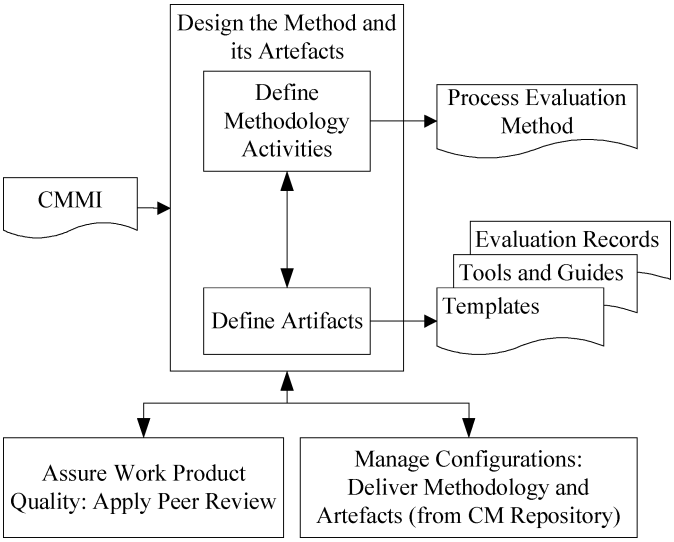

Fig. 2 The design process of the method

(SCAMPI) (Members, 2001), Software Capability Evaluation (SCE) (Averill, 1993), and other methods called "mini-assessments".

Other documents where looked at and provided influence on the design of the method, such as ISO/IEC 15504 (ISO/IEC 15504-2, 2003) and Assessment Requirements for CMMI^{SM} (ARC) (CMMI Product Team, 2001).

2.4. The design process

The design activities that were followed to develop this process evaluation method are shown on Figure 2 along with necessary support activities. The different inputs to the design process are the CMMI models and the ISO/IEC 14598-5 standard. Other standards, such as ISO/IEC 15504 and ISO/IEC 9126, and the designer's experiences with software assessment methodologies also influenced the design process. The core activities of the design process were to define the method activities and the artefacts that go along with it. Peer reviews were applied on all the artefacts that were generated. Supporting activities to this design process were defined. Theses activities were to plan the design process activities, track the design and manage the design process outputs (the artefacts created), as shown in Figure 3.

A Configuration Management (CM) tool was used from the beginning to establish baselines and maintain integrity of the method and its artefacts. At the end of the project, process evaluation method deliverables were provided from the CM repository.

2.5. The challenge of integrating ISO/IEC 14598-5 and the CMMI

The biggest challenge was to replace references made to ISO/IEC 9126 by the CMMI as the main input to the method. ISO/IEC14598-5 provides the list of output to produce and a brief overview of the evaluation report, as shown in Figure 4.

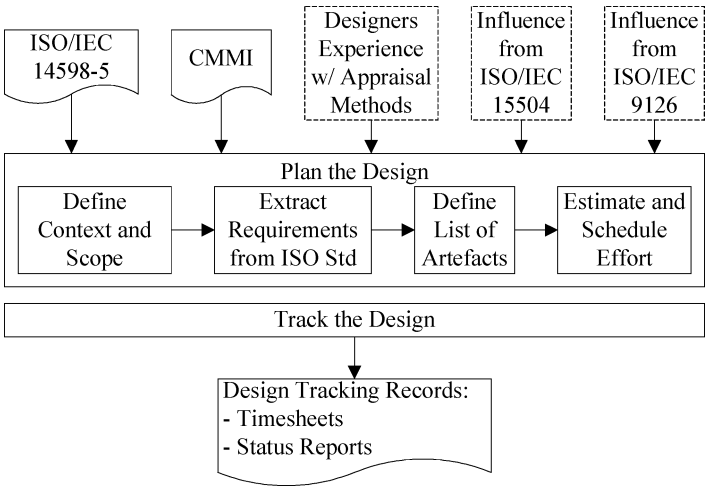

Fig. 3 Managing the design of the method

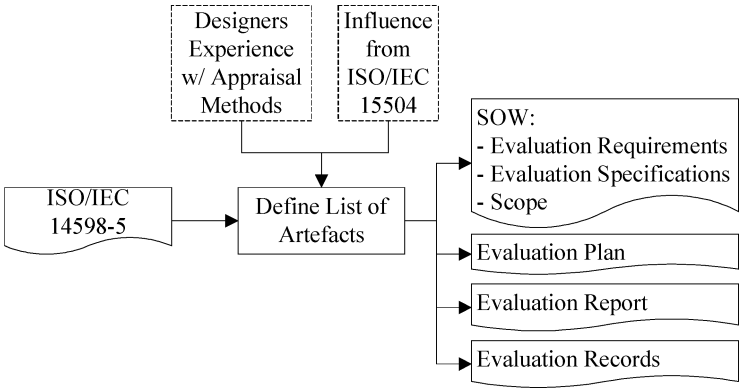

Fig. 4 Usage of ISO/IEC 14598-5 during design of the method

Figure 5 shows that the CMMI provided the content of most method artefacts. In order to design a usable process evaluation method, "synthesis" knowledge level (Bloom, 1956) of the CMMI is necessary. Experience with at least one process assessment method is also required.

2.6. Usage of a process template

A simple process template was used to document the evaluation method. The process is described in a 2-layers approach. The first level is a diagram showing a high level graphical view of the complete process showing major steps with their inputs and outputs. The second level is a more detailed textual description of the process. This way of representing the process provides a quick overview of the process that allows faster understanding of the overall process. The textual description provides the required detail of activities for performing the evaluation.

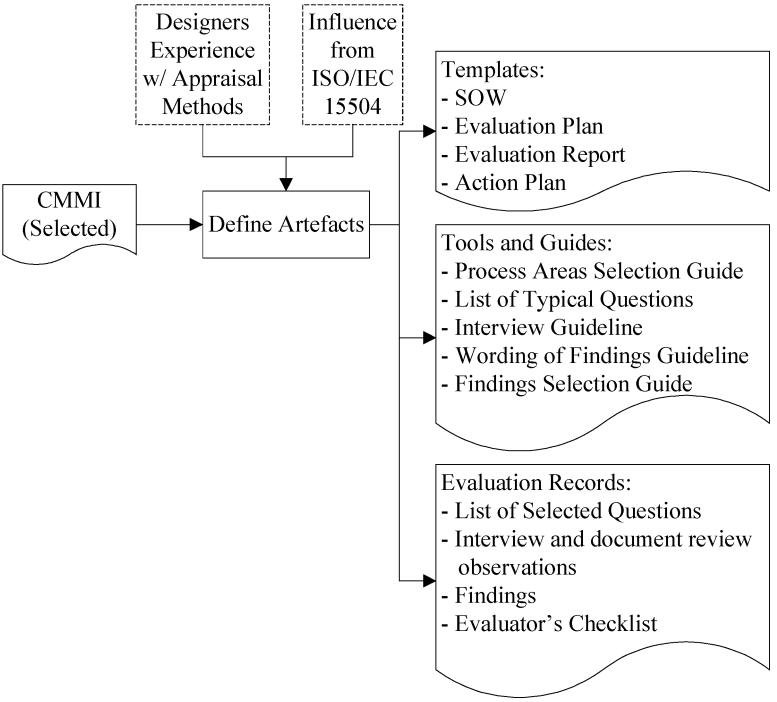

Fig. 5 Usage of the CMMI during design of the method

3. The CMMI-based process evaluation method

This section describes the overall method that was designed, its major steps, artefacts (outputs), layout of activities, templates, and compliance and differences with ISO/IEC 14598-5.

3.1. Steps of the method

The overall evaluation method is shown in Figure 6. The diagram provides a high level view of process with required inputs (to the left) and produced output (to the right). Inputs and outputs in **bold** are those created as part of the process evaluation method. The seventh step was added to provide value to the organization.

3.2. Activities

Each step is further described in the method similar to a use case description. For each step, there is:

- A short description;
- Step objectives;
- Inputs;
- Pre-conditions to the realization of the step;
- Normal flow of activities;
- Alternate flows, if appropriate;

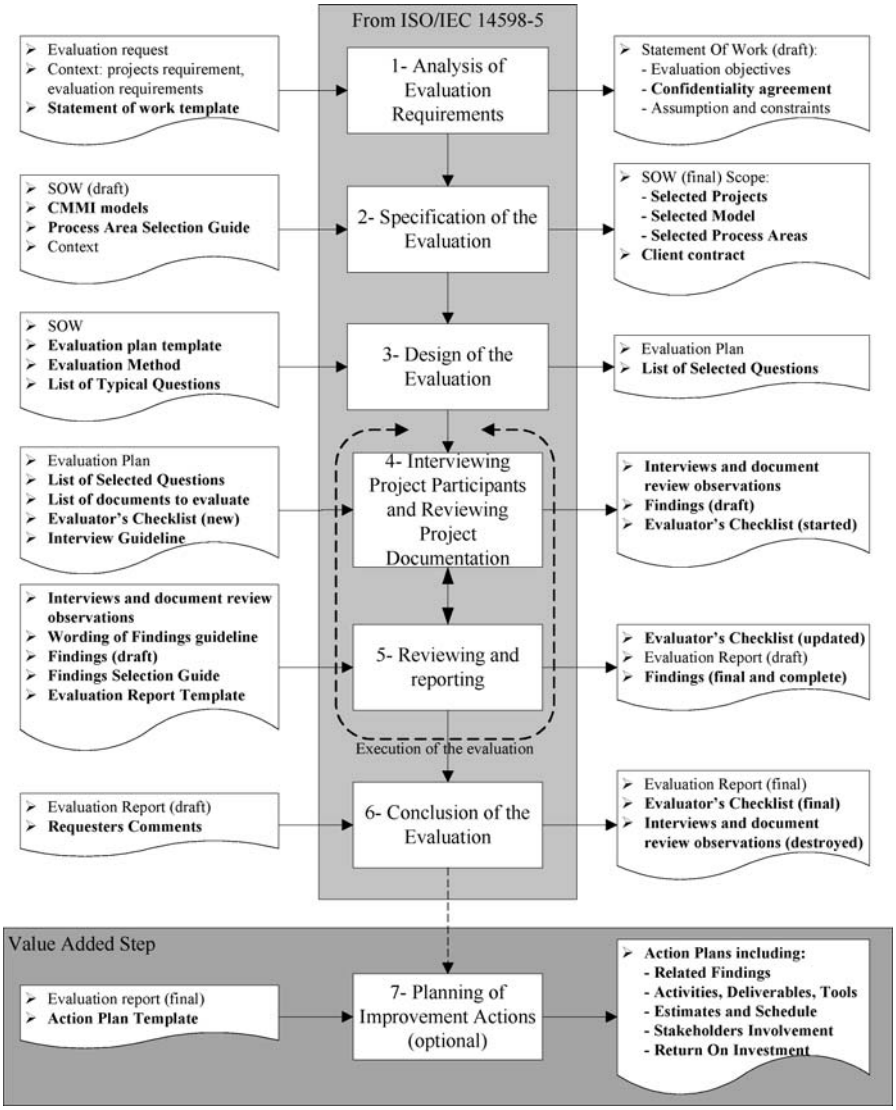

Fig. 6 Process evaluation method

- The list of verification activities that are conducted;
- Outputs;
- Post-conditions to the realization of the step; and
- The measures that need to be taken.

3.3. Artefacts

Table 2 lists the artefacts produced through the method (outputs), their objectives, and the step in which they are created or modified.

Table 2 Artefacts (outputs) of the process evaluation method

Artefact	Objectives	Realization Activities
Statement of Work	Define evaluation objectives and scope.	• Analysis of evaluation requirements • Specification of the Evaluation
Client contract	Obtain agreement with the requester. Also contains a confidentiality agreement.	• Specification of the evaluation
Evaluation plan	Provide detailed planning on reviews and interviews to be conducted	• Design of the Evaluation
List of selected questions	Provides a list of interview questions that cover the selected CMMI process areas.	• Design of the evaluation
Interviews and document review observations	Provides reminders for finding analysis. These are evaluator's personal notes and are destroyed at the conclusion of the evaluation to preserve confidentiality.	• Interviewing project participants and reviewing project documentation • Conclusion of the evaluation
Findings	Describes "sanitized"[Note1] findings found during the evaluation.	• Interviewing project participants and reviewing project documentation • Reviewing and reporting
Evaluator's checklist	Provide proof that the activities have been conducted	• Interviewing project participants and reviewing project documentation • Reviewing and reporting • Conclusion of the evaluation
Evaluation report	Present the major findings.	• Reviewing and reporting • Conclusion of the evaluation
Action plans	Provides planning of actions to improve the software process based on major findings.	• Planning of improvement actions (optional)

Note 1: It is important that no finding be related to an individual in order to get the truth from the participants.

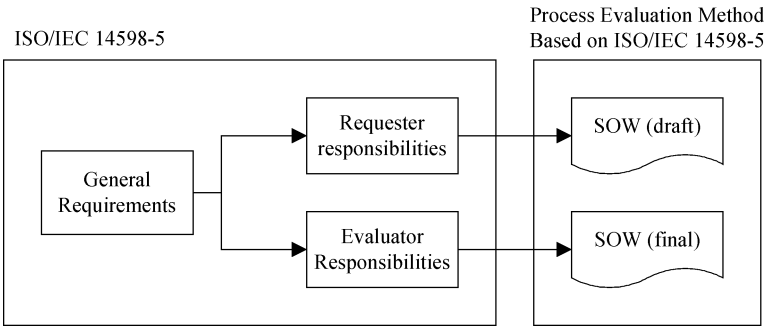

Fig. 7 Including requirements and responsibilities into a SOW

3.4. Templates

The following templates were made in order to reduce the required effort from the evaluators to generate some of the evaluation artefacts:

- Statement Of Work template;
- Evaluation Plan template;
- Evaluator's checklist;
- Evaluation Report template; and
- Action Plan template.

3.5. Compliance with ISO/IEC 14598-5

This section describes the similarities and differences between the Process Evaluation Method and the compliance for each Evaluation Process Requirement of the ISO/IEC 14598-5 standard as defined in Section 6 of the standard.

3.5.1. General requirements

The general requirements apply to the Process Evaluation Model. Where applicable, fulfilment of these requirements has been embedded in the Process Evaluation Method. A design decision was taken to include the evaluation requirements and stakeholders' responsibilities in a Statement Of Work (SOW) for which a reusable template was developed, as shown in Figure 7.

3.5.2. Analysis of evaluation requirements

This step has same name in the Process Evaluation Method. The analysis step is slightly different when used in the context of doing a CMMI evaluation for improvement. To match the context of use, evaluation requirements represent the requester's needs and objectives. Using objectives provide enough of the organizational context to allow for selection of process areas.

3.5.3. Specifying of the evaluation

This step has the same name in the Process Evaluation Method. There is no need to conduct an analysis of the product specification at the component level as proposed in ISO/IEC 14598-5

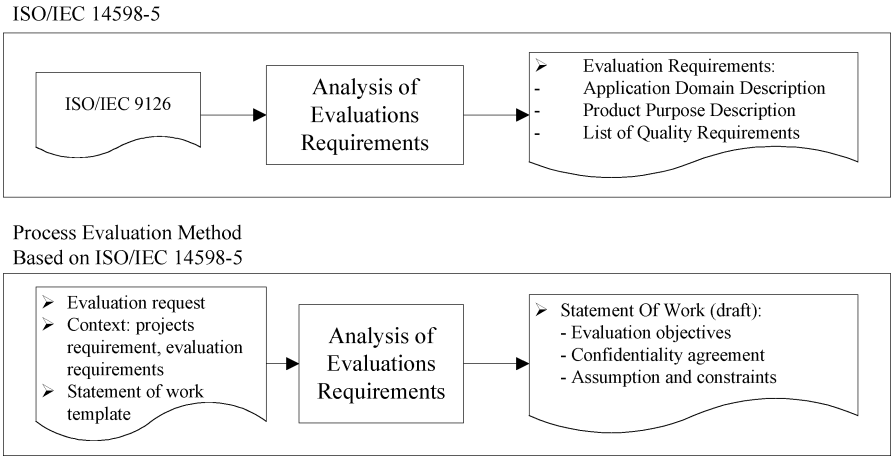

Fig. 8 Adaptation of analysis of evaluation requirements

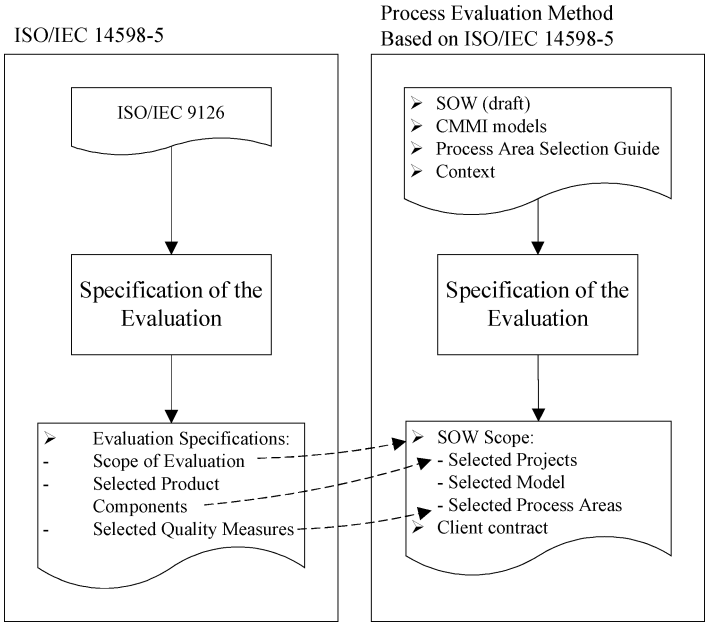

Fig. 9 Adaptation of specification of the evaluation

since the process is examined rather than a product. Instead of selecting product components as proposed, projects are selected along with their documentation that needs to be reviewed. Instead of specifying measurement, the list of process areas to evaluate is defined in order to meet evaluation objectives, as shown in Figure 9.

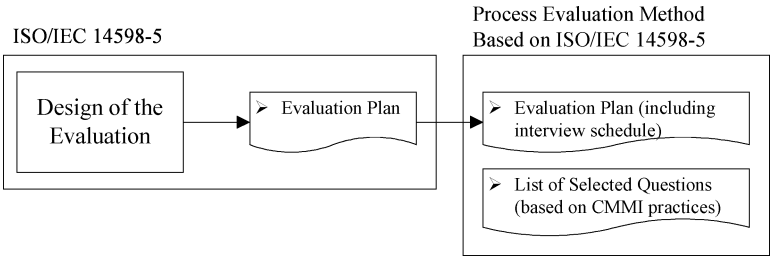

Fig. 10 Adaptation of design of the evaluation

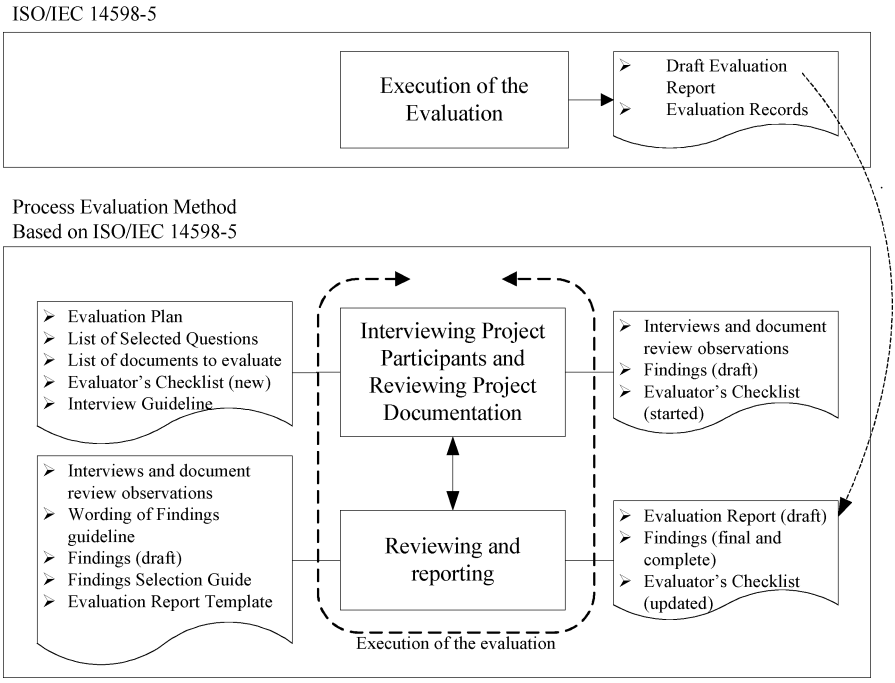

Fig. 11 Adaptation of execution of the evaluation

3.5.4. Design of the evaluation

This step has the same name in the Process Evaluation Method. The evaluation of process areas is conducted using interviews of project participants and document reviews. This kind of evaluation requires a simple tooling that consist of evaluation questions and findings tables. The questions allow for identification of the observation that will be grouped and consigned in findings tables.

The activity planning involves scheduling interviews with participants. More emphasis on people have to be made in this planning than the than the ISO/IEC 14598-5 propose. Figure 10 shows adaptation made to the plan and the added artefact "List of Selected Questions" that is required for a CMMI evaluation for repeatability purposes.

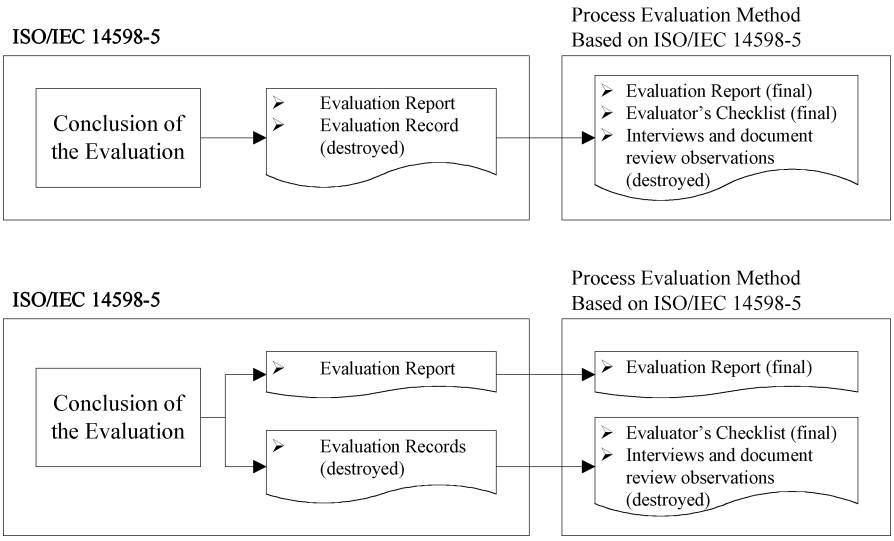

Fig. 12 Adaptation of conclusion of the evaluation

3.5.5. Execution of the evaluation

This step is split in two steps in the Process Evaluation Method, which are Interviewing Project Participants and Reviewing Project Documentation, and Reviewing and Reporting. The rationale for breaking these in two is that the interviews and reviews must be completed prior to the reporting. Another rationale to separate the two is the fact that the review and interviews may be conducted in parallel by the evaluators. Then, the evaluators agree on major findings to be reported and act upon.

3.5.6. Conclusion of the evaluation

This step has the same name in the Process Evaluation Method. This step completely conforms to the ISO/IEC 14598-5, as shown in Figure 12.

3.5.7. Planning of improvement actions

Although the ISO/IEC 14598-5 does not mention any activities happening after the conclusion of the evaluation, the authors strongly believed that a "Planning of Action" step was necessary to provide value to the requester when the objective is to improve the software process. After such an evaluation, more often the requester is left alone with a list of major findings, without knowing what to do and how to do it. Action planning uses decision criteria to prioritize actions. An action plan template is used to define a sequence of activities for each prioritized action, along with required skilled, roles and responsibilities of involved stakeholders, and a rough order of magnitude of return on investment to develop and deploy the action plan.

4. Field trials

The method has been used at various stages of its development to evaluate software engineering processes of two different organizations, in different contexts also. Feedback from evaluators was provided and used to improve the method and its artefacts.

The first field trial was done at an early method development stage in a medium size software organization (60 developers) (organization names and date of evaluations are confidential data under non-disclosure agreement) that wanted to improve its software development processes (reduce cost and schedule overruns) and learn how to evaluate software processes. An evaluation team of 6 people (one external and 5 internal) interviewed half of the development staff through 4 software development projects. The evaluation duration was 3 weeks, due to the scope size, the organization size, and the knowledge transfer on a CMMI based evaluation method to the internal team. At that time, the only artefacts available were the high-level process evaluation method and the evaluation plan template. All other evaluation artefacts were developed from scratch and used as inputs to develop the method templates. The customer stated being very satisfied at the end of the evaluation. The highest value for the organization was the action plan template with a simple way to calculate return on investment for process improvement action.

The second field trial was done when most of the method was developed (except the list of typical questions) in a medium size software organization (30 developers) that also wanted to improve its software development processes and learn how to evaluate software processes. An evaluation team of 3 people (one external and 2 internal) interviewed 12 developers through 2 software development projects. The evaluation duration was 1 week of 55 hours. It is estimated that approximately 20% of the time was spent teaching one of the internal resource about the CMMI who did not had sufficient knowledge of the model. All templates were used with minor tailoring to suit organization's needs. The customer indicated being very satisfied with the results.

Results are encouraging, leading the authors to believe that 2 evaluators can apply the process evaluation method in a small software organization with a maximum duration of 45 hours, including some action plans.

5. Open and resolved issues

5.1. Relation with ISO/IEC 9126

ISO/IEC 14598-5 is closely related to ISO/IEC 9126 since an evaluation method of a software product requires defined measures. So how 14598-5 could be used to evaluate software processes without measures? This issue was resolve by replacing all references made to 9126 by the CMMI content wherever it was deemed appropriate. As an example, process areas are used during planning of the evaluation and CMMI practices are used to design the list of typical questions.

5.2. Repeatability of the method

In order to develop a method that is reproducible and repeatable, it was decided to build up a list of typical questions. This list is to be used, at step 5 of the evaluation method,

during the project participant's interviews to help the interviewer to cover all practices from process areas that are part of the evaluation scope. At first, it was planned to develop a list of typical questions that would fit on 2 pages. But due to the number of practices in the CMMI process areas, it was preferable to link questions directly with every practice within the scope of the method. Sixteen pages of questions were written, with at least one question per specific practice through the 15 process areas and one set of questions for each generic practice. Writing down all those questions was the only efficient way towards repeatability of the method.

5.3. No measures for software processes compliance to the CMMI

ISO/IEC 9126 does not provide measures for software process compliance to a known model such as the CMMI. The process evaluation method does not either. The CMMI structure does allow for a "level" measure that can be applied either on the whole model in the staged representation or on every process area in the continuous model providing a "profile" of levels. The method does not provide such measure as a SCAMPI may give through voting and reaching consensus among evaluators. Nonetheless, the method was developed for customers who want to improve their software process performance without caring about a level. The method has been designed to uncover major findings to act upon. From the customer point of view, growth in capability should be measured by fewer cost and schedule overruns and higher customer satisfaction with their products. Measures may be added such as process evaluation method compliance during execution and evaluators satisfaction of the method usage. But is it possible to evaluate a software process and come up with a set of measures that would be repeatable and reproducible? This is still an open issue.

6. Future work

The software evaluation method that was produced in this research can be further extended to mid-size organizations. It may not be possible to complete the software evaluation process within a week in that case due to the team size increase. More time would then be spent in the execution of the method when interviewing project teams as it was experienced in the first field trial. In order to fit the need of a mid-size organization, the only change to the method would be to the evaluation plan where the activities and the schedule are defined. The time allocated to interviews would need to be increased to address a higher number of evaluation participants.

The method could also be used in organizations that were already formally assessed using the CMM (Paulk, 1993) or the CMMI models. This software process evaluation method could be used to help them perform their periodic informal assessment. To assess a CMM accreditation, the method should then be slightly modified to fit the CMM Key Process Areas (KPA). It is not planned at this stage to try out the method in such an organization, but it is recognized that the method is suitable to fit this need.

It is also considered to add measures to this software evaluation method to measure the model compliance and satisfaction (how it is seen from its users). These measures could be added to the execution of the method through the completion of a satisfaction questionnaire.

This would then help the evaluators in assessing the part of the model (the processes) that are not effectively used or not complete. The development of such a satisfaction questionnaire involves time to build up the right question that would address all the selected process areas of the CMMI model.

7. Conclusion

This research objective was to develop a CMMI-based evaluation method driven by ISO/IEC 14598-5 recommended activities and deliverables in order to produce a software process evaluation method at a reasonable cost for small organizations. A design approach was developed to help defining the expected context usage of the method, the scope of the evaluation and the steps needed to produce the software process evaluation activities and deliverables. The method that was produced contains 6 different steps that are aligned with the ISO/IEC 14598-5 evaluation process. One extra step was added to ensure that the findings resulting from the evaluation were addressed in action plans where priorities were established and responsibilities were assigned to the relevant stakeholders. The field trials that were done using this method have shown encouraging results and have demonstrated that the research objective can be met. Using this method, indications demonstrate that it is possible to perform a software process evaluation in a small organization (team size of 2 to 10 people), within one week including a first draft of action plans (1 to 4) to address some of the major findings.

References

Averill, E. 1993. Software capability evaluation (SCE) version 1.5 method description, Pittsburgh, July, eds software engineering institute, Carnegie-Mellon University.

Bloom, B.S. 1956. *Taxonomy of Educational Objectives: The Classification of Educational Goals: Handbook I, Cognitive Domain*. New York, Toronto, (eds.) Longmans, Green.

CMMI Product Team, Appraisal Requirements for CMMISM, Version 1.1 (ARC, V 1.1). (2001). Pittsburgh, December, eds Software Engineering Institute, Carnegie-Mellon University.

CMMI Team, Capability Maturity Model Integration for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE/SW/IPPD/SS), Version 1.1, Continuous Representation, 2001. Pittsburgh, December, eds. Software Engineering Institute, Carnegie-Mellon University.

Donaway, D.K. and Masters, S. 2001. CMM®-Based Appraisal for Internal Process Improvement (CBA IPI) Version 1.2 Method Description,. Pittsburgh, November, eds. Software Engineering Institute, Carnegie-Mellon University.

ISO/IEC 14598-5. 1998. Information Technology—Software Product Evaluation—Part 5: Process for Evaluators, International Organization for Standardization.

ISO/IEC 15504-2. 2003. Software engineering—Process assessment—Part 2: Performing an assessment, International Organization for Standardization.

ISO/IEC 9126-1. 2001. Software engineering—Product quality—Part 1: Quality model, International Organization for Standardization.

Members of the assessment method integrated team, standard CMMI appraisal method for process improvement (SCAMPI) version 1.1: Method Definition Document, 2001. Pittsburgh, December, eds Software Engineering Institute, Carnegie-Mellon University.

Paulk, M. (1993). Capability maturity model for software, version 1.1. Pittsburgh, February, (eds.) Software Engineering Institute, Carnegie-Mellon University.

Sylvie Trudel has over 20 years of experience in software. She worked for more than 10 years in development and implementation of management information systems and embedded real-time systems. Since 1996, she works as a process improvement specialist, implementing best practices into organizations processes from CMM and CMMI models. She performed several CMM and CMMI assessments and participated in many other CMM assessments such as CBA IPI, SCE, and other proprietary methods. She obtained a bachelors degree in computer science in 1986 from Laval University in Québec City and a Masters degree in Software Engineering at École de Technologie Supérieure (ÉTS) in Montréal. Sylvie is currently working as a software engineering advisor at the Centre de Recherche Informatique de Montréal (CRIM).

Jean-Marc Lavoie has been working in software development for over 10 years. He performed and published a comparative study between the guide to the SWEBOK and the CMMI in 2003. Jean-Marc obtained a bachelor degree in Electrical Engineering. He is pursuing a Masters degree in Software Engineering at École de Technologie Supérieure (ÉTS) in Montréal while working as a software architect at Trisotech.

Marie-Claude Pare has been working in software development for 7 years. Marie-Claude obtained a bachelor degree in Software Engineering from École Polytechnique in Montréal. She is pursuing a Masters degree in Software Engineering at École de Technologie Supérieure (ÉTS) in Montréal while working as a software engineer at Motorola GSG Canada.

Dr Witold Surynis a Professor at the École de technologie supérieure, Montreal, Canada (engineering school of the Université du Québec network of institutions) where he teaches graduate and undergraduate software engineering courses and conducts research in the domain of software quality engineering, software engineering body of knowledge and software engineering fundamental principles. Dr Suryn is also the principal researcher and the director of GELOG : IQUAL, the Software Quality Engineering Research Group at École de technologie supérieure. From October 2003 Dr. Suryn holds the position of the International Secretary of ISO/IEC SC7 – System and Software Engineering.