PAPER

Software Process Capability Self-Assessment Support System Based on Task and Work Product Characteristics: A Case Study of ISO/IEC 29110 Standard

Apinporn METHAWACHANANONT[†a)], **Marut BURANARACH**[†], **Pakaimart AMSURIYA**[†], **Sompol CHAIMONGKHON**[†], **Kamthorn KRAIRAKSA**[†], ***Nonmembers*, *and* Thepchai SUPNITHI**[†], ***Member***

SUMMARY A key driver of software business growth in developing countries is the survival of software small and medium-sized enterprises (SMEs). Quality of products is a critical factor that can indicate the future of the business by building customer confidence. Software development agencies need to be aware of meeting international standards in software development process. In practice, consultants and assessors are usually employed as the primary solution, which can impact the budget in case of small businesses. Self-assessment tools for software development process can potentially reduce time and cost of formal assessment for software SMEs. However, the existing support methods and tools are largely insufficient in terms of process coverage and semi-automated evaluation. This paper proposes to apply a knowledge-based approach in development of a self-assessment and gap analysis support system for the ISO/IEC 29110 standard. The approach has an advantage that insights from domain experts and the standard are captured in the knowledge base in form of decision tables that can be flexibly managed. Our knowledge base is unique in that task lists and work products defined in the standard are broken down into task and work product characteristics, respectively. Their relation provides the links between Task List and Work Product which make users more understand and influence self-assessment. A prototype support system was developed to assess the level of software development capability of the agencies based on the ISO/IEC 29110 standard. A preliminary evaluation study showed that the system can improve performance of users who are inexperienced in applying ISO/IEC 29110 standard in terms of task coverage and user's time and effort compared to the traditional self-assessment method.

key words: *software development capability, gap analysis, self-assessment, knowledge-based approach*

1. Introduction

A key driver of software business growth in developing countries is the survival of software small and medium-sized enterprises (SMEs). Quality of products is a critical factor that can indicate the future of the business by building customer confidence. Software development agencies need to be aware of meeting international standards in software development process. In practice, consultants and assessors are usually employed as the primary solution, which can impact the budget in case of small businesses. Self-assessment tools for software development process can potentially reduce time and cost of formal assessment for software SMEs. However, the existing support methods and tools are largely insufficient in terms of process coverage and semi-automated evaluation.

Selecting standards for improving the organization's software development process needs to be rational and objective [1]–[3]. Various software process standards have been continuously updated and published such as CMMI, ISO/IEC 12207, ISO/IEC 29110, COBIT [4]–[17]. Each standard has different objective and scope including assessment method. Self-assessment based on such a standard is important to software development companies. Specifically, software agencies do not know what to improve in order to pass the standard criteria. Therefore, gap analysis can help in guiding them to develop and improve the software development process more properly. Furthermore, readiness assessment and review help to support the agencies to increase its chances of success.

In our project, we propose to build a support system for software process capability assessment. The system has been embedded the assessment knowledge including process reference model (such as ISO/IEC 29110) and process assessment model (ISO/IEC 15504, ISO/IEC 33000) to encourage the users to be able to analyze a gap between existing software process and a required standard and to do self-assessment before official appraisal appropriately. We focus on the ISO/IEC 29110 standard, which is currently promoted in many countries worldwide and is less complex than the other standards which make it more suitable for SMEs. We adopted a knowledge engineering approach in developing a domain knowledge of the ISO/IEC 29110 standard in ontology form. The ontology was used as a conceptual agreement among experts, system and database designers. The assessment knowledge based on the standards and experts was made explicit as assessment knowledge base in supporting the system.

Our approach differs from the traditional self-assessment tools of ISO/IEC 29110 based on task list which has a disadvantage that it is difficult to understand what to do in each task list. Thus, we made the details of task list more explicit by breaking down into task characteristics (TC). In addition, we identified related work product characteristics (WPC) and defined mapping rules between task characteristics and work product characteristics accordingly. As a result, the proposed tool can support the users to

Manuscript received August 30, 2018.
Manuscript revised June 11, 2019.
Manuscript publicized October 17, 2019.
[†]The authors are with the National Electronics and Computer Technology Center (NECTEC), Thailand.
a) E-mail: apinporn.methawachananont@nectec.or.th
DOI: 10.1587/transinf.2018EDP7303

understand and perform self-assessment effectively. A preliminary analysis has shown enhancement in process assessment for users who are inexperienced in applying ISO/IEC 29110 standard in terms of task coverage and time and effort using the support system over the manual self-assessment process of the ISO/IEC 29110 standard. Finally, we discuss some limitations and potential extension of our framework.

2. Background

2.1 Software Process Assessment Models

The concept of process assessment that theprocesses are evaluated to take into account the capability of the process implementation following an established standard. And this measure may lead to improve those existing processes [17]. The agency needs to set objectives, scope and constraints, as well as prepare a resource including all stakeholders for process assessment. Each standard specifies its evaluation guidelines such as CMMI, using SCAMPI [18], ISO standards, whether it is ISO/IEC 12207 or ISO/IEC 29110, identifying ISO/IEC 15504 (or SPICE: Software Process Improvement and Capability Determination) [19]–[23] which is being updated to ISO/IEC 33000 [24]–[31] which consists of PRM (Process Reference Model), PAM (Process Assessment Model) and mapping procedure to rate the process capability.

2.2 ISO/IEC 29110 Standard

ISO/IEC 29110 standard [8]–[15] is based on the ISO/IEC 12207 standard, which is a full-fledged process, to allow small businesses to upgrade their product quality through an acceptable process. This standard is adopted in 2010 consisting of five parts: overview, framework and taxonomy, assessment guide, basic profile specifications and management & Engineering guide for basic profile. Nevertheless, ISO/IEC 29110 focuses on two essential processes: Project Management (PM) and Software Implementation (SI), as shown in Fig. 1.

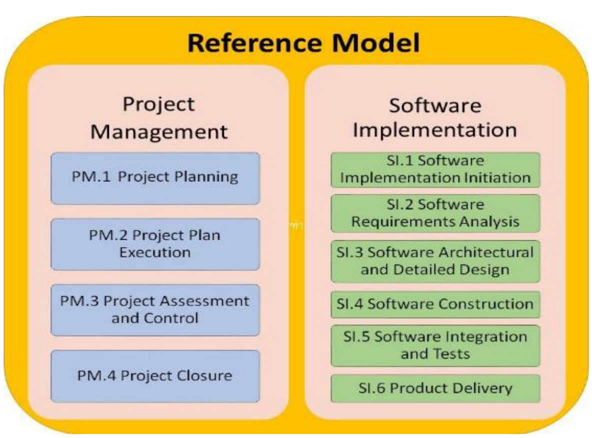

Fig. 1 Process reference model of ISO/IEC 29110 [8]

PM process consists of four main activities: Project Planning, Project Plan execution, Project Assessment and Control, and Project Closure. All four activities are linked and yielded. The SI process composes of six activities: Software Implementation Initiation, Software Requirement Analysis, Software Architecture and Detailed Design, Software Construction, Software Integration and Test, and Product Delivery. All of these activities will result in the development of software or product that meets the standard.

2.3 Related Work

Currently, there are different methods and tools for evaluating the use of software development processes in accordance with ISO/IEC 29110 for SMEs/VSEs: 1) Survey/Studies such as OWPL Tool (Observatoire Wallon des Pratiques Logicielles Tool by CETIC: Belgium) for self-assessment in 2015, with emphasis on 10 processes and 39 main tasks of SI and PM processes. (Entry Profile Level) including NOEMI Tool (CETIC: Luxemburg) [32]. The survey such as CHAOS survey by questionnaire is provided in several languages: English, French, German, Korean, Portuguese, Russian, Spanish, Thai, and Turkish. The data will be analyzed and summarized for further process improvement [32]. 2) Methods and Tools such as Deployment Packages (INCOSE: International Council on Systems engineering) [33], [34], MARES (Brazil), Full Assessment [35], TOPS, FAME, RAPID (Australia - focused on CM), SPM, EAP, Micro-Evaluation, METvalCOMPETISOFT (Botswana, South Africa), SPINI, EvalProSoft, ADEPT, and MA-MPS [36] by supporting tools including paper forms, software tool, electronic process guides, data collection. These methods and tools are not yet comprehensively implemented. In addition, the tools have been developed to fully comply with ISO/IEC 15504 for the assessment with the use of other international standards: ISO 12207, COBIT, etc. in paper form, spreadsheet, and software (such as ConCatenator & SPiCE Master Tool develop questionaire/apply to various standards/record/rating/report in word and powerpoint formats) [37], SPiCE 1-2-1 (self-assessment for strength and weakness/graphic and report) [38], SPiCE-Lite (self-assessment for specific processes taking 3-4 hrs./report like the SPiCE 1-2-1) [39].

Our approach differs from these methods and tools which are mostly depending on manual process based on task list. Our approach focusses on support the users to understand what to do in each task list. Thus, the details of task list were made more comprehensible and explicit by breaking down into task characteristics which were offered and verified all key tasks in each task list by certified consultants and assessors. The related work product characteristics consisting of all key components in each work product and mapping rules between task characteristics and work product characteristics were also identified and verified by certified consultants and assessors. As a result, our approach can lead the users who are inexperienced in applying ISO/IEC

29110 standard to do process assessment by themselves efficiently.

3. Framework

3.1 Conceptual System Architecture

The layered architecture of the knowledge-based software process assessment support system is shown in Fig. 2.

The domain reference model layer consists of the domain reference model (ISO/IEC 29110) focusing on two main processes: project management (PM) and software implementation (SI) and the task and work product characteristics (TC & WPC), described in the Sect. 3.2.2. The ISO/IEC 29110 ontology, described in the Sect. 3.2.1, is developed. The ontology represents a domain knowledge blueprint that can be shared and reused in this project and other projects implementing the ISO/IEC 29110 standard.

The knowledge and data layer consists of three main components: Knowledge base, Software project database and Rating and recommendation engine. The knowledge base consists of ISO/IEC 29110 Ontology, TC & WPC mapping rules and Recommendation rules. The software project database contains the software project information collected from participating software companies. The rating and recommendation engine generates the scores and ratings based on the database, knowledge base and formulas described in the Sect. 3.3. In addition, recommendation for the users is also generated based on recommendation rules in form of decision tables.

The application layer consists of a functional system for evaluating processes and providing recommendation consistent with the ISO/IEC 29110 standard. It also supports authentication and software project information management functions. The services are divided into 3 types: 1) Manage project profile (Information Management Service), 2) Assess initially of the gap between current process and the standard (Self/Pre-assessment Service), and 3) Provide guidance on how to adapt the organizational processes to the required standard (Recommendation Service). All of these services will enhance the user's ability to assess the capabilities of an enterprise software process before adopting international standards (Gap Analysis), examine self-processes before getting an official assessment (Readiness Review), and self-assessment to improve the quality (Self-Assessment), self-renewal for a standard renewal (Surveillance), gaining recommendation to fill the gap, making it possible to improve and develop the process faster.

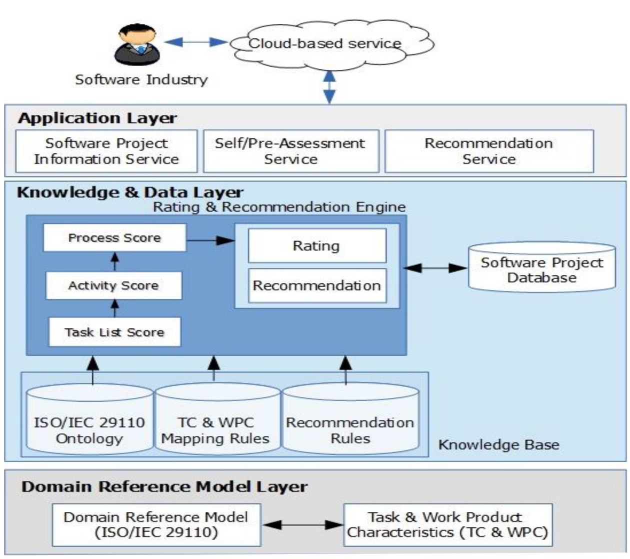

Fig. 2 Layered architecture of ISO/IEC 29110 self-assessment support system

3.2 Knowledge Engineering Process

3.2.1 Ontology

In developing a domain knowledge model, sources of knowledge are ISO/IEC 29110 (Software engineering–Life cycle profiles for Very Small Entities (VSEs)) and ISO/IEC 33000 (Information technology – Process assessment) [24]–[31] and certified consultants/assessors. Domain knowledge modeling of the ISO/IEC 29110 standard was designed in ontology form. The ontology was used as a conceptual agreement between experts and system and database designers during the entire project. Figure 3 shows some main concepts that are involved with the assessment process and criteria. Six key concepts of ISO/IEC 29110 include Project, Process, Activity, Role, Task List and Work Product. Assessment of Project relies on assessment of its Process, Activity and Task List respectively. Task Listis related to Role and Work Product. Two additional concepts, proposed by our framework, are Task Characteristics (TC) and Work Product Characteristics (WPC). The two concepts are defined as the sub-components of Task List (TL) and Work Product (WP) respectively. Their relation provides the links

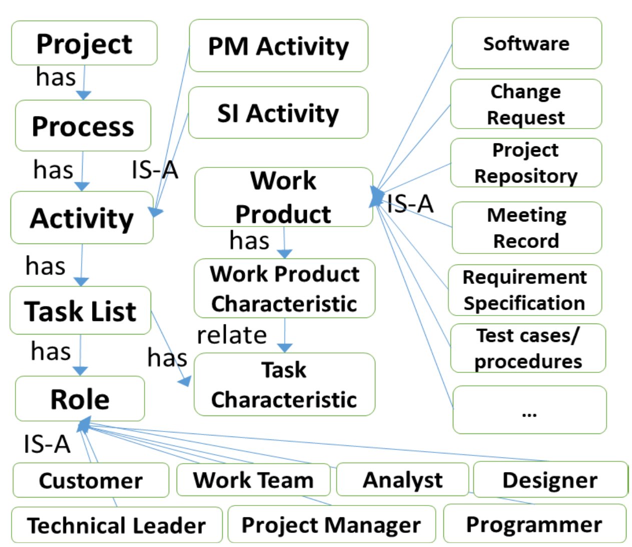

Fig. 3 Domain ontology of ISO/IEC 29110 standard

between Task List and Work Product. Subclasses and properties of the concepts are additionally defined. For brevity, only partial subclasses are shown. The ontology is primarily used in guiding design of database, user interfaces and reports.

3.2.2 Task and Work Product Characteristics

In building a knowledge base for software process assessment support, Task-related knowledge was extracted to create objective evaluation criteria: Task Characteristics (TC), Work Product Characteristics (WPC) and their relationships. TC is the key subtask of TL that is analyzed by experts from the TL specified in the standard. These TCs constitute each TL and are related to the WPs of the TL which are specified in the standard. WPC is a key component of each WP that is analyzed by experts from WP description in the standard. Each TC will be associated with at least 1 WPC. The WPC can support users to verify the evidence associated with each TL. This can help to provide recommendation for improving the quality of each evidence. In this research, we have identified 170 TCs of the 67 TLs and identified 275 WPCs of the 22 WPs in accordance with the ISO/IEC 29110 standard. In addition, we have defined over 420 relationships between the TCs and the WPCs. TC under PM process is identified as PTC which the first letter "P" comes from the first letter of "PM". The definition of STC is the same as PTC but changes from PM process to SI process. PTL and STL refer to TL of PM and SI processes in sequence. For example, as presented in Table 1, we found that a TL involved in maintenance documentation is not only relevant to Maintenance Document but it is also related to Change Request, Software Configuration, Software Repository, and Verification Result. The exemplified relationships can be elaboratedas follows

STC.6.3.1 "Document/update" is related to WPC.4.1 "Reference to all elements developed during implementation" and WPC.4.2 "Environment identification for development and testing".

STC.6.4.1 "Verify consistency with software configuration" is related to WPC.4.3 "Application status (verified and baselined)".

STC.6.4.2 "Document the Verification Results" is related to WPC.21.43-WPC.21.49 including item to verify, participants, date including duration and place, verification check-list/criteria, passed/failed/pending items of verification, defects identified during verification, and follow up information, respectively.

STC.6.4.3 "Correct until approved" is related to WPC.4.4 (Approved by TL).

STC.6.4.4 "Request changes if needed" is related to WPC.2.1-WPC.2.6 involving in purpose of change, request status, impacted system, impact to associated documentation, informed change to stakeholder, and application status (initiated, evaluated, and accepted), respectively.

STC.6.5.1 "Incorporate to the baseline" is related to WPC.14.23 "Maintenance Documentation uniquely identified" and WPC.14.24 "applicable status (delivered and accepted).

The list of TCs, WPCs and their mappings rules are maintained in a spreadsheet format, as shown in Table 1, and automatically imported to the system knowledge base as decision tables. The use of decision tables maintained in external spreadsheets allows the TC and WPC lists and their mapping rules to be managed by the domain experts separately outside the program.

Table 1 Example of relation between task characteristics and work product characteristics

PTC/STC \ WPC	Change Request			Maintenance Document				SW Configuration		Verification Result		
	2.1	2.2	....	4.1	4.2	4.3	4.4	14.23	14.24	21.43	21.44	.....
STC.6.3.1 (Document/Update)				X	X							
STC.6.4.1 (Verify Consistency)						X						
STC.6.4.2 (Verification Result)										X	X	X
STC.6.4.3 (Approved by TL)							X					
STC.6.4.4 (Request Change)*	X	X	X									
STC.6.5.1 (Configuration)								X	X			

3.3 Assessment Grading Formula

Assessment processor calculates scores of task characteristics using the rules based on the relationship between task characteristics (TC) and work product characteristics (WPC). Calculating each TC score of each TL under the PM process will be assigned a corresponding reference code, such as PTC1.1.1, a reference to a TC score of TC having the reference code 1.1.1 of TL with reference code 1.1 under PM with reference code 1. Likewise, the calculation of the score of each TL under the PM process is assigned a corresponding code. For example, PTL1.1 represents the TL score of TL having the reference code 1.1 under the PM process with reference code 1. The STC and STL can be considered in the same as PTC and PTL but use SI process instead. Work Product Characteristics scores are processed the completeness of the work product (WP). Each task characteristic (TC) score is calculated from the scores of related work product characteristics (WPC) by the formula 1. After that each task list (TL) can be formulated from all scores of associated task characteristics by the formula 2. These scores of in-depth data can be linked to calculate the score of the activity and the process by existing methods continuously.

Formula 1: To calculate the score of each task characteristic of PM process

$$PTC_{(j)} = \frac{\sum_{r=1}^{n} WPC_r}{n}$$

Where $PTC_{(j)}$ is the score representing the coverage of Task Characteristic with the j reference code under the Task List of PM process, based on the relationship specified in the knowledge base. WPC_r is the value of each WPC associated with PTC in order of r, starting from 1 to n, which is the total number of WPCs involved. The WPC value is set to 1 if found, and 0 if no evidence is found.

Formula 2: To calculate the score of each task list of PM process

$$PTL_{(j)} = \frac{\sum_{i=1}^{n} PTC_i}{n}$$

Where $PTL_{(j)}$ is the score representing the coverage of Task List with the j reference code under the PM process, based on the relationship specified in the knowledge base. PTC_i is the value of each PTC associated with PTL in order of i, starting from 1 to n, which is the total number of PTCs involved. The range of PTC value is between 0 to 1.

The activity level score is then calculated following the ISO/IEC 33000 [14], [15]. The score for each activity is the average of all TL scores under that activity expressed as a percentage, which can be rated as N/P/L/F based on the condition: 0-15% for N, which means no achievement, >15–50% is P, which means partially achieved, >50–85% is the L that shows that largely achieved, >85–100% as F represents the fully achieved.

Each process is then evaluated at the capability level based on ISO/IEC 33000. Currently, there are two capability levels: Capability Level 0 (CL0) and Capability Level 1 (CL1). If all activities under this process are rated as L or F only, the process will be evaluated as CL1, indicating that the organization is implementing the software process and achieving its objectives within the defined process scope. The other cases are evaluated as CL0, which states that the organization does not implement software process or implement it but not achieve the intended purpose.

Finally, the assessment result of the ISO/IEC29110 implementation of the proposed method for each organization is based on the consideration of the capability of all processes (PM and SI), which the organization can achieve if all processes are rated as CL1, otherwise it will fail.

4. A Case Study: ISO/IEC 29110 Self-Assessment System

4.1 User Input and Report Design

The system was designed to be simple for all users. Participating users can access the service via the DIGEST System website (https://digest.openservice.in.th). While users start assessing the process, the user can specify the scope of the activity as shown in the first part of Fig. 4 (e.g. "Scope of Activities: PM.1-PM.4, SI.1-SI.6" for the project: DRM-Dynamic Resource Management) to be evaluated according to ISO/IEC 29110 standard, which consists of two main processes and ten activities that can be implemented in the project as shown in the second part of Fig. 4 (presenting all activities under the identified scope such as "PM.1 Project Planning, PM.2 Project Plan, etc."). Moreover, the users can import the task information via the evidence or the work product information relating to the identified tasks as shown in the third part of the Fig. 4 (presenting all task lists PM.1.1 and PM.1.2 under specified activities PM.1 involving related task characteristics and their work product characteristics. In this part, the user has to confirm the artifacts made under the task). The task or work product information associated with the user input is then retrieved following the established relationship rules. An input screen is shown in Fig. 4.

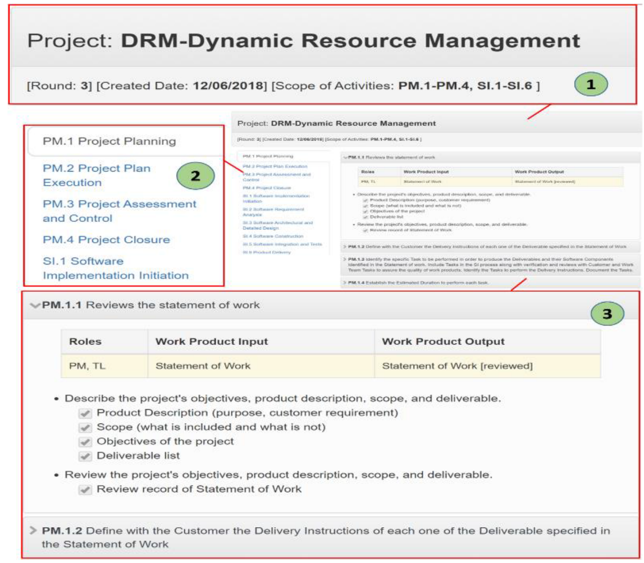

Fig. 4 Example of pre-assessment input form

The process assessment result can be viewed in tabular and graphic format as shown in Fig. 5. The implementation of the task characteristics will be also reported. Furthermore, the assessment result can present in two capability levels detailed as follows:

- Activity level, the system can view the capability of implementing each activity as shown in the second part (presenting the capability rating of each activity calculated from the formula referring to the Sect. 3.3) and the third part (presenting % achievement or implementation of each activity of PM and SI process with graph) of Fig. 5.
- Process level, the system can show the capability of implementing each processas shown in the first part (presenting the self-assessment results of the organization based on the capability level of every processes in accordance with ISO/IEC 29110 standard) and forth part (proposing activities under the process that should focus on improving to achieve standards) of Fig. 5.

4.2 Preliminary Evaluation

To ensure the benefits of the proposed approach, the research team has made a preliminary evaluation in term of spent time and task coverage, by comparing with the existing method. The assessment results were obtained by

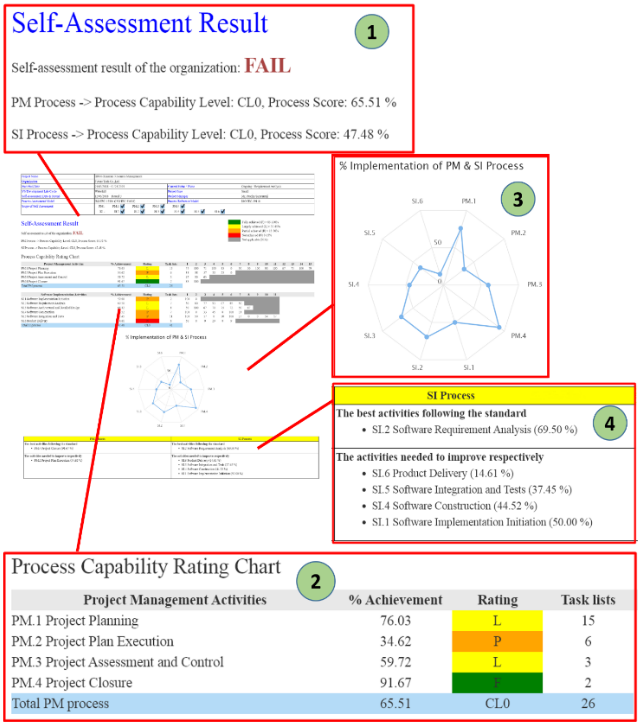

Fig. 5 Example of an assessment result report

conducting the tests with 16 users who have been involved in software development process. Two test cases, i.e. existing and proposed methods, were conducted in accordance with the software implementation process, focusing on the activity "Software Requirement Analysis" (SI.2) of ISO/IEC 29110, which consists of 7 task lists (SI.2.1-SI.2.7). The test cases are detailed as follows.

- Test Case#1-Existing method: The test is based on a paper-based assessment template for evaluating Task List that adopted the current Deployment Package [40].
- Test Case#2–Proposed method: The test according to the proposed method is to evaluate Task Characteristics and Work Product Characteristics, using the DIGEST system. The scores are calculated to assess each Task List respectively.

Details and conditions of the test cases are defined as follows:

1. The project used inthe test cases involved a software project that supports software engineering process in a company, focusing on the activity "Software Requirement Analysis".
2. Sixteen users from private and government organizations participated in this evaluation. These users have been involved in software development process but have no experience in applying ISO/IEC 29110 standard before the tests. Before conducting a preliminary evaluation, we provided an overview training of ISO/IEC 29110 standard and the assigned project. Therefore, each user has basic knowledge about ISO/IEC 29110 standard but not the detailed assessment knowledge according to the standard prior to the tests.
3. Each user has to test both cases, including the existing self-assessment method and the proposed self-assessment method using the process SI.2.1, SI.2.2, SI.2.3, SI.2.4, SI.2.5, SI.2.6, SI.2.7 in ISO/IEC 29110. The order of the tests was the existing method followed by the proposed method for every user. The order of the test was fixed because the detailed Task Characteristics and Work Products shown by the DIGEST system could affect the answers made in the paper-based template otherwise. We understand that the fixed order could also introduce some learning effect. However, we believe that this effect was not significant because the users have been overviewed about the assigned project before the tests. So the learning process about assessing the project mostly occurred before the tests, not during the tests.
4. The time spent in each method of each user (16 users), starting from beginning the task to submitting the answers, is recorded.
5. The number of subtasks under each TL in each method of each user is recorded. Average of task coverage of all TLs (SI.2.1-SI.2.7) is calculated from the average number of correct task characteristics (#TC) for each task list in each method.

The percentage of improvement in completion time and improvement in task coverage are calculated as follows.

$$\%\mathit{Improvement\ in\ Completion\ Time} = (T_e - T_p) \times 100/T_e$$

Where T_e is the average completion time of the existing method (minutes), T_p is the average completion time of the proposed method (minutes).

$$\%\mathit{Improvement\ in\ Task\ Coverage} = (TC_p - TC_e) \times 100/TC_p$$

Where TC_p is the number of tasks involved (or task characteristics) in each task list of the proposed method, TC_e is the number of tasks involved in each task list of the existing method.

Test results based on these measurements are presented in Table 2. The average improvement in completion time (%) is $((32.3 - 11.6) \times 100)/32.3$ or 64%. The average improvement in task coverage (%) of the Task List SI.2.1, SI.2.2, SI.2.3, SI.2.4 and SI.2.7 is $(50 + 20 + 51 + 43 + 50)/5$ or 43%.

In addition, we consider the variance of task coverage of both methods. The variance of the time spent of all 16 users in the proposed method is 15.46 which is less than that in the existing method which is equal to 33.43. The variance of task coverage in the proposed method is equal to zero in each task list SI.2.1-SI.2.7 because the system only shows the related tasks and work products. The variances of task coverage in the existing method are equal to 0, 0.65, 2.13, 0.73, 0, 0, and 0.80 for the task lists SI.2.1-SI.2.7 respectively, which are greater than those of the proposed method. Thus, using the support system would be less dependent on

Table 2 Test results comparing between existing and proposed methods

Model	Average of Time Spent (Minutes)	Average Number of Tasks Involved in each task list (#TC)*						
		SI.2.1	SI.2.2	SI.2.3	SI.2.4	SI.2.5**	SI.2.6**	SI.2.7
Existing Method (Template)	32.3	1.0	2.4	3.4	2.3	0.0	0.0	1.0
Proposed Method (DIGEST System)	11.6	2.0	3.0	7.0	4.0	0.0	0.0	2.0
Improvement (%)	64	50	20	51	43	-	-	50
		43						

Note:
*Average Task Coverage is the average number of tasks involved or task characteristics (#TC) according to the standard in each given task list (SI.2.1-SI.2.7) that has been inputted by all users (16 users).
** Since the task list SI.2.5 and SI.2.6 are optional and are not required in the assigned project. Therefore, they cannot be considered for improvement in task coverage. It should be noted that the average numbers of task characteristics that the users entered incorrectly for SI2.5 and SI2.6 are 0.7 and 1.4 respectively for the existing method and are both zero for the proposed method. This shows that the proposed method can help to prevent the users from inputting erroneous tasks for such task lists.

individual skills/expertise of users compared to the existing method. In addition, the users receive more comprehensive information in case of tool-supported work because the tool can automatically present task and relevant products that support the users in entering less erroneous data and resulting in more complete information. Higher degree of task coverage under each task list can lead to more relevant task evaluations, leading to better self-assessment results. Thus, the preliminary results indicated that the proposed method by DIGEST system can potentially support the users to conduct more effective self-assessment.

5. Discussion and Conclusion

In this paper, we present a knowledge-based support system for software process capability assessment of the ISO/IEC 29110 standard. Our framework adopted a knowledge engineering approach in modeling embedded the knowledge of the standard guideline and experience of assessors and consultants to consolidate them in an explicit, formal and reusable knowledge form. The approach differs from existing system development that usually embedded the domain knowledge and business rules in the programs which limited its reuse and is often difficult to maintain.

Our approach focusses on support the SMEs to understand what to do in each task list. Thus, the details of task list were made more comprehensible and explicit by breaking down into task characteristics. We made the details of task list more explicit by breaking down into task characteristics (TC). In addition, we identified related work product characteristics (WPC) and defined mapping rules between task characteristics and work product characteristics accordingly. The domain ontology was used as a conceptual agreement among assessors, consultants, system and database designers. The rule base knowledge primarily focuses on defining relations between characteristics of tasks and work products. The rating and recommendation engine can automatically generate scores and evaluate the standard based process implementation according to the proposed formulas. There are also recommendations to point the way for users to improve their processes. The process aimed to formalize and make explicit the assessment criteria that is implicit in the standard guideline. With this knowledge, software development capability can be assessed in terms of gap analysis and pre-assessment of readiness review before requesting the official assessment. The assessment result is rated based on the scores of work products, task lists, activities, and processes respectively.

A case study of developing a support system prototype for self-assessment of the ISO/IEC 29110 standard is presented. A preliminary evaluation results demonstrate significant improvements in terms of task coverage and time spent.

One limitation of the framework is that the user-specified task and work product characteristics are given the same priority. Thus, the weight of each item is not currently assessed. Therefore, our future work plan to incorporate weight-based analysis of task and work product characteristics in the assessment process.

Acknowledgments

This work is supported by National Electronics and Computer Technology Center (NECTEC) under National Science and Technology Development Agency (NSTDA), Ministry of Science and Technology, Thailand.

References

[1] D. Pajk, M. Indihar-Stemberger, and A. Kovacic, "Reference model design: An approach and its application," Proc. ITI 2012 34th Int. Conf. on Information Technology Interfaces, pp.455–460, 2012.
[2] D. Stojanovic, D. Slovic, I. Tomasevic, and B. Simeunovic, "Model for selection of business process improvement methodologies," 19th Toulon-Verona International Conference on Excellence in Services, pp.453–467, 2016.
[3] W. Emmerich and A. Finkelstein, "Software process improvement through standards," Tech. Rep., Dept. of Computer Science, University College London, UK, 1997.
[4] CMU/SEI, "CMU/SEI-2010-TR-033 CMMI for development, version 1.3: Improving processes for developing better products and services," 2010.
[5] ISACA, "ISO Standards: ISO 12207, ISO 15504 & ISO 9126," CETIC Meeting, May 23, 2007.
[6] IEEE, "ISO/IEC 12207: Systems and Software Engineering – Software Life Cycle Processes," Second Edition, 2008-02-01.
[7] "ISO/IEC 12207." [online]. Available: https://en.wikipedia.org/wiki/ISO/IEC_12207. [Updated: 17-Aug-2017].
[8] ISO, "ISO/IEC TR 29110-5-1-2 Software engineering - Lifecycle profiles for very small entities (VSEs) - Part 5-1-2: Management and engineering guide: generic profile group: basic profile," http://standards.iso.org/ittf/PubliclyAvailableStandards/c051153_ISO_IEC_29110-5-1-2_2011.zip
[9] "ISO/IEC 29110." [online]. Available: https://en.wikipedia.org/w/index.php?title=ISO_29110&oldid=752435619. [updated: 28-aug-2017].
[10] "ISO/IEC 29110: Software Devlopment Process Standard," in CIO

Training Project by Ministry of ICT, SIPA, and FTI, Thailand, 2013.
[11] A. Temprasirt, P. Sangpa, S. Suwanaroj, and T. Utayanaka, "Thailand Initial Implementation - ISO29110-VSE Model," 2018.
[12] C.Y. Laporte and R. O'Connor, "Software process improvement standards and guides for very small organizations - An overview of eight implementation," CrossTalk - The Journal of Defense Software Engineering, vol.3, no.3, pp.23–27, 2017.
[13] C.Y. Laporte, R. O'Connor, and G. Fanmuy, "International systems and software engineering standards for very small entities," Crosstalk - The Journal of Defense Software Engineering, vol.26, no.3, pp.28–33, 2013.
[14] C.Y. Laporte and R. O'Connor, "Software Engineering Standards and Guides for Very Small Entities: Implementation in two startups," 10th International Conference on Evolution of Novel Approaches to Software Engineering (ENASE 2015), 2015.
[15] L. Garcia, C.Y. Laporte, J. Arteaga, and M. Bruggmann, "Implementation and certification of ISO/IEC 29110 in an IT startup in Peru," Software Quality Professional Journal, vol.17, no.2, pp.16–29, 2015.
[16] ISACA, "COBIT 5: The framework exposure draft," 2011.
[17] R. O'connor, "Introduction to ISO/IEC software engineering standards," The Irish Software Engineering Research Centre (LERO), Dublin City University, Ireland.
[18] CMU/SEI, "CMU/SEI-2011-HB-001 standard CMMI appraisal method for process improvement (SCAMPISM) A, version 1.3: Method definition document," 2011.
[19] ISO, "ISO/IEC TC JTC1/SC7 WG1 N739 ISO/IEC 15504-5:2012 (E), Information technology - Part 5: An exemplar software life cycle process assessment model," 2012.
[20] J. Ivanyos, "Implementing COBIT based process assessment model for evaluating IT controls," 2009.
[21] ISACA, "ISACA's COBIT Assessment Programme (based on COBIT 5)," 2014.
[22] M. Pyhajarvi, "SPICE-International Standard for Software Process Assessment," 2004.
[23] "ISO/IEC 15504." [online]. Available: https://en.wikipedia.org/wiki/ISO/IEC_15504 (updated 17 June 2017).
[24] ISO, "ISO/IEC 33001:2015(E) Information technology - Process assessment - Concepts and terminology," 2015.
[25] ISO, "ISO/IEC 33002:2015(E) Information technology - Process assessment - Requirements for performing process assessment," 2015.
[26] ISO, "ISO/IEC 33003:2015(E) Information technology - Process assessment - Requirements for process measurement framework," 2015.
[27] ISO, "ISO/IEC 33004:2015(E) Information technology - Process assessment - Requirements for process reference, process assessment and maturity models," 2015.
[28] ISO, "ISO/IEC 33002:2015(E) Information technology - Process assessment - Process measurement framework for assessment of process capability," 2015.
[29] ISO, "ISO/IEC TR 33014:2013(E) Information technology - Process assessment - Guide for process improvement," 2013.
[30] C. Bach, "New ISO/IEC 33001 updates ISO/IEC 15504 series," 2015.
[31] T. Varkoi, "SPICE Update," Finnish Software Measurement Association (FISMA), 2014.
[32] A. Majchrowski, C. Ponsard, S. Saadaoui, J. Flamand, and J. Deprez, "Software Development Practices in Small Entities: an ISO29110-based Survey," EuroSPI 2014.
[33] "Deployment Package: Self-Assessment Systems Engineering Basic Profile," INCOSE 2013.
[34] C.Y. Laporte, "Deployment packages for the generic profile group for VSEs developing systems and/or software," available: https://profs.etsmtl.ca/claporte/English/VSE/VSE-packages.html
[35] J. Deprez, C. Ponsard, and D. Durieux, "Improving small-to-medium sized enterprise maturity in software development through the use of ISO29110," ERCIM News 99 - Special theme: Software Quality, pp.38–39, Oct. 2014.
[36] Y. Ayalew and K. Motlhala, "An ISO/IEC 15504 based software process assessment in small software companies," International Journal of Software Engineering and Its Applications, vol.8, no.6, pp.121–138, 2014.
[37] "Quest Tool Suite: ConCatenator & SPICE Master." [Online]. Available: http://www2.hms.org/cms/en/concatenator-master.html
[38] "SPICE 1-2-1 for International standard." [Online]. Available: http://www.spice121.com/cms/en/.
[39] "SPICE-Lite Assessment Tool." [Online]. Available: http://www.spicelite.com/cms/en/.
[40] T. Varkoi, "Deployment package: Self-assessment basic profile," Tech. Rep., Tampere University of Technology, Finland, 2009.

Apinporn Methawachananont received the B.Ed. degree in Mathematics and Science from Chulalongkorn University in 1988. She received the M.Sc. and Ph.D. Degrees in Computer Science from the University of New Haven and Illinois Institute of Technology, USA in 1994 and 1998, respectively. She is currently a senior researcher at the Standard and Testing Development Section (STDS) at NECTEC in Thailand.

Marut Buranarach received the B.Eng. degree in Engineering from King Mongkut's Institute of Technology Lardkrabang in 1995. He received the M.S. and Ph.D. degrees in Information Science from the University of Pittsburgh in 1998 and 2004, respectively. He iscurrently a senior researcher at the Language and Semantic Technology Lab at NECTEC in Thailand.

Pakaimart Amsuriya received her B.A. degree from Chulalongkorn University in 1992. She obtained her M.Sc. in Information Technology Management from Mahidol University in 1996. She is currently a senior engineer at the Standard and Testing Development Section (STDS) at NECTEC in Thailand.

Sompol Chaimongkhon received the B.Sc. degree in Medical Technology from Mahidol University in 1990. He received the M.Sc. degree in Operation Research from National Institute of Development Administration in 1993. He is currently a senior engineer at the Standard and Testing Development Section (STDS) at NECTEC in Thailand.

Kamthorn Krairaksa received the B.Sc. degree in Computer Science from Khon Kaen University, Thailand in 2000. He is a research assistant at National Electronics and Computer Technology Center (NECTEC), Thailand. His researches include Service Innovation and Cloud Technology.

Thepchai Supnithi received the B.S. degree in Mathematics from Chulalongkorn University in 1992. He received the M.S. and Ph.D. degrees in Engineering from the Osaka University in 1997 and 2001, respectively. He is currently the head of the Language and Semantic Technology Lab at NECTEC in Thailand.