Research Article GU J Sci, Part A, 10(4): 542-554 (2023) 10.54287/gujsa.1384048

Gazi University
Journal of Science
PART A: ENGINEERING AND INNOVATION
http://dergipark.org.tr/gujsa

Enhancing Software Process Assessment with an Ontology-Based Tool: Integrating CMMI, SPICE, and TMMI Models

Rukiye BAŞKARA[1*] Ahmet COŞKUNÇAY[2]

[1] Yildiz Technical University, İstanbul, Türkiye
[2] Atatürk University, Erzurum, Türkiye

Keywords

Software Process Reference Model

Process Assessment Tool

Semantic Web Tool

Software Process Improvement

Abstract

Software process reference models like Capability Maturity Model Integration (CMMI) and Software Process Improvement and Capability Determination (SPICE) have played a crucial role in evaluating processes for several decades. Their adoption by institutions has become essential for consistently delivering software projects to customers on time and with expected quality. However, conducting software process assessments demands considerable time, effort, and skilled personnel. This research introduces an ontology-based tool that facilitates software process assessment for organizations by integrating widely used process reference models, namely CMMI, SPICE, and Test Maturity Model Integration (TMMI). The development of this tool involves multiple stages: creating ontologies for each process reference model, integrating them into the tool, enabling querying capabilities, and visualizing the ontologies. Through a validation study in a selected organization, two sets of processes were assessed using the Ontology-Based Software Process Assessment Tool (OSPAT). The results demonstrated that organizations can benefit from OSPAT in evaluating their software development processes across diverse reference models, thereby enhancing overall process efficiency and quality.

Cite

Coşkunçay, A., & Başkara, R. (2023). Enhancing Software Process Assessment with an Ontology-Based Tool: Integrating CMMI, SPICE, and TMMI Models. *GU J Sci, Part A*, *10*(4), 542-554. doi:10.54287/gujsa.1384048

Author ID (ORCID Number)

0009-0007-5408-208X Rukiye BAŞKARA
0000-0002-7411-310X Ahmet COŞKUNÇAY

Article Process

Submission Date	31.10.2023
Revision Date	06.12.2023
Accepted Date	26.12.2023
Published Date	31.12.2023

1. INTRODUCTION

Software organizations have long relied on software process reference models like CMMI and SPICE to assess their existing processes. Through process assessment, institutions gain valuable insights into the strengths, weaknesses, and potential gaps in their software process implementations (15504, 2003). Furthermore, process reference models enable organizations to gauge the competence and maturity of their processes. By evaluating software project processes using internationally recognized frameworks, organizations can identify and address process shortcomings.

Failure to improve processes based on process assessments using reference models can lead to challenges in delivering software projects to customers as planned, with expected quality. In such cases, organizations may struggle to measure project efficiency, identify deficiencies, and monitor process compliance across multiple reference models, making it more difficult to understand deviations between them (Gazel et al., 2009). To streamline the assessment process with different models, there is a pressing need for an intuitive and comprehensive assessment tool.

The advent of the semantic web has empowered machines to comprehend the meaning of data and address associated problems using well-defined ontologies (Türkyılmaz & Yaşar, 2008). Ontologies present

*Corresponding Author, e-mail: rukiye.baskara@yildiz.edu.tr

promising solutions for computer-aided software process assessment (Feloni & Braga, 2015). As they are expressed in machine-understandable language, ontologies facilitate the identification and maintenance of mappings between process reference models and an organization's processes (Mora et al., 2010). Utilizing existing ontologies from the literature saves researchers time compared to creating ontologies from scratch.

However, despite some preliminary studies on CMMI, SPICE, and TMMI models, no tools in the literature support both ontology-based modeling and these specific process reference models. The primary goal of this article is to develop a comprehensive tool that facilitates the overall process assessment of organizations by unifying the CMMI, SPICE, and TMMI models into a common ontology. The use of a shared ontology allows for easy comparison of assessment results obtained from different models within the same organization. The ontologies created in this research will serve as a valuable resource for professionals in the field of software process assessment, aiding in understanding the models, identifying relationships between concepts, and discerning similarities and differences among the models. Besides these, other main contributions of the paper are to focus on the ability of the developed tool to support different process reference models by providing flexibility to organizations and to make process evaluations more effective by facilitating data access and increasing information sharing using semantic web technologies.

1.1. Literature Review

The results of assessments with software process reference models can be extensive and complex. In a study, they contributed to the development of 5 different software companies by measuring the CMMI maturity levels quickly and easily through a survey (Yucalar & Erdoğan, 2009). However, it was mentioned that this survey study is not suitable for large-scale companies. They also said that it does not claim to replace an official CMMI assessment. Therefore, there is a need to develop more reliable and effective methods or tools that measure the maturity levels of companies. In another study, the WMSC model for project management knowledge domains was presented, but it was found that there was a need to develop a generalizable expert system that would provide real-time dynamic solutions for maturity assessment (Göçmen, 2021). In addition, one study analysed the Smart Grid Maturity Model (SGMM), which is proposed as a maturity model in the field of smart grid. After the use of the SGMM, it has been revealed that there is a need for a model where practices are specialised according to the roles in the sector, can change according to different market structures, has clear scoring and criteria, can be easily used by everyone and has different maturity levels (İlisulu et al., 2020).

In addition to this, Due to the large number of process improvement models, observation, modification and validation of the models are time consuming and error prone. Therefore, although the SPI-CMMI ontology in the literature has many benefits for this purpose, it has been revealed that it needs to be further developed in terms of functions (Zaied et al., 2021). The need for tool support may be necessary to simplify, comprehend and analyze the results (Hunter et al., 1997). Such tools allow software development organizations to evaluate project processes quickly and accurately. A computer-assisted assessment tool is more cost-effective, easier to change and update, and can be used without requiring external consultants (Soydan & Kokar, 2006). Tools such as these are meant to help institutions reduce the costs associated with assessment and increase the reliability of assessment reports. As part of the process assessment, software tools were used to support the data collection phase and to monitor compliance with the software processes of the institutions (Gazel et al., 2012). Process assessment tools that support a single model are rarely sufficient to meet the needs of many organizations. By using tools that evaluate using different models, the results of the assessment can be compared. To put it another way, institutions may want to determine how they compare to different models by evaluating their processes against different process models. A software process assessment tool with diverse features should also enable the definition of a new model with a comprehensive meta-model (Yürüm, 2014). In one study, CMMI_Dev version 3.1 and PMBOK presented an ontological approach to represent, guide and analyse knowledge (Silega et al., 2023). However, they mentioned that there is a need to extend the ontology to represent the knowledge of various standards and to identify common concepts between different standards. They stated that the new methodology created by combining TMMI and PRISMA methodologies will increase the efficiency and effectiveness of the testing process in terms of time and budget (Aktaş et al., 2021).

The fact that most of the tools in the literature are non-ontology-based software process assessment tools indicates a deficiency in this area. Because, if making process assessments using an ontology-based tool is compared with doing it manually, assessments can be made in a shorter time with an ontology-based tool (Athiththan et al., 2018). In one study, they examined whether the ontological approach used significantly reduced the analysis time and found that the analysis of code defects with the ontological-based approach was 10 times faster than other methods (Castellano et al., 2023). By using ontology, the information about software development standards is represented. However, in order to increase its benefits, there is a need to represent more than one standard on this ontology and to define common concepts between different standards (Silega et al., 2023). In addition, since ontologies have computer-executable semantics, inferences can be made about the maturity level of models of companies based on some data provided by the institution (Soydan & Kokar, 2006). Among the related studies, the Ontology-based CMMI Mapping and Query Tool (OCMQT) is a software process assessment tool developed to support the data collection phase of process assessment and to observe its compliance with the CMMI model and is an add-on to an open-source EPF-based process management tool (Gazel et al., 2012). This tool supports the need to search for process definitions and CMMI-related information. However, the OCMQT tool is not capable of making inferences about the assessment rating. GSPA, on the other hand, is a meta-model process assessment tool created using class diagrams to integrate the most widely used process reference models, CMMI and ISO 15504, into a single model. Although this tool supports process assessment with two models, it is not an ontology-based tool (Yürüm, 2014). In one of the studies on the use of ontologies in different fields in the literature, the ontology was used in a supply chain assessment application because it has the advantages of easy extensibility and fast querying between different databases. However, in this study, it is mentioned that there is a need to work on some more adaptive tools based on ontology-based knowledge representation in order to make a more user-friendly application, access accurate information, increase its efficiency and process knowledge in modifiable format (Zhang et al., 2022). In another study, the necessity of using ontologies to make the information about Agile applications easier to understand by non-experts and people without prior knowledge was mentioned. Thus, an ontology-based user-friendly simplification tool called OBAMA was developed (Kiv et al., 2022). When the current process assessment tool studies are examined, it is observed that no ontology-based tool provides assessment and comparison with three different models. In this article, an ontology-based software process assessment tool is proposed by combining three different models in a common ontology. They developed the Software Integration Model (SIM) by combining the CMMI, IMM and SOVRM models to help organizations evaluate and improve their activities related to software integration in global software development (Haider et al., 2023). However, it has been observed that there should be a tool that can perform various activities and produce different evaluation reports for organizations.

According to the literature, the use of ontologies in various fields has increased in recent years. However, in order to solve the semantic, visual and analysis problems faced by organizations in software process evaluation, the need for a support tool that can perform evaluation with various models using ontologies has emerged.

2. MATERIAL AND METHOD

CMMI, SPICE, and TMMI, which are widely used software process reference models that play a crucial role in software quality assurance, have been examined as suitable for use in the process assessment tool. Considering the similarities between the process descriptions and applications of these three models, it was appropriate to match the three models. Due to their common features, these three models share similar concepts. The first step was to create ontologies for the three models. Ontologies were created based on questions about their purpose, which questions they would answer, and who would use them. Ontologies will be able to address and provide answers to the questions some of which are listed below. Defining and limiting the scope of ontologies is achieved through these questions.

- What are the purposes of the SPICE, TMMI, and CMMI models?
- What are the scopes and constraints of SPICE, TMMI, and CMMI models?
- What risk factors do SPICE, TMMI, and CMMI carry?

- What practices do SPICE, TMMI, and CMMI include?
- What work products do SPICE, TMMI, and CMMI suggest?
- What are the assessment methods used by SPICE, TMMI, and CMMI models?
- What are the roles of SPICE, TMMI, and CMMI models?
- What maturity/capability levels do SPICE, TMMI, and CMMI include?
- What types of projects are SPICE, TMMI, and CMMI used for?
- What are the process and process areas of the SPICE, TMMI, and CMMI models? To what goals do these processes, concepts, and relationships belong?

Table 1 illustrates how concepts belonging to Process reference models are matched (Vanamali et al., 2008). Despite the different names of some concepts in this table, it has been observed that they have a similar meaning. Moreover, since the TMMI model is derived from the CMMI model, it was easy to match as they share the same concepts. A concept found only in the CMMI model called Continuous Capability is absent in the TMMI model.

Table 1. *Matching the Concepts of the Models*

CMMI	SPICE	TMMI
Process Area	Process	Process Area
Purpose	Process Purpose	Purpose
Specific Goal	Process Outcomes	Specific Goal
Specific Practice	Base Practices	Specific Practice
Subpractices	-	Subpractices
Typical Work Products	Output Work Products	Typical Work Products
-	Work Product Characteristics	-
Generic Goal	Process Attributes	Generic Goal
Generic Practice	Generic Practices	Generic Practice
Generic Practice Elaborations	-	Generic Practice Elaborations
-	Generic Resources	-
Examples	-	Examples
Amplifications	-	Amplifications
Continuous Capability	-	-
Staged Capability	-	Staged Capability
Capability Levels	Capability Levels	Capability Levels

After creating the ontology of the three models, the common classes, and relationships of the 3 models were taken into account while creating the common YGOM ontology. While matching ontology, it is given the importance that the concepts of the models have the same function. In light of these questions, classes, subclasses, and relations of the models were created. In Figure 1, the conceptual scheme of YGOM drawn in the Draw.io [1]program is given.

In this ontology in Figure 1, the various concepts and the relationships between them denote the YGOM. Here are the concepts and relationships in this schema: ContinuousRepresentation and StagedRepresentation: These refer to the maturity levels of an organization. ContinuousRepresentation presents a continuous

[1] Draw.io: https://app.diagrams.net

development model, while StagedRepresentation focuses on specific stages or levels. Representation: This is a meta-concept that combines the above two concepts into a general category. Level and MaturityLevel: These represent different aspects of an organization's process maturity. Level refers to a general level and MaturityLevel refers to a specific level in terms of maturity. Organization: This represents the organization being assessed or to which the model is applied. Projects: Indicates the projects the organization is working on. Assessment: This represents the process of assessing the organization's processes ProcessArea: Refers to the different process areas within the organization. CapabilityLevel: Indicates the capability or maturity level of a process area. AssessmentOutput: Represents the assessment result, which is usually an output related to the capability level of a process. Goal, SpecificGoal, and GenericGoal: These concepts refer to the general and specific goals that the organization aims to achieve. Practice: Represents the methods or practices applied to achieve specific goals. The different links between relations (e.g. "isA", "hasA", "ownedBy", "consistOf") show how these concepts are interconnected and how they contribute to each other. For example, an Organization can be seen to be owned by Projects, which in turn are composed of ProcessAreas, and these ProcessAreas are themselves related to Goals and Practices. In Table 2, the concepts of the three models and the relationships between them are defined with their meanings in the conceptual schemes.

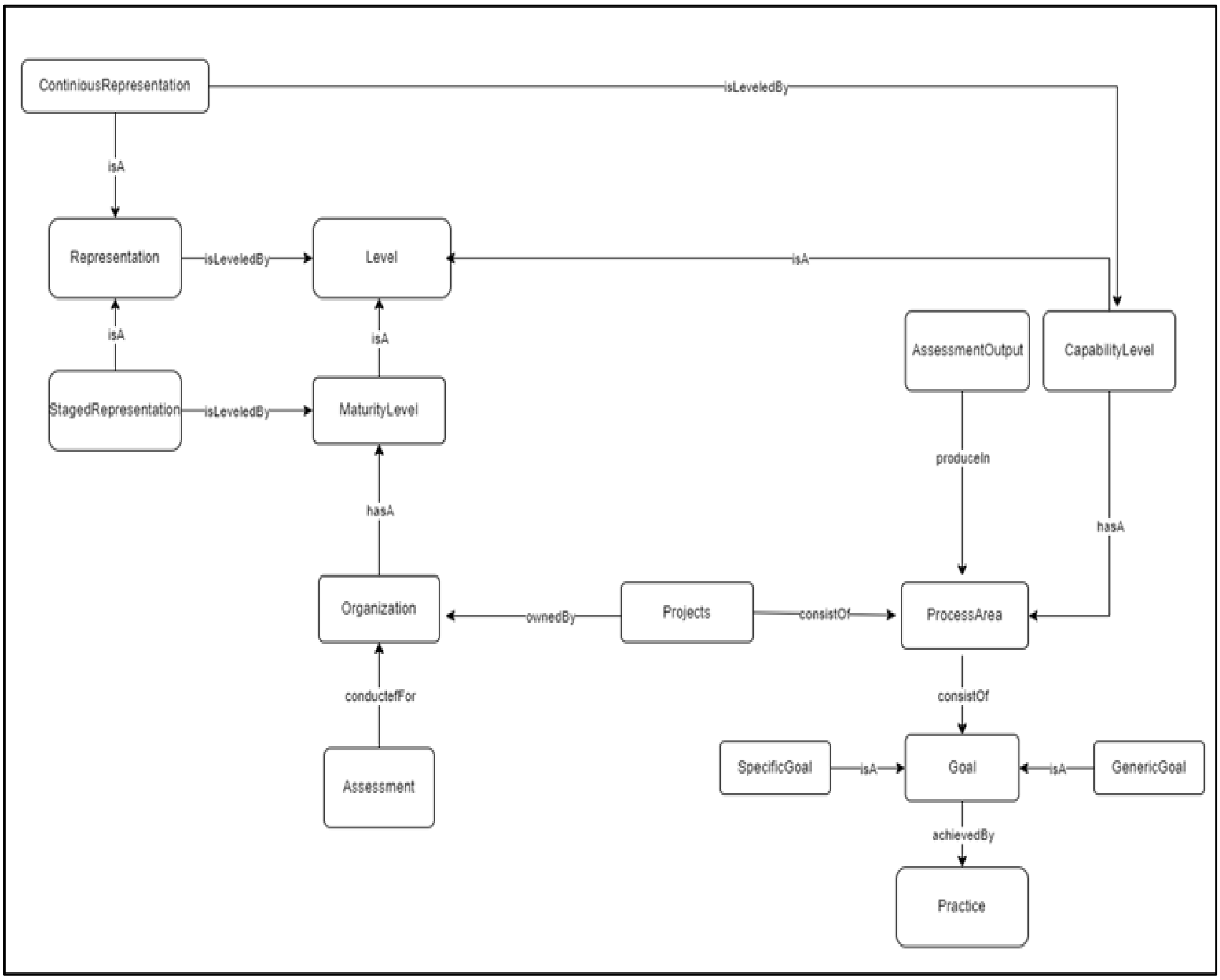

Figure 1. *YGOM Diagram*

Table 2. *Description of relationships in diagrams*

Relationships Between Concepts		**Example Relationships From Diagrams**
1	ownedBy	Organizations own the projects.
2	ConsistOf	Processes are included in projects.
3	ConductedFor	Assessments are made for organizations
4	isA	Generic practice is a Practice. It is used in superclass-subclass (Inheritance) relationships.
5	achievedBy	When all practices of a goal result in success, it is specified in the model it belongs to.
6	representedBy	Represented in the presentation from the process area
7	producedIn	Assessment output produced in the Process Area
8	isLeveledBy	Impressions rated by maturity level
9	satisfiedBy	Goals fulfilled by process areas
10	outputOf	Work products are the output of projects
11	developedBy	Progressive representation enhanced by Maturity Level
12	subClassOf	Process dimension is a subclass of Process.
13	fulfilledBy	Assessment Output produced in assessment
14	associatedWith	Work products associated with Process output
15	Uses	Generic resources are used by applications
16	hasLevel	Assessments have skill levels
17	sameAs	Artifacts are the same as work products

2.1. OSPAT (Ontology-Based Software Processes Assessment Tool)

2.1.1 Development of the Tool

Java was chosen as the programming language for the tool because it proved most suitable as several Java ontology libraries exist. Java is an open-source language with free libraries for developing semantic web applications, which has played a major role in its selection. The interface of the application was created using Java Swing, which is part of the Java AWT package and provides all the tools necessary to develop Graphical User Interfaces (GUIs) using Java. As the development platform, the open-source Eclipse[2] or Netbeans IDEs, which provide a Java-based application development environment, were suitable. Despite Netbeans' AWT and Swing graphical user interface support, the Eclipse environment has been preferred because it is more convenient to test the previously created frame codes. To integrate the ontology support, the Java framework Jena[3] Library was used. Jena was chosen because it provides full support for the JAVA language that we use in ontology-based development, because it can work in harmony with other components, and because it is an open source library for building Semantic Web applications. In this way, the information kept in the ontology was transferred to the Eclipse environment with the Jena Library. After the ontologies are loaded, SPARQL[4] , a standard rich query language similar to SQL, is used to access RDF files, that is, classes, properties, and individuals in ontologies through the interface. In Eclipse, Java codes were used to load the desired data from the imported ontology using SPARQL queries. These queries

[2] Eclipse: https://www.eclipse.org/
[3] Jena: https://jena.apache.org
[4] SPARQL: https://www.w3.org/TR/rdf-sparql-query/

transfer the information stored in the ontologies and present it over the interface. By using queries, information about process areas, applications, and work products can be retrieved from the OWL files of the models. The OWLGrEd[5] editor is used to visualize the ontologies. OWL files can be opened in this editor, providing the ability to visualize ontologies with OWLGrEd and distribute the formal information held in ontologies. The visualizer might make the tool easy to understand and easy to use, even for those who are "ontology illiterate".

2.1.2 Properties of the Tool

OSPAT supports assessments based on CMMI, SPICE, and TMMI models. The tool aims to ensure that the assessment results would be brought together in a common ontology. The ability to make process assessments with these different models contributes to the flexibility of the tool. OSPAT also can report different assessment results to the organizations by keeping and querying the assessment data. The fact that the tool is suitable for assessment across three models provides the opportunity to make comparisons by observing the assessment results of the three models. To compare the process assessments of the three models in the same organization, it is necessary to align the processes that have similar contexts whether they are named the same or not.

This ontology-based tool, which allows assessment with more than 25 process areas, contains the quality level of the organizations based on the processes assessed. This tool performs many functions such as entering grades that should be used during assessment for the models it supports, entering success percentages, and entering work products. In the interface of the tool, short explanations about the process areas in the CMMI, SPICE, and TMMI models and the practices in a process area are given for the convenience of the users. By using SPARQL, queried data is listed through the tool interface by using the ontologies, and a query function is added to the tool. With the queries information such as the process areas of the models, the applications of the process areas, and the work products of the applications are listed. For the sake of completeness, the concepts added through the interface are added to the OWL files connected with other concepts. It is also possible to read the ontologies created with the Protégé through the interface. Another feature of this tool is that it keeps data safe so that we can reuse the results during assessment and reporting and make comparisons with subsequent assessments. It is reliable in its ability to store assessments permanently. Finally, this tool has a simple and visual user interface to allow assessment by non-expert users. For process assessment tools quality and efficiency are important aspects (Lok & Walker, 1997). OSPAT is designed by conforming to Lok and Walker's (1997) quality characteristics.

2.1.3 Validation of the Tool

After the development of OSPAT, validation of it was carried out at a state university. This university has been found suitable for software process assessment-based validation due to the size of the software projects it has. An Ethics Committee approval was obtained to carry out the study. The university has an understanding of quality based on continuous improvement but has not yet documented itself with international standards or models in terms of its software development processes. First, a preliminary interview was held about the processes included in the university's software projects. Two groups of processes were selected for assessment. For these selected groups of processes, the common process areas of the three models are shown in Table 3. For matching the models and their processes, we referred to Vanamali et al.'s (2008) work. Software Requirements Analysis and Requirements Management processes do not have a corresponding process in the TMMI model, so no requirements related assessment is planned for TMMI as the blank cell in Table 3 indicates.

A software project was selected to make the assessment. Interviews were made with the System Analyst who worked on the selected project. First, demographic questions were asked in these interviews to gather information about the organization and the analyst participating in the assessment. After the demographic questions during the interview, questions about five selected process areas were asked and so the process

[5] OWLGrEd: http://owlgred.lumii.lv/

GU J Sci, Part A 10(4) 542-554 (2023) 10.54287/gujsa.1384048

assessment started. These questions are the regular assessment questions on the processes of the selected models (Kalaycı, 2007).

During the assessment, notes regarding the processes in the project were taken. Data such as the strengths and weaknesses of the application were entered into the tool by making inferences from the answers given to the application questions related to the selected process areas during the interview. The work product information is also added. After entering data, process achievements were evaluated. In Figure 2, the interface for process achievement evaluation, and work products belonging to the Software Requirements Analysis process, which is one of the SPICE process reference model processes, is given.

Figure 3 shows the interface where the assessment data can be added to the OWL file and the previously added assessment can be deleted after the information is entered regarding the Software Requirements Analysis Process during the assessment.

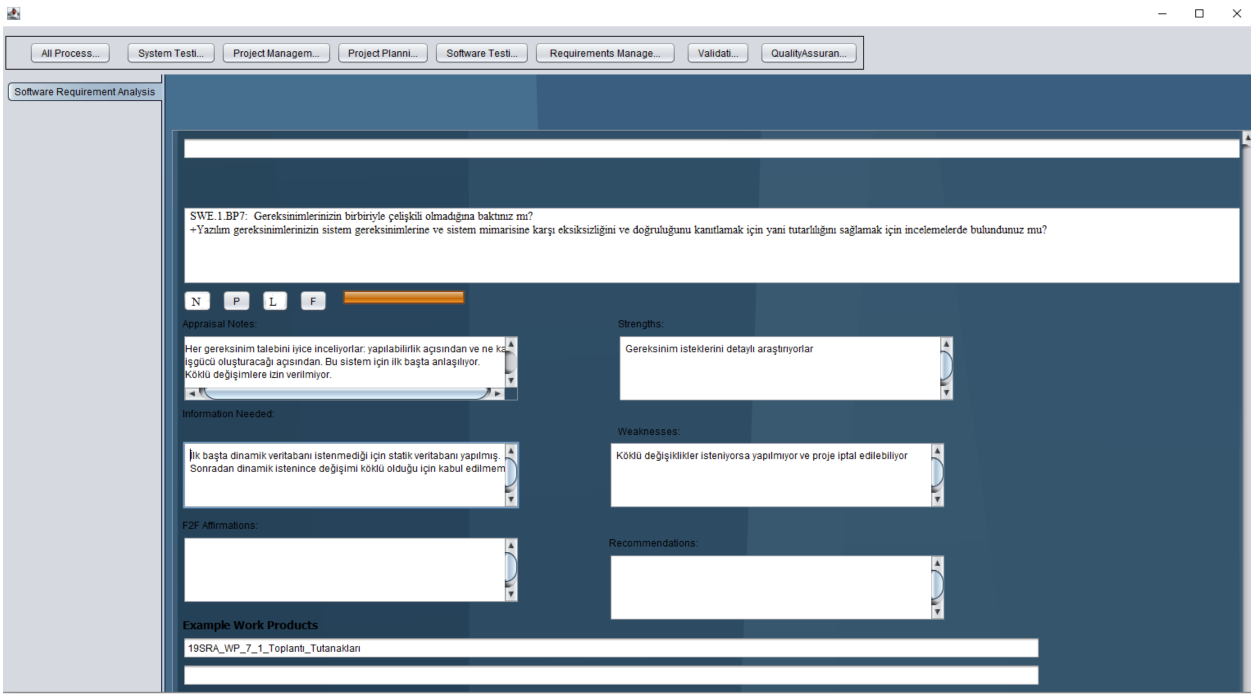

Figure 2. *The Interface for Assessing the Software Requirements Analysis Process*

Table 3. *The Matching Processes of the Models*

Groups Of Processes	SPICE	CMMI	TMMI
1st group of processes	Project Management	Project Planning	Test Planning
2nd group of processes	Software Requirements Analysis	Requirements Management	-

Using this interface, individuals and features can be added by entering data to the OWL file. The achievement rate of the Software Requirements Analysis process was 75%. The name of the assessment made in the university (i.e. Organization1) is Assessment1 and the name of the evaluated project is entered as Project1. The process work products were added as individuals through the interface. The achievement rate of the Requirements Management process is 70%. Based on the 1st group of process assessment, there is a 5% difference between the Requirements Management Process of the CMMI model and the Software Requirements Analysis Process of the SPICE model. This data shows that the assessment results of the two models are close to each other. According to the process reference models, both process areas of Organization1 achieve a good achievement rate. As a result of this assessment, the organization's process competence in the requirements management process is rated.

After the data added through the interface of the Software Requirements Analysis process is transferred to the OWL file, the work products and success percentage are displayed in Protégé as in Figure 5.

With the assessment of the processes in the 2nd group, the achievement rate of the Project Planning process was 76%. The achievement rate of the Project Management process was 78%. When the assessment of the Project Planning of the CMMI model and the Project Management process of the SPICE model are compared, there is a 2% difference between the two processes. This shows that the assessment results of the two models are very close to each other. Furthermore, both process areas of Organization1 achieve high achievement percentages based on the process reference models. The practices are applied sufficiently in these processes. Test planning, however, is assessed in the TMMI model with a 42% achievement rate. This low score shows that this organization has deficiencies in its practices in the Test Planning process area and that there is a need for improvement. The results of the achievement rates resulting from the validations are shown in Figure 4 for better comparison.

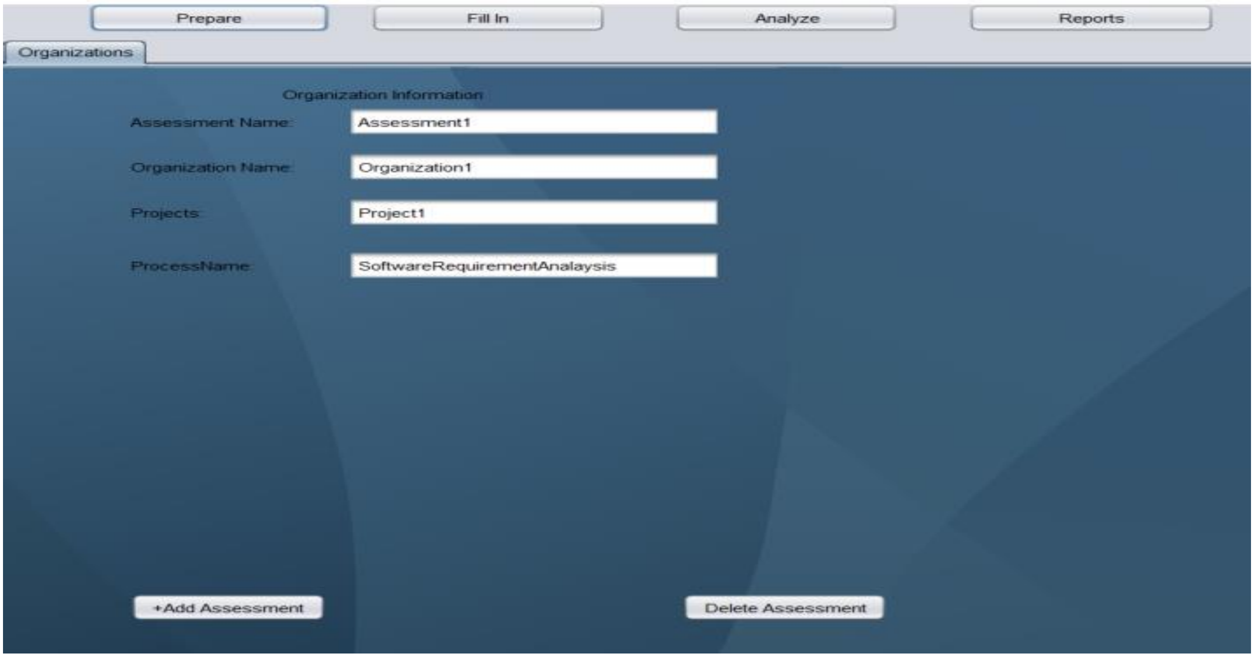

Figure 3. *The interface for entering and updating the assessment data kept in the ontology*

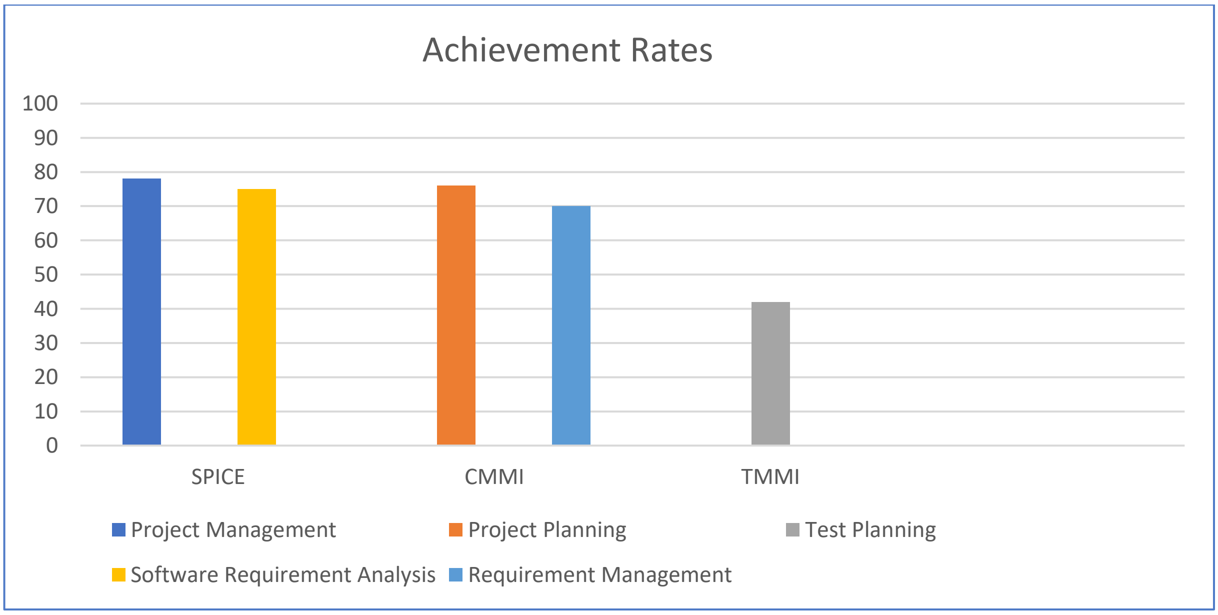

Figure 4. *Achievement Rates of Validation Results*

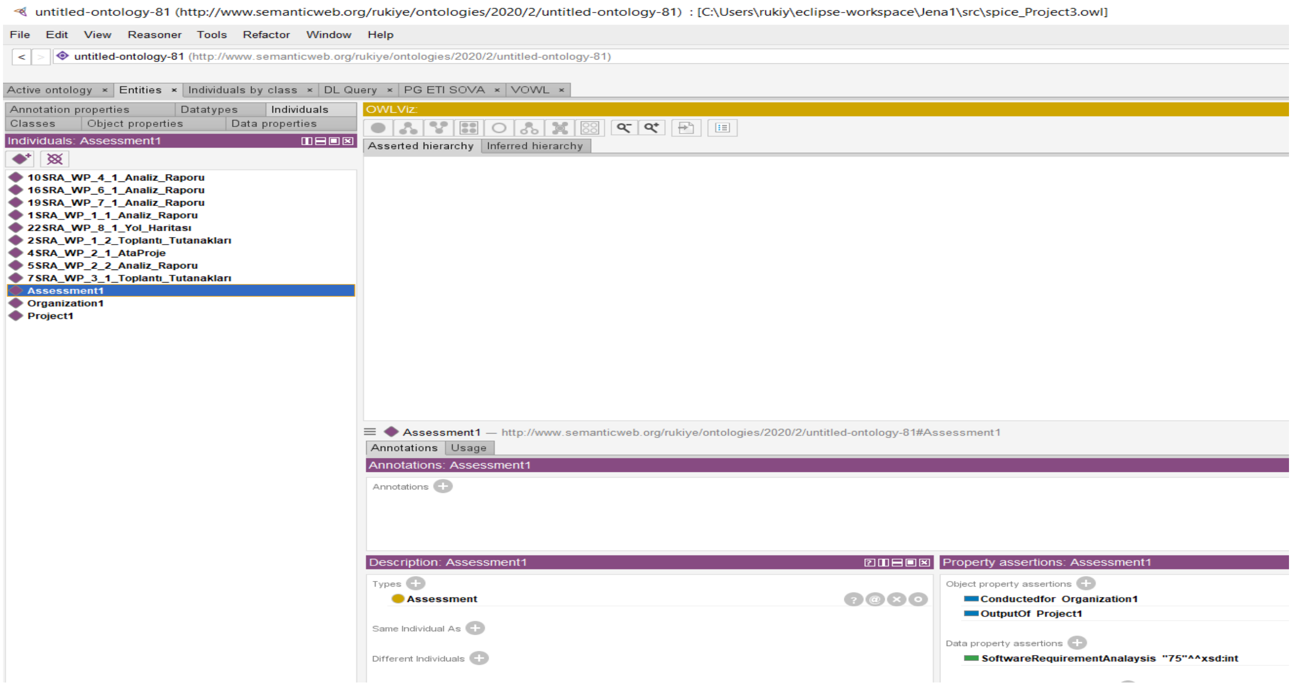

Figure 5. *Display of individuals added to the OWL file in the Protégé editor of the Software Requirements Analysis process after assessment*

Finally, with the OWLGrEd editor used, users are provided with the features to visualize the classes, properties, and individuals of the selected ontologies. The visualization is provided for CMMI, SPICE, TMMI, and YGOM ontologies and their instances associated with the assessments. In this way, instead of examining the complex and long OWL file, it is possible to see the entire ontology at once with the visualization editor, which is easy to understand and use even for those who are not "ontology literate". OWLGrEd provides visualizations that are similar to UML Class Diagrams by displaying OWL classes as UML classes, data properties as class attributes, object properties as associations, and individuals as objects.

For the objective evaluation of the interface of the OSPAT tool, an independent user was surveyed after performing a process evaluation. The questions in Table 4 were asked and answers were obtained (Charfi et al., 2015).

Table 4. *User Interface Survey Questions and Answers*

Number	User interface survey questions	Strongly agree	I agree	I don't know	Disagree	Strongly disagree
1	Did you realize what you wanted to do?		x			
2	Was the interface easy to use?	x				
3	Have you waited for something in the user interface and can't find it?				x	
4	Is the definition of each process area and application well defined?	x				
5	Did the tool guide you correctly in carrying out an action?		x			
6	Is it easy to read the content of the interface pages?	x				

3. RESULTS AND DISCUSSION

We have developed an advanced, ontology-based tool for general process assessment, combining three widely utilized models for evaluating software development processes within organizations. The integration of the model ontologies within the tool has resulted in a unified, common ontology, enabling seamless querying of assessment data and model information. Users benefit from a visual representation of the ontologies, facilitating comprehension, verification, and updates. Since ontologies are expressed in a formal, machine-intelligible, and shareable language (Gazel, 2009), the assessment information generated by the tool is equally precise and supported.

To validate the tool's effectiveness, a partial software process assessment was conducted using the three reference models to assess different groups of processes within a university project. The tool proves to be versatile, accommodating software process assessments across multiple reference models. Its ontology-based architecture ensures low effort requiring updates for incorporating new models or integrating changes into existing ones. During the evaluation process, the tool provided assessment results for each of the three process reference models, allowing the organization to compare its performance against each model individually. Additionally, by supporting assessments with multiple models, the tool facilitates comparisons of assessment results from different models within the same organization, enabling comprehensive process evaluation across various projects.

The OSPAT tool is designed for self-assessment purposes and, though comprehensive, lacks an automatic assessment feature. Nevertheless, it effectively fulfills its intended features, displaying a user-friendly design. However, there are opportunities for further enhancing its functionality. Notably, the tool presently lacks the ability to assess multiple processes simultaneously within a project. To overcome this limitation, introducing a parallel assessment capability would enable organizations with single or multiple assessment teams to evaluate multiple projects simultaneously. Such an improvement could significantly streamline and expedite the assessment process (Yürüm, 2014). Overall, the OSPAT tool demonstrates promising potential in aiding organizations with their process assessment needs and can be further refined to better accommodate complex project evaluations.

4. CONCLUSION

This paper presents the development of a new ontology-based software process evaluation tool called OSPAT. The tool offers the capability to perform evaluation based on three process reference models, and to validate the effectiveness of the tool, a partial software process evaluation was performed on a university software project, whose process achievement rates are shown in Figure 4. As an ontology-based tool, OSPAT was conceived in response to the lack of a process assessment support tools capable of incorporating various reference models with the ontology models it uses.

By leveraging OSPAT, users can retrieve queries results from the ontologies of three process reference models and conduct assessments in a unified manner. The tool's ability to display all three models on a single platform facilitates comparisons of assessments made on the same project with different models. In order to objectively evaluate the interface of the OSPAT tool, a survey was conducted on an independent real user. In line with the answers given, the usefulness of the tool was also verified. The visual representation of ontologies, including classes, subclasses, relations, properties and individuals, was facilitated using the OWLGrEd editor and helped to support the decision-making process for people using the tool by increasing clarity.

The tool demonstrated the possibility of integrating three models into a single ontology and highlighted similarities among the matched process areas. Moreover, it provided a glimpse into the potential of unifying diverse reference models within a shared ontology, promoting open-source accessibility and dissemination of valuable model information to interested experts.

Organizations can benefit from this tool as it facilitates the comparison of results from regular process assessments, enabling them to track their progress in software process assessments. Ultimately, OSPAT presents a promising advancement in ontology-based software process assessment, streamlining assessments across multiple models and fostering a more efficient and insightful assessment process for software projects. Furthermore, the ontology used makes it easier to model complex relationships between software standards, process areas and other related concepts. This allows the tool to perform more comprehensive analyses and evaluate different aspects of projects in more depth. With this ontology, a flexible and extensible software evaluation tool has been developed. This means that the information in the tool can be updated when new concepts arise during the evaluation process.

5. FUTURE WORK

In this study, validation was conducted within a single organization, allowing for the evaluation of processes across different projects. However, due to time constraints, verifications on multiple projects were not feasible. To enhance the tool's validation, it is planned to extend assessments to multiple projects and involve diverse organizations. Additionally, the tool can be expanded by incorporating assessment features for other software process reference models beyond the three selected in this study.

Notably, the OSPAT tool is well-suited to support process assessments based on SPICE, CMMI, and TMMI as these reference models share similar concepts. In the current research, ontologies were developed based on the v1.3 version of the CMMI process reference model. However, to keep the tool up-to-date, future studies can integrate changes from the CMMI v2.0 version into the ontologies.

By extending validation efforts to multiple organizations and incorporating updates from newer versions of relevant process reference models, the OSPAT tool can continue to evolve and remain a valuable asset for efficient and comprehensive software process assessments across various projects and organizations.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

15504, A. I. I. (2003). Information Technology–Process Assessment.

Aktaş, A. Z., Yağdereli, E., & Serdaroğlu, D. (2021). An introduction to software testing methodologies. *Gazi University Journal of Science Part A: Engineering and Innovation*, *8*(1), 1-15.

Athiththan, K., Rovinsan, S., Sathveegan, S., Gunasekaran, N., Gunawardena, K. S., & Kasthurirathna, D. (2018, December). *An ontology-based approach to automate the software development process*. In: Proceedings of the 2018 IEEE International Conference on Information and Automation for Sustainability (ICIAFS) (pp. 1-6). IEEE.

Castellano, I. L., Aguilar, G. F. C., Silega, N., Kamal, T., Al-Gaashani, M., Samee, N. A., & Alabdulhafith, M. (2023). An Ontology-Based Approach to Reduce the Negative Impact of Code Smells in Software Development Projects. *IEEE Access*, *11*, 100146-100153 https://www.doi.org/10.1109/ACCESS.2023.3300575

Charfi, S., Ezzedine, H., & Kolski, C. (2015). RITA: a useR Interface evaluaTion frAmework. *Journal of Universal Computer Science*, *21*(4), 526-560.

Feloni, D., & Braga, R. T. V. (2015). Methodologies for Evaluation and Improvement of Software Processes in the Context of Quality and Maturity Models: a Systematic Mapping. In CIbSE (p. 123).

Gazel, S. (2009). *"CMMI-DEV" için bir ontoloji ve CMMI esaslı süreç değerlendirmenin ontoloji tabanlı bir araç ile desteklenmesi*. MSc Thesis, Hacettepe University.

Gazel, S., Sezer, E. A., & Tarhan, A. (2012). An ontology based infrastructure to support CMMI-based software process assessment. *Gazi University Journal of Science*, *25*(1), 155-164.

Gazel, S., Tarhan, A., & Sezer, E. (2009). A CMMI ontology for an ontology-based software process assessment tool, Proceedings of the 16th EuroSPI2 Conference.

Göçmen, E. (2021). A maturity model for assessing sustainable project management knowledge areas: A case study within a logistics firm. *Journal of Advanced Research in Natural and Applied Sciences*, *7*(4), 536-555.

Haider, W., Ilyas M., Khalid S., & Ali, S. (2023). Factors influencing sustainability aspects in crowdsourced software development: A systematic literature review. *Journal of Software: Evolution and Process*, e2630. https://www.doi.org/10.1002/smr.2630

Hunter, R., Robinson, G., & Woodman, I. (1997). Tool support for software process assessment and improvement. *Software Process: Improvement and Practice*, *3*(4), 213-223.

İlisulu, F., Kolukısa, A., & Kavak, K. (2020). Akıllı Şebeke Olgunluk Modelinin Dünya Çapındaki Uyarlamaları ve Türkiye için Öneriler. *Bilişim Teknolojileri Dergisi*, *13*(2), 123-136. https://www.doi.org/10.17671/gazibtd.533221

Kalaycı, O. (2007). Yöneticiler için Doğru Sorular CMMI. Shamrock Süreç İyileştirme ve Yenilikçilik, Kanada.

Kiv, S., Wautelet, Y., Heng, S., & Kolp, M. (2022). OBAMA, an Ontology-Based Software Tool for Agile Method Adoption. arXiv preprint arXiv:2206.02207.

Lok, R. H., & Walker, A. (1997). *Automated tool support for an emerging international software process assessment standard*. In: Proceedings of the IEEE International Symposium on Software Engineering Standards. IEEE, (pp. 25-35).

Mora, M., Cervantes-Pérez, F., Garrido, L., Wang, F., & Angel-Sicilia, M. (2010). On KM, KMS and ontology-based KMS in the domain of SwE standards and models of processes: A conceptual survey. *Intelligent Decision Technologies*, *4*(1), 75-98.

Silega, N., Castro Aguilar, G. F., Martillo Alcívar, I., Faggioni, K. M., Rogozov, Y. I. ,& Lapshin, V. S. (2023). An ontology-based approach to support the knowledge management of software quality standards. *Enfoque UTE*, *14*(3), 49-56. https://www.doi.org/10.29019/enfoqueute.946

Soydan, G. H., & Kokar, M. (2006). An OWL ontology for representing the CMMI-SW model, Workshop on Semantic Web Enabled Software Engineering (SWESE). Citeseer.

Türkyılmaz, İ., & Yaşar, C. (2008). Semantik web teknolojileri. Akademik Bilişim 2008.

Vanamali, B., Bella, F., & Hörmann, K. (2008). From CMMI to SPICE-Experiences on How to Survive a SPICE Assessment Having Already Implemented CMMI, 2008 32nd Annual IEEE International Computer Software and Applications Conference. IEEE, pp. 1045-1052.

Yucalar, F., & Erdoğan, Ş. Z. (2009). A questionnaire based method for CMMI level 2 maturity assessment. *Journal of Aeronautics and Space Technologies*, *4*(2), 39-46.

Yürüm, O. R. (2014). *GSPA: A generic software process assessment tool*. MSc Thesis, Middle East Technical University.

Zaied, A. N., H., Eldrandaly, K., & Tantawy, A. (2021). An OWL-Based Ontology Structure for representing Multimodel Process Improvement Framework. *Kafrelsheikh Journal of Information Sciences*, *2*(1), 1-11. https://doi.org/10.21608/kjis.2021.22087.1007

Zhang, L., Olsen, A., & Lobov, A. (2022). An ontology-based KBE application for supply chain sustainability assessment. *Resources, Environment and Sustainability*, *10*, 100086.