A Systematic Approach for Evaluating BPM Systems: Case Studies on Open Source and Proprietary Tools

Andrea Delgado, Daniel Calegari, Pablo Milanese, Renatta Falcon, Esteban García

HAL Id: hal-01320162

https://inria.hal.science/hal-01320162

Submitted on 23 May 2016

A systematic approach for evaluating BPM Systems: case studies on Open Source and Proprietary tools

Andrea Delgado, Daniel Calegari, Pablo Milanese, Renatta Falcon, Esteban García

Instituto de Computación, Facultad de Ingeniería, Universidad de la República,
Montevideo, Uruguay
{adelgado, dcalegar, pmilanese}@fing.edu.uy
{renafal, estebangrd}@gmail.com

Abstract. Business Process Management Systems (BPMS) provide support for modeling, developing, deploying, executing and evaluating business processes in an organization. Selecting a BPMS is not a trivial task, not only due to the many existing alternatives, both in the open source and proprietary realms, but also because it requires a thorough evaluation of its capabilities, contextualizing them in the organizational environment in which they will be used. In this paper we present a methodology to guide the systematic evaluation of BPMS that takes into account the specific needs of each organization. It provides a list of key characteristics of BPMS which are ranked by the organization and evaluated using test cases and quantitative criteria. We also present case studies of open source and proprietary BPMS evaluations following our proposal.

Keywords: Business Process Management Systems (BPMS), Open Source and Proprietary BPMS, evaluation methodology, systematic approach

1   Introduction

Every organization executes daily operations to achieve its goals, applying certain mechanisms to enable continuous improvement. The business process (BPs) vision is the identification of the set of activities that are performed in coordination within an organizational and technical environment to achieve defined business goals [1]. It provides support for the definition, control and continuous improvement of business operation. In this context, Business Process Management (BPM) [1,2] offers a framework to support the business process lifecycle [1] from modeling, through developing, deploying, executing and to the evaluation of their execution. BPM Systems (Business Process Management System, BPMS) [1,3] arise as the technology response to support the BPs lifecycle. These platforms integrate several components that allow modeling processes, executing them, controlling business rules, defining execution measures and monitoring processes, among others.

There are several process modeling and execution languages with different backgrounds and abilities, such as Business Process Model and Notation (BPMN 2.0) [4], XML Process Definition (XPDL) [5] and Web Services Business Process Execution

Language (WS-BPEL) [6], among others. Likewise, there is also a wide variety of BPMS, both open source and proprietary, with different support levels for the defined solution. In addition, several open source products offers several business models such as community edition with limited functionality, fees for maintenance and support, enterprise versions, among others. To be able to compare features within different BPMS, it is necessary to provide an objective evaluation regarding the fulfillment of key technical features that should be provided, as defined in academia [7,8] and industry [9,10] studies. However, the selection of the most adequate BPMS for an organization depends not only on the technological support it provides, but also on the characteristics of the organization itself. The evaluation should also be guided by a systematic procedure to ensure the quality of the results and its repeatability.

In this paper we present a methodology for evaluating BPMS considering the specific needs of each organization. Our approach includes the definition of key activities to guide the evaluation and a list of key features that are relevant to this kind of systems. This methodology has been developed within our research group and has been applied for evaluating open source and proprietary BPMS in several projects[1]. To illustrate the approach, we present results from these projects which constitute both a validation and assessment of our proposal and a contribution to knowledge regarding the capacities of selected BPMS technologies.

The rest of the article is organized as follows. In Section 2 we discuss related work and in Section 3 we present the methodology for evaluating BPMS. Then, in Section 4 we present case studies regarding the evaluation of open source and proprietary BPMS. Finally, in Section 5 we present some conclusions and future work.

2   Related Work

There are several approaches for evaluating BPMS, which we have analyzed and taken into account when defining our methodology and the list of characteristics we provide. We have taken into account software quality characteristics standards such as ISO/IEC 9126 (superseded by SQUARE [11]), and others such as [12]. Although the characteristics defined in these are not specific to BPMS, some can be applied to software of any kind and are very important for evaluating the quality of software from different points of view. A key reference from academy for evaluating BPs languages capacities and BPMS support for modeling and execution is the workflow patterns [7], used for example in [13] to evaluate the support provided by selected open source tools. This kind of assessments can identify potential limitations for modeling and execution of BPs in the selected languages and/or BPMS that is better knowing in advance. Other works such as [8] evaluate selected open source workflow engines against their compliant to the WfMC model, defining key characteristics for them. Other evaluations are contemporary to our work, such as [14] in which WS-BPEL engines; both open source and proprietary are evaluated. They differ from ours in many aspects a key one is that ours is more generic and allows many types of en-

_____
[1] http://www.fing.edu.uy/inco/investigacion/grupos/COAL

gines to be evaluated and compared, since it is not restricted to a single language. Other proposals are more generic for the selection of any type of COTS software such as [15,16] or with focus on specific characteristics of OSS software such as [17]. In [18] the authors present a survey on processes for selecting COTS. Although our proposal shares some similarities with these works regarding the process and some general desirable characteristics, our focus is on BPMS for which our list of characteristics provides a unique insight of this type of systems. Moreover, we provide several test cases and guides to evaluate the characteristics we defined.

Regarding industry reports of evaluation of proprietary BPMS, we have mainly considered Gartner [9] and TEC [10] reports, and also checked out the Forrester [19] approach. Gartner evaluations we have analyzed include the Magic Quadrant for BPMS [20] and Magic Quadrant for iBPMS [21], the latter adding elements of Business Intelligence. The criteria used by Gartner include commercial characteristics, such as: price, customer experience, market understanding and strategy, business model, among others. TEC provides software features for evaluating different types of software, in a web application with a list of defined and categorized characteristics, obtaining a recommendation of selected tools that best suits the indications provided. Forrester [19] also provides a similar approach with a set of characteristics and a tool to support the recommendation. However, these approaches are based on information provided by vendors who answer a questionnaire valuating each characteristic (Forrester also includes laboratory evaluation). In [20,21] the results are not context-dependent since the importance of the characteristics is given and not selected for each evaluation. Unlike these works, our approach does not include any view from the vendors themselves, but from our objective evaluation from carrying out each test case for each defined characteristic on each selected BPMS. The importance of the characteristics is also assigned each time by each organization, guaranteeing that the results are adequate for each organization every time.

3   BPMS EVALUATION APPROACH

The main results of our work are the list of characteristics which can be used as a basis for evaluating BPMS, and the methodology to carry out the evaluation.

3.1   List of characteristics

The list is organized according to a defined structure which groups characteristics allowing an intuitive understanding. The highest level corresponds to the modules, which in turn are composed of categories grouping cohesive characteristics. We have analyzed and selected characteristics based on many sources, as presented in Section 2. Two main modules are defined: (1) Technical, which involves everything related to software itself, and (2) Non-technical, which encompasses other characteristics such as community support. Table 1 shows the defined structure including both modules and its categories, the total of characteristics defined within each one (# DC) and, due to space limitations, an example of characteristics in each category.

Table 1. Structure and example of defined modules, categories and characteristics

Module	Category	# DC	Example of characteristics
Technical	Technology, Architecture and Interoperability	15	• BPMS Architecture • Integration with social networks
	Process Design and Modeling	12	• Modeler type • List of versions of process models • Collaborative work on processes
	Form management	9	• Dynamic Forms • Support for mobile devices
	Workflow engine	24	• Support of workflow patterns • Configurable Schedule of full System • Linking task and document
	Security Management	5	• Permission Mechanism for users • Role Definition
	Management, Monitoring and Audit	9	• Process Monitoring • Backups
	Document management system	5	• Integrity and document security • Indexing and search mechanisms
	Portal	7	• Customizing the portal • Searching Mechanisms
Sub-Total		79	
Non-Technical	Installation and support	8	• Installation packages • Available documentation • Supported Languages
	Maturity	6	• Time in market • Community activity
	Commercial	1	• License costs
Sub-Total		15	
Total		94	

3.2 Evaluation Methodology

Fig. 1 models the proposed evaluation methodology using BPMN, showing the different activities to be carried out within each organization, including the sub-process of actually evaluating the tools. In the first place the list of characteristics is reviewed and updated if needed and the tools to be evaluated in the current evaluation are selected, based on initial criteria such as being open source or proprietary, the language provided, among others defined to help narrow the selection. Then, each characteristic is weighted by the organization using a scale we provide, both to define how important is each characteristic to the organization and to obtain a ranked list to select the most important characteristics to be evaluated (as evaluating all of them can be expensive and time consuming). Then the test cases to evaluate the selected characteristics are defined (or adapted if needed, as we provide many test cases) and the case study to be carried out within each tool is specified (as we also provide many case studies). Then, the evaluation is performed valuating each characteristic within each tool in another scale we provide for results (using the test cases for each one and the case study to assess the tool globally). Finally a total score for each tool is calculated.
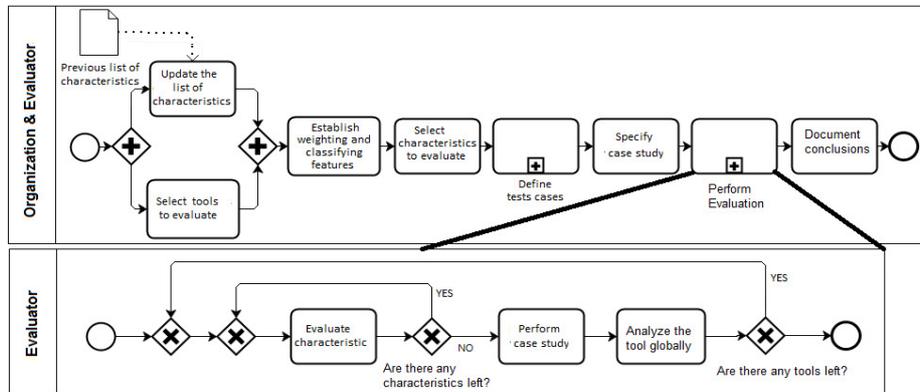
Fig. 1. Evaluation methodology process modeled in BPMN

The list of defined characteristics for BPMS tools is reviewed and updated for each evaluation, when needed. This update could involve the addition of new characteristics, modifying existing ones and/or deleting other, allowing working with a suitable list of characteristics at the time the evaluation is made. In parallel with this, the tools to be evaluated are selected (from those lists provided by organizations such as OMG, OASIS and WfMC), based on initial criteria defined such as open source or proprietary, presence in a specific market, or support for a defined modeling notation.

Each characteristic is then assigned a level of importance by the organization performing the evaluation. The scale defines the following levels: (1) Mandatory; (2) Medium priority and (3) Low priority. The classification of the characteristics on this scale depends on the needs of the organization for each evaluation and therefore allows to instantiate the evaluation to the organizational context. Each characteristic is further valuated in a three level scale of support: (1) Totally supported, the tool has the characteristic; (2) Partially supported, the tool does not cover the entire specification of the characteristic; (3) Not supported, the tool does not provide it. Additionally three levels of compliance are defined for the support scale: (1) Native, the feature is part of the tool; (2) Particularization, specific software can be developed to achieve such compliance; (3) Integration, it is necessary to include a third component to support it. In order to obtain a quantitative evaluation (score) associated with each tool, we also calculate a final value regarding the importance defined and the results level. Moreover, when two tools present different levels with respect to the same support scale, e.g. third components are needed but with different development costs, we can assign different values. The list of ranked characteristics regarding their importance as assigned by the organization is reviewed to select the key ones to be used in the current evaluation, to help reduce the number of characteristics to be evaluated. Nevertheless all the characteristics could be selected if the organization wants, taking into account that it will require more time and effort. Since the list of characteristics provide a shared criterion for evaluating BPMS, previous results of evaluations can be used as a basis for carrying out an organization-dependent evaluation process.

Two ways for evaluating the characteristics are defined: theoretical and practical. The theoretical evaluation does not require executing the tool, but is mainly based on the tool documentation, e.g. when non full versions are available or when characteristics are not a priority for the organization. The practical evaluation do requires executing the tool, with a specific test case to evaluate the level of support it provides. The main purpose of the test cases is to standardize the evaluation with respect to the same basis. In addition, a case study is defined to be performed within all tools. The main objective of the case study is to give an overview of the support provided by the tool regarding the actual execution in a daily basis operation.

Finally a global analysis of each tool is performed, including the score assigned by the defined formula. This allows performing a comparison between tools based on each characteristic evaluation result and the overall score assigned to each tool.

4 BPMS evaluation case studies

In this section we present the results of evaluation projects of open source and proprietary BPMS we have carried out between years 2010 and 2013 as a validation and assessment of our proposal. We have followed the guidelines in [22] for cases studies, but due to space limitations we present here only the results discussion.

4.1 Open source BPMS evaluation

The open source BPMS assessment was carried out in two projects, one focusing on BPMS based on the XPDL standard and another on the WS-BPEL standard. Two more tools were also selected which implemented the new BPMN 2.0 standard which was released in the course of the evaluation projects. The following tools were selected and evaluated mainly based on their availability:

- XPDL: Bonita CE[2], Enhydra[3], Joget[4], OBE[5], WfMOpen[6]
- WS-BPEL: Apache ODE[7], jBPM[8], Orchestra[9], Petals[10], Intalio CE[11], Riftsaw[12]
- BPMN 2.0: Activiti[13], jBPM5[8]

---

[2] BonitaSoft BPMS, http://www.bonitasoft.com/

[3] Together Enhydra Shark, http://www.together.at/prod/workflow

[4] Joget, http://www.joget.org/

[5] OBE, Open Business Engine, http://obe.sourceforge.net/

[6] WfmOpen, http://wfmopen.sourceforge.net/

[7] Apache ODE, http://ode.apache.org/

[8] jBPM, jBPM5, http://www.jboss.org/jbpm/

[9] Orchestra, http://orchestra.ow2.org/

[10] Petals, http://petals.ow2.org/

[11] Intalio BPMS, http://www.intalio.com/bpms

[12] Riftsaw, http://www.jboss.org/riftsaw

[13] Activiti BPMS, http://www.activiti.org

Fig. 2 shows an example of the results for the evaluation of some selected characteristics for XPDL and BPMN 2.0 (Activiti) and WS-BPEL and BPMN 2.0 (jBPM5) engines, using the semaphore metaphor: Green for Totally supported, Yellow for Partially supported and Red to indicate Not supported.

(a) XPDL and BPMN 2.0	Enhydra	Joget	Bonita	OBE	WFMOpen	Activiti
Flow routing	🟡	🟡	🟡	🟢	🟡	🟢
Process Designer	🟢	🟡	🟢	🔴	🔴	🟢
Calendar	🔴	🔴	🔴	🔴	🔴	🔴
Business Rules	🟢	🟢	🟢	🟢	🟢	🟢
Notifications and Alerts	🟡	🟡	🟢	🟢	🟡	🟢
Tasks Allocation by Roles	🟢	🟢	🟢	🔴	🟡	🟢
Work lists	🟢	🟢	🟢	🟢	🟡	🟢
Task assignments	🟡	🟡	🟢	🔴	🟡	🟡
Process and/or Activity Monitoring	🟢	🟢	🟢	🟢	🟡	🟡

(b) WS-BPEL and BPMN 2.0	jBPM BPEL	Apache ODE	Intalio	Riftsaw	Petals	Orchestra	jBPM5
Easy of installation	🟡	🟡	🟢	🟡	🟢	🟡	🟢
Deploy assistance	🟢	🟢	🟢	🔴	🟢	🟢	🟢
Execution of multiple versions of a process	🟢	🟢	🟢	🟡	🔴	🟡	🟡
Suspend or rollback a process	🔴	🔴	🟡	🔴	🟡	🟡	🔴
Audit	🟡	🔴	🟢	🟡	🔴	🔴	🔴
Reports	🔴	🔴	🔴	🔴	🟡	🔴	🟢
Integration with different Data Bases	🟢	🟢	🟢	🟢	🟢	🟢	🟢
Obtaining execution indicators	🟢	🟡	🟢	🟡	🟡	🔴	🟡

Fig. 2. Example of evaluation results: (a) XPDL and BPMN 2.0; (b) WS-BPEL and BPMN 2.0

Fig. 3 shows the overall scores obtained by each tool in each evaluation, by means of the formula defined (c.f. Section 3.2). As mentioned above, to obtain those values the importance assigned to each characteristic by each organization performing the evaluation is also taken into account, to weight the most important ones. This final score serves mainly for the purpose of "eliminating" tools which does not reach certain level, focusing on the ones presenting the best scores.
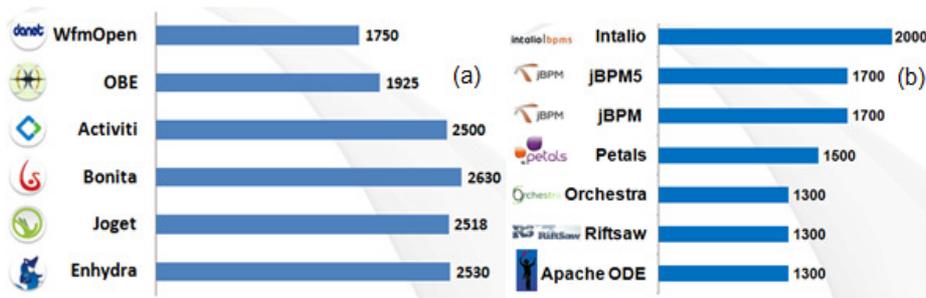

(a)		(b)	
WfmOpen	1750	Intalio	2000
OBE	1925	jBPM5	1700
Activiti	2500	jBPM	1700
Bonita	2630	Petals	1500
Joget	2518	Orchestra	1300
Enhydra	2530	Riftsaw	1300
		Apache ODE	1300

Fig. 3. Overall scores for evaluations: a) XPLD and BPMN 2.0; (b) WS-BPEL and BPMN 2.0

As conclusions some interesting observations were found: in the first place the version of the XPDL standard of the engine plays an important role in their capabilities, as it has several versions, 1.0, 2.0 and 2.1 with different features. Four of the six engines implemented version 1.0 which has significant limitations. Versions 2.0 and 2.1

already have elements of the BPMN standard and therefore better capabilities. Some of the engines had a high complexity of installation at the time, as WfMOpen and OBE. Bonita presents support for most features, including process simulation, being one of the most complete engines. Enhydra and Joget for XPDL and Activiti for BPMN 2.0 also provide support for most of the characteristics evaluated. Being the Activiti an initial version of the BPMN 2.0 standard implementation, it was expected to improve considerably, which has already occurred in recent years. As for the execution language WS-BPEL all evaluated engines implement the WS-BPEL 2.0 standard and some like Intalio CE, include extensions to provide support for humans such as HumanTask or BPEL4People. Intalio CE is the engine that provides better support for the evaluated characteristics and better results throughout the evaluation, with a friendly environment, stable behavior and an active community. Other engines such as jBPM, jBPM5 and Petals, also provide support for most characteristics.

4.2   Proprietary BPMS evaluation

The evaluation on proprietary tools also included a reevaluation of the Bonita open source BPMS since a major update was performed in the newly released 6.0 version. The selection was mainly based on their presence in the local market, and included:

- Proprietary: IBM BP Manager [14],Oracle BPM[15], Apia[16],GXFlow[17]
- Community edition: Aris Platform[18], Bizagi[19], Bonita[1].

Fig. 4 (a) shows an example of the results obtained from evaluating selected characteristics on the tools. As before, the semaphore metaphor is used, and the theoretical and practical evaluations are shown. Being proprietary tools, for the practical evaluation it was necessary to obtain licenses for the products to evaluate. Due to difficulties in obtaining the software and/or corresponding licenses, two tools were only evaluated theoretically, while the rest were also evaluated in practical (cf. Section 3.2), as shown in Fig 4 (a). The overall score is shown in Fig. 4 (b) which is obtained by applying the formula as before, taking into account both the importance defined for each characteristic by the organization and the results obtained by the characteristic evaluation. We do not disclose which value corresponds to which tool for confidentiality reasons regarding definitions in the projects carried out with the enterprises involved.

As conclusions of this evaluation we can highlight as a key point the original nature of the tool. Three of the tools have as main objective support for BPM, so their architecture and feature set are intended for these purposes; while the remaining tools are extensions of larger tools which were developed with other objectives. Those tools which focus on BPM have advantages in characteristics such as: installation, usabil-

---

[14] IBM BP Manager, http://www-01.ibm.com/software/ integration/business-process-manager/
[15] Oracle 11g, http://www.oracle.com/us/technologies/bpm/suite/overview/
[16] Apia, http://www.statum.biz/web/guest/apia
[17] GXflow, http://www.genexus.com/gxflow
[18] ARIS Platform, http://www.softwareag.com/corporate/products/aris
[19] Bizagi BPM Suite, http://www.bizagi.com/

ity, understanding, documentation, simpler architectures, etc. The other tools must adapt to certain preset parameters, providing less specific support to BPM, increasing the learning curve and presenting more installation problems, among others. However, the latter proved more powerful, with more features available, management tools and administrative consoles. As for the language of the process engine both XPDL and WS-BPEL are provided for process execution, and there is less support for BPMN 2.0 which is mainly provided within modelers but not for execution. In most of the tools wizards support is provided for BPs implementation helping reduce the development time, for example to integrate web services from WSDL definitions.
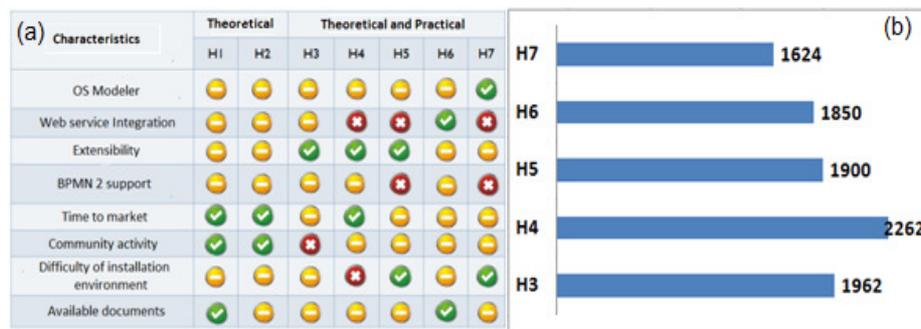

Fig. 4. Sample of (a) characteristics evaluated in Proprietary tools (b) overall scores

5   Conclusions

We have presented a systematic approach for evaluating BPMS tools both open source and proprietary, which includes a methodology and a list of key characteristics for this kind of software, as well as a way to instantiate each evaluation to the specific context of the organization performing it, providing different results for different needs. We provide a list of relevant key characteristics for BPMS tools, and a way of evaluating the provided support by means of tests cases and a case study to provide an overall view of the tool support. We believe that the evaluation methodology we propose can be also applied to other type of software, by changing the list of characteristics according to the software being evaluated, and following the defined process. We have also illustrated the use of the approach by means of case studies regarding open source and proprietary BPMS. We can conclude that all tools have advantages and disadvantages, and can be suitable for different contexts. As there is an increasing demand from organizations to incorporate BPMS platforms, we believe that a key element and contribution of our evaluation approach is to take into account the organizational context, allowing different organizations to select different tools regarding their specific needs. We can also conclude that regarding this kind of software some open source BPMS such as Bonita, Activiti and Intalio CE are, with respect to some aspects, as complete and competitive as other products from major existing vendors.

As future work, we plan to evaluate again some open source BPMS since they all have improved significantly in the last few years (we have continued using them in

many projects and courses, mainly Activiti and Bonita), and to add new others. We also plan to provide tool support for the methodology by generating a benchmark for the evaluation as well as a tool allowing evaluators and users to assign weights to the characteristics and generate recommendations and comparisons between tools. Finally, we plan to include other aspects to the evaluation, e.g. non functional aspects.

Acknowledgements

REFERENCES

[1]   M. Weske: BPM: Concepts, Languages, Architectures, Springer, 2007.

[2]   W.M.P. van der Aalst, A. ter Hofstede, M. Weske: BPM: A Survey. In Proceedings International Conference on Business Process Management (BPM'03), 2003.

[3]   J. F. Chang: BPM Systems: strategy and implementation. Auerbach Publications, Taylor & Francis Group, 2006.

[4]   OMG. Business Process Model And Notation (BPMN) v2.0, 2011

[5]   WfMC, XML Process Definition Language (XPDL), 2008.

[6]   OASIS, WS Business Process Execution Language (WS-BPEL), 2007.

[7]   W.M.P. van der Aalst, A. ter Hofstede, B. Kiepuszewski, A. Barros: Workflow patterns. Dist. &Parallel Databases, 14(3). 2003

[8]   R. Garcês, T. Jesus, J. Cardoso, P. Valente.: Open Source Workflow Management Systems: A Concise Survey, In 2009 BPM & Workflow Handbook, Future Strategies Inc., 179-190, 2009

[9]   Gartner Group, http://www.gartner.com/technology

[10] TEC, Technology Eval., http://www.technologyevaluation.com/

[11] ISO/IEC 9126, Software engineering, Product quality, superseded by ISO/IEC 25000 SQuaRE http://www.iso.org/

[12] A. Tsalgatidou: Selection Criteria for Tools Supporting Business Process Transformation for Electronic Commerce, Proceedings of EURO-MED NET, 1998

[13] P. Wohed, B. Andersson, A. ter Hofstede, N. C. Russell, W.M.P. van der Aalst: Patterns-based Evaluation of Open Source BPM Systems: The Cases of jBPM, OpenWFE, and Enhydra Shark. Information and Software Technology, 51(8):1187-1216, 2009.

[14] S. Harrer, J. Lenhard, G. Wirtz: Open Source versus Proprietary Software in Service-Orientation: The Case of BPEL Engines, In Proceedings 11th International Conference on SOC (ICSOC'13), 2013

[15] D. Morera: COTS Evaluation Using Desmet Methodology & Analytic Hierarchy Process (AHP). PROFES 2002, LNCS 2559: 485-493. Springer, 2002.

[16] P. Lawlis, K. Mark, D. Thomas, T. Courtheyn: A Formal Process for Evaluating COTS Software Products. IEEE Computer, 34 (5): 58-63, 2001

[17] D. Taibi, L. Lavazza, S. Morasca: OpenBQR: a framework for the assessment of OSS. OSS 2007: 173-186

[18] F. Tarawneh, F. Baharom, J. Yahaya, F. Ahmad: Evaluation and Selection COTS Software Process: The State of the Art. International Journal on New Computer Architectures and Their Applications (IJNCAA), pp: 344-357, The Society of Digital Information

[19] C. Richardson, D. Miers, A. Cullen, J. Keenan, The Forrester Wave: BPM Suites, Q1 2013, How The Top 10 Vendors Stack Up For Next-Generation BPM Suites, March, 2013

[20] J. Sinur, J. Hill: Magic Quadrant for Business Process Management Suites. Gartner Inc. 2010

[21] J. Sinur, W. Schulte, J. Hill, T. Jones: Magic Quadrant for Intelligent Business Process Management Suites. Gartner Inc. 2012

[22] R. Yin: Case Study Research: Design and Methods. Sage Publications, Inc, 2002