A survey on business processes management suites

A. Meidan*, J.A. García-García, M.J. Escalona, I. Ramos

Web Engineering and Testing Early (IWT2), University of Seville, Avenida Reina Mercedes s/n, Seville 41012, Spain

1. Introduction

BPM (Business Processes Management) is a field of management that can be defined as a paradigm that includes methods, techniques, and tools to support the design, enactment, management and analysis of operational business processes [45]. BPM aims to strategically assess the processes a company carries out and to continually improve effectiveness and efficiency of Business Processes (BP) within organizations in order to: (i) achieve lower costs; (ii) improve quality; and (iii) gain in productivity and competitiveness in relation to other organizations of the same business area.

Today, international best practice guidelines for project management, such as PMBOK (Project Management Body of Knowledge) [31] or Prince2 (Projects in Controlled Environments 2) [38], reference standards, such as CMMI (Capability Maturity Model Integration) [3], and general guidelines for quality assurance, such as ISO 9001:2008 [19], recommend that organizations should formally manage their BP in order to increase their ROI (Return on Investment). Thus, defining and managing business process models seems to be an essential mechanism to improve any business activity.

Nowadays, BPM as a continuous improvement method [24], is a common practice followed by a large number of organizations in all areas of business. In fact, organizations are aware of the need to deploy well-defined processes, pursuing not only raising their maturity level, but also improving the way in which their products are developed and managed, and thus, their quality [10].

At present, there are a wide variety of software tools (named Business Process Management Suites, BPMS) that allow managing BP lifecycle to make easier BPM application in business environments. However, each of these BPMS has a wide assortment of prices and functionalities. In this context, the selection of a concrete solution may be quite a difficult, laborious and complex undertaking. A sound selection requires a complete analysis of the most popular available solutions. Otherwise, it may lead to choosing an inadequate workflow product that will not support efficiently BP in an organization.

This paper aims to describe a survey with which organizations can compare specific BPMS according to their own organizational objectives. This survey is oriented to each phase of the process lifecycle and it has been carried out using a proven method [8] which combines well-known techniques such as Systematic Literature Review (SLR) [23] and quality frameworks [7] (based on a characterization scheme). This technique used in our survey has already been used in other contexts [9] and we have obtained successful results thanks to the application of the quality models.

In this sense, the definition of our characterization scheme has been defined taking into account the expert opinion of important Spanish companies which are Information, Communications and Technologies (ICT) companies. Our research group has a long collaboration with companies and has received important feedback from them.

Moreover, this survey describes how our systematic method has been instantiated on specific open source [39] BPMSs which have been selected after performing a SLR. This paper is focused on open source BPMSs because open source software is having a growing impact on software industry by becoming an important competitor to commercial software [43]. In fact, many organizations (from commercial, government and non-profit sectors) have found benefits in applying open source software in their organizations [42]. In addition, we have chosen to focus on a case study related to open source BPMSs because commercial BPMSs can become very expensive for SMEs (Small and Medium Enterprises).

This evaluation is very helpful for business and academic communities to choose the most appropriate BPMS either for business or

* Corresponding author.
E-mail address: ayman.meidan@gmail.com (A. Meidan).

research objectives. This paper also illustrates the relation between the academic research on BP lifecycle and the feature supported in today's BPMS. These aspects help identify possible areas for improvement and guide future researches.

Finally, this paper is structured as follows: Section 2 introduces other studies on BPMS in order to know other analyses previously conducted and their scope thereof. Section 3 describes the survey conducted by our team and it proposes a quality model based on a characterization scheme for evaluating BPMS and how these tools support each one of phases of the BP lifecycle. Section 3 also presents how our model has been instanced on open source BPMS. At the ending, Section 4 concludes stating some lessons learned and possibilities for future work.

2. Related work

In recent years, different works have been presented in order to analyze and compare BPMS in different application environments.

In [27], Murray offers a case study that analyzes the implementation of a commercially available healthcare workflow system in two hospital environments. The case study protocol uses an analytical framework built upon six theoretical propositions identified as having a major impact on the implementation of workflow technologies. The proposed framework also includes an organizational perspective which is enriched and detailed in our work.

In [11], authors evaluate the main features offered by ten different open source BPMS. The comparative framework proposed in this paper is based on the Workflow Management Coalition (WfMC) reference model [17] and on the runtime and design time perspective of workflow systems.

The framework proposes general features—so many important criteria are not included—like the interoperability of the process definition models, the modeling of the business rules, exception handling, document management, or process validation and simulation, among other important features.

The research developed by Stoilova et al. [40] addresses problems related to the assessment and comparison of workflow management systems. A special emphasis is placed on comparing existing open source software supporting the different technologies. These authors suggest an evaluation template (composed of eight categories) according to the recommendations of the standard ISO/IEC 9126 [18] for assessing the quality of software products. This work describes a methodology based on general qualities of a software product which has been applied to evaluate a single product as an example, but authors do not provide a comparison between different BPMSs. In this sense, our paper (along with the methodology used and our quality model) extends, completes and improves the Stoilova's paper because our article provides a comparison for different open source BPMS currently available in the market.

In [41], authors provide a short chronology about the evolution of BPM systems and they present four classification groups according to its support of modeling and execution functionalities (software for business and scientific workflows, software related to their software language design, software associated with their supported standard; and open source and commercial software) according to different criteria. Moreover, although providing conditional structures is not mandatory for a BPMS, it is very important to provide support for conditional workflows in terms of error handling, flexibility and robustness. However, authors do not evaluate the available BPMS in the same group; for example, they do not provide comparison between open source BPMSs.

In [1], authors study BPMS and their support for conditional structures such as *if*, *switch*, and *while*. Authors compare implementation of common conditional structures using each of these workflow management systems via case studies, as well as discuss capabilities of each system.

In [26], authors conducted a research study on BPMS. This paper details some workflow-related concepts and their typologies, references of some BPMS and current research trends and hotspots.

Finally, there are various BPMS comparisons carried out yearly by renowned analysts such as Gartner [35] [36], Lustratus Research [4] and Forrester Research [32]. The first one sets a new bar for BPM market by evolving beyond a focus on process analysis toward systems that guide and recommend the next most effective action or decision. This report also assesses vendors' predictive analytics, complex event processing, social media and mobile platform capabilities. In the second one, Lustratus only considers BPM offerings from a relatively young vendor, Appian, together with two industry stalwarts in the shape of Oracle and IBM. The third one presents findings about how well each vendor fulfills the evaluation criteria and where they stand in relation to each other to help companies select the right solution to support launching and scaling enterprise-wide BPM programs and initiatives. For that purpose, Forrester evaluates the 10 most significant BPM suite vendors against 59 criteria that reflect the requirements of organizations running large-scale BPM programs.

After carrying out this initial study on related works, it should be noted that many of these works describe a specific set of features that cannot be grouped into the lifecycle phases of BPM. In addition, most of these studies are based on commercial BPMSs and they do not provide feedback on open source BPMS that can become important for SMEs.

3. Protocol followed at this survey

As mentioned before, this paper describes a survey based on a systematic review method and quality models with which it is possible to identify and compare BPMSs. However, it is not necessary to define a new systematic review method because over the last decade, many authors have proposed similar methods. These methods are usually known as SLR (Systematic Literature Review) and relatively recent in the software engineering context.

SLR has become relevant in this area as a means to identify, evaluate and interpret all available data to answer research questions on a particular topic in software engineering. It has been gaining in importance as a systematic and structured approach regarding literature reviews since 2004, when Barbara Kitchenham [20] proposed special guidelines that were adapted to cope with specific problems in the software engineering area.

A few years later, Zhang and Babar consider [53] that SLRs are mainly carried out to find innovative ideas for further research. In this sense, SLR is conceived as a complete process based on identifying, evaluating and interpreting all available documents related to a particular thesis in a specific investigation area. This process has its own terminology and can be applied in different ways. In this line, authors introduce new vocabulary concepts and different perspectives of SLRs [53].

Regarding SLR process, Zhang et al. consider that one critical step in any SLR is to design and execute an appropriate and effective search strategy [52]. Authors argue that SLR process consists of time-consuming and error-prone activities and consequently, these activities must be carefully planned and implemented. They also explain that there is an apparent need for a systematic approach to design, execute, and evaluate a suitable search strategy for optimally retrieving the target literature. With this aim, authors propose an approach [52] consisting of a collection of known studies, and corresponding 'quasi-sensitivity' to the search process for evaluating search performance. This proposal is obtained after selecting and studying a wide range of SLRs in order to understand the *state-of-the-practice* of search strategies in evidence-based software engineering.

Moreover, some authors suggest that the software engineering research community is starting to adopt SLRs consistently as a research method [5]. However, the majority of SLRs do not evaluate the quality of primary studies and fail to provide guidelines for practitioners, thus decreasing their potential impact on software engineering practice.

Taking into account all these proposals, we have decided to follow the protocol defined by Kitchenham in our survey [[21–23]] because it is considered one of the most acknowledged in software engineering. Our paper also takes into consideration Wohlin and Prikladnicki's [51] conclusions about SLRs in software engineering, since they consider that the search strategy is key to ensure a good starting point for the identification of studies and ultimately for the actual outcome of a particular study. In addition, we have combined Kitchenmham's protocol with SEG's (Software Engineering Group) proposal. SEG [34] proposes to establish a common characterization criterion (i.e., a quality model) to define each approach before conducting the review.

Fig. 1 shows the SLR process we have followed in our survey and the next sections describe in detail each of the phases it involves. This SLR essentially involves three phases:

(i) Planning the review, which aims to decide which method will be used to carry out the review as well as identify and formulate the thesis that the systematic review must validate;

(ii) Conducting the review, which consists in finding and evaluating whether many primary studies associated with the research questions are adequate and relevant enough to be possible sources for further analysis; and

(iii) Reporting the review, which deals with writing up the results of the review and reporting them to potentially interested parties.

3.1. *Planning the review*

In this first phase, our research team has planned the review protocol in order to delimit the specific context of our goals, and identify and formulate the thesis that the systematic review must prove. In this case, our theses are: "What are main open source BPMSs available to manage the BP lifecycle?" and "What could quality model be provided to evaluate these BPMSs?".

The planning phase also includes the following issues: Defining Technical and Research Questions (TRQ), that means, the specific questions the study should answer; performing Quality Assessment (QA), which defines each characteristic that has to be valued for each technical solution; and designing a Characterization schema, which can be considered the global result of this first phase and a quality model to evaluate BPMSs. In the end, the research team has to get a quality model similar to a checklist, where each QA is represented in order to answer each TRQ.

Later, this schema is going to be instanced in Section 3.1.1 and completed for each technical solution under study, but firstly it is necessary to concrete above issues. The following sections will describe in detail each of these aspects.

3.1.1. *Technical and research questions*

On the one hand, this paper aims to answer the next TRQ and decide on our information sources:

- RQ1. What are open source BPMS available in the market and what do they offer?
- RQ2. What areas of improvement are needed for the selected BPMS?

A large number of identified search keywords picked up from these questions have been used in the review process. Some of them are: "Business Process Management Suite(s)", "BPM tool" and "Workflow Management System", among others.

On the other hand, we have initially followed Brereton's recommendations about information sources. In [2] this author identifies seven electronic information sources of relevance to software engineers in an attempt to perform an exhaustive search: IEEExplore, ACM Digital library, GoogleScholar, Citeseer library, Inspec, ScienceDirect, and EI Compendex. In addition, we include some other important sources in Computer Science to be searched in, such as Google, ISI Web of Knowledge, SCOPUS, DBLP, Springer Link and Wiley Online Library.

3.1.2. *Quality assessments*

A questionnaire, which has to be filled out for each included BPMS, has been elaborated with the purpose of assessing the quality of the obtained studies. Three possible answers can be chosen for each question: yes, no or partially. Below, we present both quality assessment questions and criteria described to evaluate them.

QA1: Has the BPMS maker provided a new version of the BPMS in the past year (July 1st 2014 to July 1st 2015)? (*It is necessary to evaluate the BPMSs whose line of development has not died in order to provide an actual comparative study useful for organizations*)

- Yes, they have.
- No, they have not.

QA2: Do the BPMS have an active user community and official support? (*Having an active community within the context of open source tools is relevant*) [48])

- Yes, it does. BPMS has had a forum with an active community for three months.
- No, it does not.

QA3: Does the BPMS maker provide sufficient documentation (manuals, videos or examples) on his/her BPMS?

- Yes, they do.
- No, they do not.

3.1.3. *Quality model based on a characterization scheme*

Finally, we have defined a Quality model to evaluate BPMS and noted down the organization needs and research objectives. This model has been implemented using a *characterization scheme* which focuses on the most marked criteria in the literature and among our partners.

It also enables obtaining a uniform definition for each BPMS, what facilitates comparative studies. In fact, we have taken out the results of several doctoral theses related to the process management and feedback obtained from our partner companies. In this sense, our research group, IWT2 (Web Engineering and Early Testing), has and continues to have extensive experience in technology transfer projects where BPM technologies are used (for example, we are working with FujitsuTS, Everis, Airbus Military, WellnessTelecom, SOLTEL, Andalusian Regional Ministry of Health and Social Welfare and Andalusian Regional Ministry of Culture, Education and Sport, among others bodies). We have defined criteria for our characterization scheme after considering the needs of these partner companies with which we have collaborated and we are currently working on various R&D projects. We received feedback after carrying out several interviews and brainstorming sessions with them.

Moreover, BPM can be considered a process-oriented management strategy [15] with a clear multidisciplinary nature, as BPM can be applied to different contexts or domains and be used by different user profiles [33]. This situation has conditioned the appearance of different views, definitions and perspectives of BPM lifecycles and its continuous improvement [45] [13] [14], as well as other ones which are more general, such as Shewhart's cycle [54], which defines a management model for continuous business improvement and incremental problem solving.

This paper presents an interpretation of one of the aforementioned BP lifecycles. Specifically, this paper pays attention to Hill's proposal [14] because it is the most complete one regarding the number of phases in BP lifecycle. In [14], authors recommend a lifecycle based on nine phases (Fig. 2): *Discovery*, to establish the business methodology for assessing the work and measure the success of continuous improvement; *Modeling*, where BP and business objectives are statically defined; *Simulation*, where bottlenecks are detected before executing; *Deployment*, where scripts to integrate BP into the organization's information systems are developed; *Execution*, where BP are instantiated; *Monitor*, where information of performance is collected in real time; *Analytics*, where key performance indicators are assessed in order to detect deviations of organizational objectives; *Optimization*, where actions are carried out to improve BP performance and reduce potential risks to the organization; and *Refine*, where the business methodology and BP are redefined taking into account analysis in previous phases (continuous improvement mechanism).

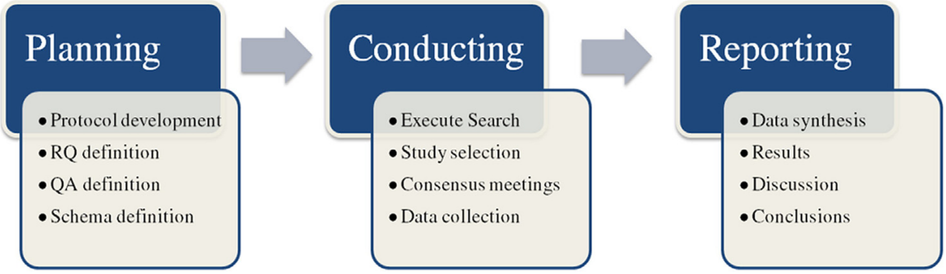


Fig. 1. Phases of our SLR process.

3.2. Characterization scheme

In this section we detail our quality model based on a characterization scheme ordered, as much as possible, by BP lifecycle phases. Consequently, we have taken Hill's BP lifecycle as a reference to group our criteria which finally includes 41 characteristics divided into these phases: Modeling; Design; Deployment; Execution and Operation; Monitoring and Control; and Analysis.

3.2.1. Modeling phase

In this phase, the organization has to use a formal language to model BPs. This model describes different BP perspectives (functional, flow control, information and organizational perspective), which allows stakeholders (domain experts, process engineers and end users, and administrators) to understand it without ambiguity [44].

There are many BP modeling languages available, such as BPMN (Business Process Modeling Notation) [29], UML and its activity diagrams [30] and YAWL (Yet Another Workflow Language) [46]. In this sense, interoperability among the different languages is an important BPMS aspect; interoperability allow the interchange (import/export) of business process models between different BPMSs [37], thus, interoperability support is an important criterion in the evaluation of the BPMSs.

Another important aspect is to allow defining business rules by which the process is constrained in terms of business goals. These business rules could be embedded in the process definition or defined using Business Rule Management Systems [36].

The possibility to reuse processes (or part of them) is also another crucial criterion for BPMS because reuse processes facilitate the task of modeling business processes in two ways: i) it improves the quality of the models through reuse of established and optimized artifacts; ii) it reduces the process modeling time by avoiding modeling the same business process or part of it multiple times [25].

Moreover, defining Process Performance Indicators (PPIs) is the next important step after describing the BP model. PPIs constitute a relevant instrument to evaluate process performance and can be considered the first step to carry out a continuous process improvement [6]. Consequently, a BPMS must allow defining and monitoring PPIs in order to carry out further analysis. The ability to define PPIs (and the ease of use for end users) is another important criterion in the evaluation of BPMS.

The ability to generate BP documentation is one of the key objectives of BPMS [12] because it is important for communication among stakeholders [49]. In this sense, it is interesting to evaluate if BPMS allow generating documentations in order to reflect different perspectives of the process, such as documentation of activities, roles and information.

Finally, after justifying criteria used, we briefly present the set of criteria with which we are going to evaluate the process modeling phase.

(1) Supported Business Process modeling languages.
(2) Interoperability; compatibility with other BP modeling languages; ability to import and export the process definition.
(3) Possibility to Reuse BP models or part of them.
(4) Supported perspectives or views (activities, flow, data and organization).
(5) Modeling of business rules, including Integrated Business Rule Management (BRM) and supported techniques to state the control flow as an expression, such as, decision tables, decision trees, Domain Specific Language and Natural Language.
(6) Modeling of PPIs, i.e., BPMS provides mechanisms with which it is possible to define indicators, kinds of types of indicators or data templates.
(7) Generation of process documentation.

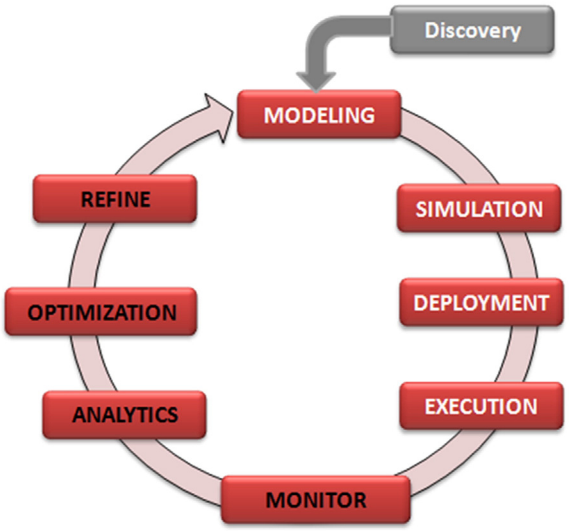


Fig. 2. Hills BP lifecycle [40].

3.2.2. Design phase

The design phase is necessary when the organization implements process models using IT (Information Technology) support. For this purpose, it is important to design and implement several features such as user interfaces, interactions with other systems and translations to execution models, among others. Next, we are going to further discuss criteria included in our characterization scheme for evaluating this phase.

On the one hand, it is well known that users interact with BP and its business data using graphical user interfaces. In this sense, it is important to assess if BPMS provide mechanisms to define and implement these interfaces in a friendly way (e.g., using separate applications or modules integrated into BPMS).

BPMS also support various programming languages in order to encourage designers to create and modify the user interface and implement interfaces to communicate with other organizational services.

Another important criterion is the user management which is necessary to define and control the access privileges of each role in the process execution; this could be done by allowing the definition of the organizational structure within the BPMS or by importing the structure of roles from an existing system like LDAP.

In addition, in the design phase, the organization could link service level agreements with its process performance indicators to facilitate monitoring and analyzing BP service level.

On the other hand, another important task in this phase of lifecycle consists in implementing exception handling to avoid any unexpected behaviour during the execution of BP (exception handling could be implemented in diverse levels, for instance, sequence flow or communication with external services) and ensuring data integrity and ACID (Atomicity, Consistency, Isolation and Durability) transaction at BP level [47].

The last important design criterion in our scheme is the translation into executable models (e.g., WS-BPEL [28] and YAWL). This process could be fully automated or in some cases need human and manual intervention. Anyway, it must guarantee traceability between BP model and execution model.

Finally, after justifying criteria used, we briefly present the set of criteria with which we are going to evaluate the process design phase.

(1) The support for programming languages for implementing services and applications.
(2) Designing user interface (manual/automatic).
(3) Describing roles in a particular manner (by names/description or capabilities).
(4) Support for importing organizational structure from other systems.
(5) Support for manual/automatic assigning roles to users.
(6) Support for adding SLA and linking SLA PPIs to BP.

Table 1
Open source BPMS.

	QA1	QA2	QA3		QA1	QA2	QA3
Activiti	•	•	•	YAWL	•	•	•
BonitaOS	•	•	•	XFLOW			
Intalio|BPMS	•	•		TAVERNA			
jBPM	•	•	•	OpenWFE			
Kbee workflow				WfMOpen			
ProcessMaker	•	•	•	Freefluo			
uEngine BPM	•	•	•	Syrup			
Camunda	•	•	•	JawFlow			
Enhydra shark							

Table 2
Evaluation of Bonita BPM.

I. Modeling criteria	
I.1 Supported BP modeling languages	BPMN 2.0
I.2. Interoperability and compatibility	Import: jBPM XML 3.2, BPMN 2.0, XPDL 1.0 Export: Only BPMN 2.0 and image file
I.3 Reuse BP models	Not supported in the community edition
I.4 Modeling views	Support all views
I.5 Modeling of business rules	Partially supported: It does not include native BRM tools, support expressions and decision table techniques
I.6 Modeling of PPIs	Not supported in the community edition
I.7 Generate process documentation	Not supported in the community edition
II. Design criteria	
II.1 Supported programming languages	Java
II.2 Designing user interface (UI)	Supports automatic generation of UI and supports manual modification in the community edition
II.3 Way of describing roles	Names/descriptions are supported
II.4 Support for importing organizational structure	Supported
II.5 Support for assigning a user to a role	In a manual way
II.6 Support for adding SLA and linking SLA KPIs	Not supported in community edition
II.7 Support exception handling and transaction control	Exception handling is supported Transactions control is not supported
II.8 Translation into executable models	In an automatic way
II.9 Supported BP execution languages	Not supported
III. Deployment Criteria	
III.1 Support for distributed execution	Not supported in the community edition
III.2 Support for integration into other systems and services	It supports more than 80 connectors to allow integration into different systems and services
IV. Execution and Operation	
IV.1 Version management of BP models	Supported
IV.2 Support for calendar management	Including connector to Google calendar
IV.3 Support for informing users	Push and pull technologies are supported
IV.4 Document management	Supported
V. Monitoring and Control	
V.1 Support for technical monitoring and control	Supported
V.2 Support for business monitoring BAM	Partially supported
V.3 Change the role or resource for a process instance activity	Supported. It is possible to perform dynamic task assignment for actors of running process instances tasks
V.4 Support for changing business rules	Not supported
V.5 Support for optimized execution	Not supported
V.6 Ability to deal with failures	Partially supported
V.7 Support for changing the workload balance	Supported
V.8 Support dashboards and reports	Not supported in the community edition
V.9 Support for detail levels	Not supported in the community edition
V.10 Support for different views of monitoring	Not supported in the community edition
VI. Analysis criteria	
VI.1 Support for process verification	Validation of the business process diagram and connectors is available. It also includes debug mode
VI.2 Support for process simulation	Supported
VI.3 Support historical data available for analysis	Partially supported (there is a simple log, APIs to access Engine and infrastructure data)
VI.4 Support for suggestions on improvement	Partially supported; after verifying (validating) the diagram, Bonita BPM displays suggestions to fix errors
VI.5 Support for BI and Process mining tools	Partially supported
VII. Other criteria	
VII.1 Documentation	Support online documentation and PDF
VII.2 Training	Videos, examples, and also a big community and many organized events and webinars are available
VII.3 Tool maturity	Initial release in 2009
VII.4 Commercial support	Available

(7) Support exception handling and Transaction Control.
(8) Translation into the executable business process model (manual/automatic).
(9) Support business process execution languages.

3.2.3. Deployment phase

In this phase, organization processes are deployed in order to be available for end users, as well as to connect and integrate the processes to other internal or external resources and services. In this sense, the organization size and the expected load balance should be studied in order to determine the best way to deploy and integrate BP to existing organization systems and resource systems.

Considering these aspects, we evaluate if BPMS allows running in distributed environments and multiple engines in the same physical server. In addition, we evaluate if BPMS support integration technologies into other organization services and resources, such as REST (Representational State Transfer) and WSDL (Web service Definition Language) technologies.

Finally, after justifying criteria used, we briefly present the set of criteria with which we are going to evaluate the deployment phase.

(1) Support for distributed execution.
(2) Support for integration with other systems and services.

3.2.4. Execution and Operation phase

The process execution engine is part of BPMS and is responsible for instantiating BP models. These instances determine the execution flow of the process according to its data, events and business rules, among other aspects. In addition, at this phase, users interact with BP instances throws graphical user interfaces in order to perform tasks, provide data to the process, or work with documents, among other activities.

Table 3
Evaluation of ProcessMaker.

I. Modeling criteria	
I.1 Supported BP modeling languages	BPMN 2.0
I.2. Interoperability and compatibility	Only supports ProcessMaker format
I.3 Reuse BP models	Supported
I.4 Modeling views	All are supported
I.5 Modeling of business rules	Partially supported: It does not include BRM tool, but it supports PHP expression technique.
I.6 Modeling of PPIs	Not supported
I.7 Generate process documentation	Not supported
II. Design criteria	
II.1 Supported programming languages	JavaScript and PHP
II.2 Designing user interface (UI)	Manual and automatic are supported
II.3 Way of describing roles	Names and description are supported and allow assigning permissions
II.4 Support for importing organizational structure	Support connection to LDAP and AD
II.5 Support for user-role assigning	Manual
II.6 Support for adding SLA and linking SLA KPIs	Supported only in the commercial edition
II.7 Support exception handling and transaction control	Triggers could be used as exception handling
II.8 Translation into executable models	Automatic
II.9 Supported BP execution languages	ProcessMaker
III. Deployment Criteria	
III.1 Support for distributed execution	Not supported
III.2 Support for integration into other systems and services	Support REST API and Web services
IV. Execution and Operation	
IV.1 Version management of BP models	Not supported
IV.2 Support for calendar management	Supported
IV.3 Support for informing users	Both techniques are supported
IV.4 Document management	Supported
V. Monitoring and Control	
V.1 Support for technical monitoring and control	Not supported
V.2 Support for business monitoring BAM	Partially supported: Case Tracker is available to provide information about the case status. Reports and dashboards are also available
V.3 Change the role or resource for an activity of process instance	Supported
V.4 Support for changing business rules	Not supported
V.5 Support for optimized execution	Not supported
V.6 Ability to deal with failures	At activity level by triggers
V.7 Support for changing the workload balance	Supported
V.8 Support dashboards and reports	Basic dashboard and report are supported
V.9 Support for detail levels	Not supported
V.10 Support for different views of monitoring	Not supported
VI. Analysis criteria	
VI.1 Support for process verification	Not supported
VI.2 Support for process simulation	Not supported
VI.3 Support historical data available for analysis	Supported by events Log
VI.4 Support for suggestions on improvement	Not supported
VI.5 Support for BI and Process mining tools	Partially supported
VII. Other criteria	
VII.1 Documentation	HTML documentation is available
VII.2 Training	Webinars and commercial training courses are available
VII.3 Tool maturity	Since July 5, 2011 three versions has been issued
VII.4 Commercial support	Supported

Taking into account these aspects, there are many other BPMS features which could support the users to organize and complete their tasks. For instance, calendar and document management, task inbox, email notifications, etc. These aspects could be internally implemented in the BPMS or externally by supporting tools or add-ons from other vendors.

Moreover, one of the most important features in the execution phase is the version management by which it is possible to simultaneously execute different versions of the same process and keep track of all running versions. This scenario is interesting when organizations need to evolve their processes without losing information of obsolete instances. This situation is very common in industrial environments (e.g., aeronautical or automotive environments) in which processes have a very long life.

Finally, considering this justification, criteria with which we are going to evaluate the Execution and Operation phase are:

(1) Version management of process models.
(2) Support for calendar management.
(3) Support for informing users of their tasks, either supporting push techniques (i.e., BPMS sends notification to the user about pending tasks) or pull techniques (i.e., BPMS allows checking if each user has pending tasks).
(4) Document management.

3.2.5. *Monitoring and Control phase*

Monitoring or tracking the organization using BPM techniques [47] is essential to continuously improve the organization. In those cases where technical monitoring deals with aspects like system response time, system load, server issues and connection problems, business monitoring focuses on supervising process instances. This monitoring is related to control the process in order to obtain information to calculate indicators.

Considering these aspects, we have included features in our quality model for assessing, controlling and monitoring techniques (e.g., the availability of log files and other historical resources like databases), user notifications in case of failures at infrastructure levels (e.g. communications or servers) and business levels (such as reaching predefined values of PPI, among others).

In addition, our characterization scheme evaluates control features with which BPMS respond to any of the above situations, such as the ability of changing the activity's role (or resource) in order to balance the workload.

Moreover, an important aspect is to check the status of all running instances and their indicators. This type of control can be carried out using dashboards with which it is possible to improve the prediction of problems and risks during the execution of processes.

Finally, considering this justification, Monitoring and Control criteria are summarized below:

(1) Support for technical Monitoring and Control, like resource consuming (human, system or communication, among others).
(2) Support for business monitoring BAM (active/passive), i.e., monitoring log files, calculating PPIs of BP and requesting information about the running process instances.
(3) Change the role or resource for an activity of process instance, i.e., change the actor or storage.
(4) Support for changing business rules for process instance.
(5) Support for optimized execution according to some measurable criteria, i.e., changing process structure or the activities flow for the running instances to respond to special situations.
(6) Ability to deal with failures (engine, system, activity and communication).
(7) Support for changing the workload balance among users.
(8) Support dashboards and reports.
(9) Support for detail levels.
(10) Support for different views of monitoring information.

3.2.6. *Analysis phase*

Analysis is one of the four keys of BPM (model, analyze, enact and manage) because it is the phase in which indicators monitored are analyzed. Once the performance data is analyzed, the organization has insight to improve their processes. That is the essence of continuous improvement.

There are many techniques for analyzing data (e.g., studying models or event logs [47], among others) that could be applied at different phases in BP lifecycle. For example, during the modeling phase the verification analysis could be used to ensure the correctness of the process model against the modeling language and the execution model [50]. Another analysis technique is the *simulation/what-if analysis*, which is very useful to get ideas on how to reduce costs while improving service levels [47].

Considering all this, it is important to have historical data and data mining tools in order to automate and streamline the application of analytical techniques of process performance.

Finally, after justifying the criteria used, we briefly present the set of criteria with which we are going to evaluate the analysis phase.

(1) Support for process verification.
(2) Support for process simulation.
(3) Supported types of historical data available for analysis, i.e., execution logs, computed PPIs, application data or status of IT infrastructure.
(4) Support for suggestions on improvement.
(5) Support for Business Intelligence and Process Mining tools.

3.2.7. *Other features*

Finally, our quality model includes other interesting features related to documentation, commercial support, and maturity. These characteristics are summarized below:

(1) Support for generating documentation, e.g. PDF or HTML format.
(2) Training, e.g., video training, case studies, examples, webinars and community events.
(3) Tool Maturity, i.e., dates of first and last version of the product.
(4) Commercial support, which covers if the provider of the tool offers commercial support, new features support or deployment support. This characteristic could be important for companies lacking an experienced technical department.

3.3. *Rating method*

Once each feature of our quality model is submitted, we establish a rating method to evaluate each BPMS consistently. In this sense, we have designed an evaluation method to obtain a quantitative evaluation for each criterion, group of criteria and for the overall BPMS. This method allows an easy and homogeneous comparison among BPMSs at many levels:

- Criterion level. It is obtained by assigning points to each criterion based on a scale from zero to four [0–4], where: 4 points mean that the BPMS provides total native support to the evaluated criteria; 3 points mean partial native support to the evaluated criteria by the evaluated BPMS; 2 points mean that the BPMS includes programming interfaces that support the evaluated criteria development; 1 point means that a third party component is necessary to support the criterion; and 0 points means that BPMS does not support the criterion.
- Partial Score (PS). It is obtained per group of criteria and is calculated by adding resultant scores of all sub-criteria of the group dividing the result by the maximum score of the group of criteria and, later, multiplying the result by 10. “Eq. (1)” shows this calculation, where *n* represents number of sub-criteria in the group, *s* represents the score of the sub criteria and max represents the maximum score of the group.

$$\mathrm{PS} = \frac{\sum_{\mathrm{i}=1}^{n} \mathrm{s_i}}{\max} \times 10 \tag{1}$$

- Final Score (FS). It is the final score of the BPMS and is obtained by adding partial scores of all the groups of criteria. “Eq. (2)” shows this calculation; where n represents number of groups and “ts” represents total maximum score of all the groups of criteria.

$$\mathrm{FS} = \frac{\sum_{\mathrm{i}=1}^{n} \mathrm{PS_i}}{\mathrm{ts}} \times 10 \tag{2}$$

4. Conducting the review

Once achieving the planning phase, the next steps are to

(i) Execute our review protocol and
(ii) Evaluate each BPMS according to our quality model.

4.1. *Selection of BPMS under study*

Following the review protocol, BPMS has been initially searched in aforementioned electronic research databases (see Section 3.1.1) by means of different keywords. However, after carrying out this search, almost all these electronic sources have not provided satisfactory and specific answers applicable to our paper. We have concluded that these information sources are not appropriate, since they are mainly focused on disseminating theoretical research results. Therefore, we have only used Google Scholar because:

(i) it is a generalist system that searches across a multitude electronic databases and
(ii) it allows us to find and obtain white papers and technical documentation of different tools (especially BPMS).
In addition, we have followed quality assessments defined in Section 3.1.2 in order to select BPMS under study.

Taking these aspects into account, Table 1 describes the open source BPMS that have been found and shows the results after applying our quality criterion (the • symbol means a QA answer is positive and its absence is negative).

After carrying out this selection process, the BPMS that do not infringe on any exclusion criteria and comply with all inclusion criteria are going to be included in this study. These are: Activiti, Bonita, jBPM, ProcessMaker, uEngine BPM, Camunda and YAWL. They will be evaluated in the next section. For this purpose, our quality model will be instantiated in terms of the review and evaluation method described in Section 3.1.

4.2. The evaluation

This section describes the evaluation of our characterization scheme for each BPMS selected in the previous section. In this sense, the evaluation process consists of testing each tool and performing a systematic review of all official documentation, official community and forums provided by the BPMS in order to evaluate supported criteria. The following sections briefly describe each BPMS under study and summarize evaluation results.

4.2.1. Bonita BPM

Bonita BPM 6.4 consists of two parts: Bonita BPM Studio and Bonita BPM Platform. The former is a graphical environment for creating the process and Web application forms, whereas the latter involves Bonita business process execution engine and Bonita Portal, which allow visualizing tasks and taking actions by users. The process administrator uses Bonita Portal as a tool to install, deploy and manage processes.

Table 2 describes in detail the evaluation of Bonita BPM along our quality model, but we highlight some aspects below:

- Modeling criteria: Bonita BPM supports BPMN 2.0 as modeling language and lets the user import processes defined by BPMN 2.0, XPDL 1.0 and jBPM 3.2. It also exports the process definition to XML BPMN2.0 and to image formats. The possibility to reuse BP model is only supported in the commercial edition, and it does not include the Integrated Business Rule Management (BRM) tool. However, it allows various control flow techniques such as expressions and decision tables. PPIs and generation of process documentation are only supported in the commercial edition.
- Design criteria: Bonita BPM allows the user to automatically generate graphical user interfaces and manually modify it. In addition, Bonita BPM uses Java as its main programming language; it allows importing organization structures from external systems of the organization and it defines users and roles. Bonita BPM also supports the definition of the exception handling, but it does not support transaction control.
- Deployment Criteria: Bonita BPM supports about 80 connectors to enable the integration and connection to other systems and services. Moreover, the distributed execution feature is only supported in the commercial edition.
- Execution and Operation: Bonita BPM supports version management, calendar management, and document management as well as push and pull techniques by which to inform users of their tasks.
- Monitoring and Control criteria: Technical monitoring is supported, although Business Activity Monitoring (BAM) is partially supported. It allows resources and role changes for process instances, but it does not support changing business rules for process instances. In contrast, the ability to deal with system failure is partially supported. Besides, dashboards and reports are not supported, either.
- Analysis criteria: Bonita BPM validates BP diagram, expressions and connectors. The debug mode is available to detect and trace errors. Both process simulation and basic historical data are also supported. Bonita BPM also provides suggestions to solve validation errors.
- Other criteria: Bonita BPM provides complete online documentation in PDF and HTML format, and has a lot of training materials like videos or examples. It includes a big community and many organized events and Webinars. Moreover, it was first released in 2009 and commercial support is available.

4.2.2. ProcessMaker 2.8.0 open source BPMS

ProcessMaker contains two main components: design environment and run-time engine. The former includes tools to map processes, defines business rules, creates dynamic forms, and adds input and output documents. The latter allows four cases to be started and run through the process. This engine turns the process map design into a fully-functioning application. ProcessMaker is a Web-based and cross browser.

Table 3 describes in detail the evaluation of ProcessMaker along quality model, but we highlight some aspects below:

- Modeling criteria: ProcessMaker supports BPMN 2.0 and BP reuse. There is no Business Rule Engine available, but it supports PHP expressions for the flow control. Nevertheless, ProcessMaker does not support the modeling of PPIs or the generation of process documentation.
- Design criteria: ProcessMaker supports PHP and Java script as programming languages, automatic or manual design of graphical user interfaces as well as the connection to LDAP and Active Directory to manage organization structures.
- Deployment Criteria: ProcessMaker supports REST APIs and Web service technologies to connect to other systems and services. The distributed execution is not supported.
- Execution & Operation: ProcessMaker does not support version management. Calendar management, document management and both push and pull techniques to inform users of their tasks are supported.
- Monitoring & Control criteria: Technical monitoring is not supported, whereas BAM is partially supported and basic dashboard and reporting are supported. ProcessMaker allows changing the role during the execution of the process instances and it also responds to activity failures.
- Analysis criteria: It does not support process verification or simulation, although it includes log and event files and, it supports the integration of Pentaho Reports.
- Other criteria: HTML documentation, Webinars, commercial training courses and support are available.

4.2.3. YAWL 3.0

YAWL (Yet Another Workflow Language) is an open source BPMS based on very rich workflow definition language. It has open interfaces based on Web standards, which enable developers to plug in existing applications as well as extend and customize the system in many ways. YAWL consists of the BP execution engine, process designer and resource services to handle the administration, users and application issues, custom services (with which external applications can interact with YAWL engine through XML/HTTP) like Worklet Service (Process Selection and Exception Handling Service).

Table 4 describes in detail how our characterization scheme describes YAWL, but we highlight some aspects below:

- Modeling criteria: YAWL is the modeling language of YAWL BPMS; importing and exporting BPMN models are also supported as well. It also supports reusing BP models, all modeling perspectives and the use of XQuery expression and Ripple-Down Rule (RDR) trees to define the rules. It does not support PPIs or process documentation.
- Design criteria: It supports Java as a programming language, allowing the user to generate the user interface both manually and automatically. The role could be defined by username, description and capabilities. Exception handling is supported, but transaction control is not supported. It uses YAWL as BP execution language.
- Deployment Criteria: Distributed execution is not directly supported, although it supports Web services and REST to interact with other systems and services.
- Execution and Operation: YAWL supports version management, calendar management and document management, as well as informing users by means of bull techniques.
- Monitoring and Control criteria: Technical monitoring is not supported, but BAM is supported. YAWL allows changing roles and resources to optimize process instances execution. It also deals with failures at all levels.
- Analysis criteria: Verification and simulation are supported maintaining log files. In addition, YAWL supports process mining tools.
- Other criteria: PDF documentation is available and training material is more focused on theory and scientific slides than on BPMS feature and its use. There are few videos, case studies and examples, but there are no Webinars. It has been available since 2009 and it has two main releases. It also includes a commercial support.

4.2.4. Camunda BPM 7.2.0

Camunda BPM is a Java-based framework for process automation that consists of a series of tools (Camunda Modeler and Cockpit) and components (Process Engine, Model Repository and Task List) which are used to define and execute processes, respectively. Besides, Camunda engine supports the integration into spring framework.

Table 5 describes in detail how our characterization scheme describes Camunda, but we highlight some aspects below:

Table 4
Evaluation of YAWL.

I. Modeling criteria	
I.1 Supported BP modeling languages	YAWL
I.2. Interoperability and compatibility	Import and export BPMN
I.3 Reuse BP models	Supported
I.4 Modeling views	All are supported
I.5 Modeling of business rules	Partially supported: It does not include BRM tool, but it supports XQuery expression and using Ripple-Down Rule (RDR) trees
I.6 Modeling of PPIs	Not supported
I.7 Generate process documentation	Not supported
II. Design criteria	
II.1 Supported programming languages	JAVA
II.2 Designing user interface (UI)	Automatic and manual are supported.
II.3 Way of describing roles	Both are supported
II.4 Support for importing organizational structure	Supported
II.5 Support for assigning a role to a user	Manual
II.6 Support for adding SLA and linking SLA KPIs	Not supported
II.7 Support exception handling and transaction control	Exception handling is supported, although Transaction Control is not supported
II.8 Translation into executable models	Automatic
II.9 Supported BP execution languages	YAWL
III. Deployment Criteria	
III.1 Support for distributed execution	Not supported
III.2 Support for integration with other systems and services	Web services and REST APIs
IV. Execution and Operation	
IV.1 Version management of BP models	Supported
IV.2 Support for calendar management	Supported
IV.3 Support for informing users	Partially supported: Only supports pull technique
IV.4 Document management	Supported
V. Monitoring and Control	
V.1 Support for technical monitoring and control	Not supported
V.2 Support for business monitoring BAM	Supported
V.3 Change the role or resource for an activity of process instance.	Supported
V.4 Support for changing business rules	Partially supported
V.5 Support for optimized execution	Partially supported
V.6 Ability to deal with failures	Partially supported
V.7 Support for changing the workload balance	Supported
V.8 Support dashboards and reports	Not supported
V.9 Support for detail levels	Not supported
V.10 Support for different views of monitoring	Partially supported
VI. Analysis criteria	
VI.1 Support for process verification	Supported
VI.2 Support for process simulation	Supported
VI.3 Support historical data available for analysis	Log files
VI.4 Support for suggestions on improvement	Not supported
VI.5 Support for BI and Process mining tools	Partially supported
VII. Other criteria	
VII.1 Documentation	PDF
VII.2 Training	It is more focused on the theory and scientific side, than on BPMS feature and its use
VII.3 Tool maturity	Since 2009 and has two main releases
VII.4 Commercial support	Available

- Modeling criteria: Camunda uses BPMN 2.0 as modeling language. It supports all modeling view perspectives, without including BRM tool and external tools like JBoss Drools (which is used to define business rules and decision tables). Camunda also supports mechanisms to define PPIs within BPMN events, but it does not support the generation of process documentation.
- Design criteria: Camunda supports Java and JavaScript as programming languages and it allows generating graphical user interfaces by users, both manually and automatically. In addition, it supports LDAP to manage organizational structures, exception handling and transaction control.
- Deployment Criteria: Distributed execution, REST API and Web service technology to integrate with other systems are supported.
- Execution and Operation: Version management is supported, but calendar management and document management are not fully supported. Moreover, push and pull technologies are supported in order to inform the user of their tasks.
- Monitoring and Control criteria: Business Activity Monitoring, changing the workload balance among users, dashboards and reports are all supported.
- Analysis criteria: Process model verification and simulation are supported, but suggestions on improvement are not supported.
- Other criteria: HTML documentation, training videos, examples, Webinars and community events are available by Camunda which first version was released in 2013. Besides, Camunda provides commercial support.

4.2.5. Activiti 5.17

Activiti is an open source light-weight workflow and BPM. Its core is a super-fast and rock-solid BPMN 2 process engine for Java, distributed under the Apache license. Activiti runs in any Java application, on a server, on a cluster or in the cloud. It consists of a modeler, designer, and kick-start for process modeling and Activiti engine to execute BP.

Table 6 describes in detail how our characterization scheme describes Activiti, but we enumerate some aspects below:

- Modeling criteria: Activiti uses BPMN 2.0 as modeling language (without including BRM tool) and it supports external tools like Drools rule engine. Modeling of PPIs and the generation of the

Table 5
Evaluation of Camunda.

I. Modeling criteria	
I.1 Supported BP modeling languages	BPMN 2.0
I.2. Interoperability and compatibility	Import and export BPMN 2.0.
I.3 Reuse BP models	Not supported.
I.4 Modeling views	All perspectives are supported
I.5 Modeling of business rules	Partially supported: It does not include BRM tool, but it supports external BRM tools, like JBoss Drools to define business rules and decision table, Techniques: Use expression language
I.6 Modeling of PPIs	Partially supported.
I.7 Generate process documentation	Not supported.
II. Design criteria	
II.1 Supported programming languages	Java, JavaScript
II.2 Designing user interface (UI)	Supports both
II.3 Way of describing roles	Names/Description
II.4 Support for importing organizational structure	Supports LDAP integration
II.5 Support for assigning a user to a role	Supported manually
II.6 Support for adding SLA and linking SLA KPIs	Not supported
II.7 Support exception handling and transaction control	Supported
II.8 Translation into executable models	Automatic
II.9 Supported BP execution languages	Not specified
III. Deployment Criteria	
III.1 Support for distributed execution	Not supported
III.2 Support for integration with other systems and services	REST and Web Services
IV. Execution and Operation	
IV.1 Version management of BP models	Supported
IV.2 Support for calendar management	Not directly supported, but it could be developed
IV.3 Support for informing users	Both are supported
IV.4 Document management	Not directly supported, but it could be developed
V. Monitoring and Control	
V.1 Support for technical monitoring and control	Not supported
V.2 Support for business monitoring BAM	Supported
V.3 Change the role or resource for an activity of process instance	Supported
V.4 Support for changing business rules	Not directly supported, but it could be developed
V.5 Support for optimized execution	Not supported
V.6 Ability to deal with failures	Supported for engine and activities
V.7 Support for changing the workload balance	Supported
V.8 Support dashboards and reports	Supported
V.9 Support for detail levels	Partially supported
V.10 Support for different views of monitoring	Partially supported
VI. Analysis criteria	
VI.1 Support for process verification	Supported
VI.2 Support for process simulation	Not supported
VI.3 Support historical data available for analysis	Log files available
VI.4 Support for suggestions on improvement	Not supported
VI.5 Support for BI and Process mining tools	Partially supported
VII. Other criteria	
VII.1 Documentation	Complete HTML documentation available
VII.2 Training	Video, examples, Webinars and events
VII.3 Tool Maturity	Since 2012, first version: 2013
VII.4 Commercial support	Supported

process documentation are not supported.

- Design criteria: Activiti supports Java and JavaScript as programming languages. In addition, it allows generating graphical user interfaces both automatically and manually. Exception handling and Transaction Control are supported.
- Deployment Criteria: It supports distributed execution, Web services and REST API to connect to other systems.
- Execution and Operation: Version management and calendar management are not supported, but it supports document management.
- Monitoring and Control criteria: Technical monitoring and control are not implemented in the community edition as well as BAM tool is not included in such edition. Moreover, changing the workload balance among users, dashboards and reports are supported.
- Analysis criteria: Model verification and simulation are supported.
- Other criteria: HTML documentation is available, but the community edition lacks official videos or Webinars. However, there are very active user forums. The first version was released in 2010.

4.2.6. *jBPM 6.2*

jBPM is a light-weight, fully open source (distributed under Apache license) written in Java. It is used to model, execute and monitor BPs throughout their lifecycle. jBPM 6.2 consists of a pure java BP engine that could be embedded to user application or deployed as a service, connected to it through Web-based user interfaces, remote APIs and human task services. jBPM allows modeling, simulating and deploying processes and other related artifacts (like process designer, data models, forms or business rules, among others) thanks to the jBPM's web-based management console.

Table 7 describes in detail the evaluation of jBPM on our characterization scheme, but we highlight some aspects below.

- Modeling criteria: jBPM supports BPMN 2.0 and allows reusing process models. It includes tools to define business rules and code constraints. For this purpose, jBPM implements techniques such as rule constraints and decision trees. Nevertheless, it does not support the modeling of PPIs or the generation of the process

Table 6
Evaluation of Activiti.

I. Modeling criteria	
I.1 Supported BP modeling languages	BPMN 2.0
I.2. Interoperability and compatibility	Import and export BPMN 2.0.
I.3 Reuse BP models	Not supported.
I.4 Modeling views	All are supported
I.5 Modeling of business rules	Partially supported: It has no native BRM tool, but it supports the use of Drools rule engine to execute business rules as well as another BRM. Techniques: Unified Expression Language.
I.6 Modeling of PPIs	Not supported.
I.7 Generate process documentation	Not supported.
II. Design criteria	
II.1 Supported programming languages	Java and JavaScript
II.2 Designing user interface (UI)	Both are supported
II.3 Way of describing roles	Name and Description
II.4 Support for importing organizational structure	Support LDAP integration
II.5 Support for assigning a role to a user	Manual
II.6 Support for adding SLA and linking SLA KPIs	Not supported
II.7 Support exception handling and transaction control	Both are supported
II.8 Translation into executable models	Automatic
II.9 Supported BP execution languages	Not specified
III. Deployment Criteria	
III.1 Support for distributed execution	Partially supported
III.2 Support for integration with other systems and services.	Web services and REST APIs
IV. Execution and Operation	
IV.1 Version management of BP models	Not supported
IV.2 Support for calendar management	Not supported
IV.3 Support for informing users	Partially supported. Pull is directly supported, but Push needs further development
IV.4 Document management	Supported
V. Monitoring and Control	
V.1 Support for technical monitoring and control	Not supported in the community edition
V.2 Support for business monitoring BAM	No BAM tool is included in the community edition
V.3 Change the role or resource for a process instance activity	Supported
V.4 Support for changing business rules	Not supported
V.5 Support for optimized execution	Not supported
V.6 Ability to deal with failures	Partial support for activity, communication using error events and exception handling
V.7 Support for changing the workload balance	Supported
V.8 Support dashboards and reports	Supported
V.9 Support for detail levels	Not supported
V.10 Support for different views of monitoring	Supported
VI. Analysis criteria	
VI.1 Support for process verification	Supported
VI.2 Support for process simulation	Supported
VI.3 Support historical data available for analysis	Partially supported: History DB and log files
VI.4 Support for suggestions on improvement	Not supported
VI.5 Support for BI and Process mining tools	Partially supported: Data sources for these tools are available such as event log or history databases.
VII. Other criteria	
VII.1 Documentation	HTML documentation available
VII.2 Training	There are no official videos or examples in the community edition, but there are very active and maintained forums. The community site does not mention Webinars or events
VII.3 Tool maturity	Since 2010
VII.4 Commercial support	Available by contracting enterprise edition

documentation.

- Design criteria: jBPM supports Java as a programming language and it allows the user to generate the user interface both automatically and manually. It supports integrations with LDAP, exception handling and transaction control.
- Deployment Criteria: jBPM supports a multi engine and multi nodes. Applications can connect the core engine through its Java API and also remotely by means of a REST and JMS API. It also supports Web service technology.
- Execution and Operation: Version management and document management are supported. Calendar management is not supported. Pull technology is supported in order to inform users of their tasks. Push technique could be implemented.
- Monitoring and Control criteria: Technical monitoring and BAM are not supported. Changing the role or resource for an activity of process instance is supported. The optimized execution according to some measurable criteria is not supported, whereas dashboards and reports are supported.
- Analysis criteria: Process verification and simulation are supported, but suggestions on improvement are not supported.
- Other criteria: jBPM provides HTML documentation, few training materials, events and Webinars. It has been available since 2003 and it offers commercial support.

4.2.7. uEngine BPM 3.6.0

uEngine is an open source BPMS that allows the complete management of end-to-end BPs. uEngine consists of three key components: uEngine BPM Foundation (process engine and process modeling tool), uEngine Process Portal (Dashboard and single sign on) and uEngine BP Analyzer (OLAP based process instance analyzer).

Table 7
Evaluation of jBPM.

I. Modeling criteria	
I.1 Supported BP modeling languages	BPMN 2.0
I.2. Interoperability and compatibility	Import and export BPMN 2.0.
I.3 Reuse BP models	Supported.
I.4 Modeling views	All supported
I.5 Modeling business rules	Partially supported: it uses Drools rule as a business rule tool, techniques: Java and MVEL code constraints, Rule constraints and Decision Tree
1.6 Modeling of PPIs	Not supported.
1.7 Generate process documentation	Not supported.
II. Design criteria	
II.1 Supported programming languages	Java
II.2 Designing user interface (UI)	Both are supported
II.3 Way of describing roles	Names/Description and by Capabilities
II.4 Support for importing organizational structure	LDAP is supported
II.5 Support for assigning a user to a role	Manual
II.6 Support for adding SLA and linking SLA KPIs	Not supported
II.7 Support exception handling and Transaction Control	Both are supported
II.8 Translation into executable models	Automatic
II.9 Supported BP execution languages	Not specified
III. Deployment Criteria	
III.1 Support for distributed execution	Supports multi engine and multi nodes
III.2 Support for integration with other systems and services.	Applications can connect to the core engine through its Java API or as a set of CDI services, and also remotely through a REST and JMS API. It also supports Web service technology.
IV. Execution and Operation	
IV.1 Version management of BP models	Supported
IV.2 Support for calendar management	Not supported
IV.3 Support for informing users	Partially supported; Pull is directly supported, although push could be implemented
IV.4 Document management	Supported
V. Monitoring and Control	
V.1 Support for technical monitoring and control	Not supported
V.2 Support for business monitoring BAM	Supported, History log available
V.3 Change the role or resource for an activity of process instance	Change role are supported
V.4 Support for changing business rules	Not supported
V.5 Support for optimized execution	Not supported
V.6 Ability to deal with failures	Provide support for engine and activities
V.7 Support for changing the workload balance	Supported
V.8 Support dashboards and reports	Supported
V.9 Support for detail levels	Supported
V.10 Support for different views of monitoring	Supported
VI. Analysis criteria	
VI.1 Support for process verification	Supported
VI.2 Support for process simulation	Supported
VI.3 Support historical data available for analysis	Supported, log available
VI.4 Support for suggestions on improvement	Not supported
VI.5 Support for BI and Process mining tools	Provide support to these tools in the history log
VII. Other criteria	
VII.1 Documentation	HTML documentation
VII.2 Training	It provides few training material, events and Webinars
VII.3 Tool Maturity	Since 2003
VII.4 Commercial support	Available

Table 8 describes in detail the evaluation of uEngine on our characterization scheme, but we highlight some aspects below:

- Modeling criteria: uEngine uses XPDL as a modeling language. It supports all the modeling perspectives, but the possibility to reuse the BP models is not supported, use ubrain as an integrated rule engine, the use of decision tables as a technique is not supported, it neither supports modeling of PPIs nor generating the process documentation.
- Design criteria: uEngine supports Java as a programming language and it also allows generating user interfaces both automatically and manually. It also supports exception handling as well as BPEL as business process execution language.
- Deployment Criteria: Distributed execution is not supported, but it supports Web Service technology to connect to other systems.
- Execution and Operation: Version management and document management are supported, although calendar management is not supported. Besides, it supports pull and push techniques to inform users of their tasks.
- Monitoring and Control criteria: Technical monitoring is not supported. However BAM, changing the role or resource for an activity of process instance, dashboards and reports are supported.
- Analysis criteria: Process verification and simulation, and suggestions on improvement are not supported, but it includes OLAP (On-Line Analytical Processing) tools.
- Other criteria: There is very little documentation available (most in Korean language) and poor training material (most in Korean language, too). It has been available since 2007. The commercial support is also available.

4.3. Analysis

This section processes and analyzes the evaluation data of each BPMS described in the previous section. Specifically, Table 9 combines

Table 8
Evaluation of uEngine.

I. Modeling criteria	
I.1 Supported BP modeling languages	XPDL
I.2. Interoperability and compatibility	Not supported
I.3 Reuse BP models	Not supported
I.4 Modeling views	All are supported
I.5 Modeling of business rules	Use ubrain -integrated- rule engine Technique: Decision table
1.6 Modeling of PPIs	Not supported
1.7 Generate process documentation	Not supported
II. Design criteria	
II.1 Supported programming languages	Java
II.2 Designing user interface (UI)	Both are supported
II.3 Way of describing roles	Name/Description
II.4 Support for importing organizational structure	Supported
II.5 Support for assigning a user to a role	Manual
II.6 Support for adding SLA and linking SLA KPIs	Not supported
II.7 Support exception handling and Transaction Control	Exception handling is supported, transaction Control is not supported
II.8 Translation into executable models	Automatic
II.9 Supported BP execution languages	BPEL
III. Deployment Criteria	
III.1 Support for distributed execution	Not supported
III.2 Support for integration with other systems and services	Web Services
IV. Execution and Operation	
IV.1 Version management of BP models	Supported
IV.2 Support for calendar management	Not supported
IV.3 Support for informing users	Partially supported; Pull is directly supported, but push could be implemented
IV.4 Document management	Supported
V. Monitoring and Control	
V.1 Support for technical monitoring and control	Not supported
V.2 Support for business monitoring BAM	Supported
V.3 Change the role or resource for an activity of process instance	Supported
V.4 Support for changing business rules	Not supported
V.5 Support for optimized execution	Not supported
V.6 Ability to deal with failures	Supported for the activities
V.7 Support for changing the workload balance	Supported
V.8 Support dashboards and reports	Supported
V.9 Support for detail levels	Not supported
V.10 Support for different views of monitoring	Supported
VI. Analysis criteria	
VI.1 Support for process verification	Not supported
VI.2 Support for process simulation	Not supported
VI.3 Support historical data available for analysis	History DB available
VI.4 Support for suggestions on improvement	Not supported
VI.5 Support for BI and Process mining tools	Include OLAP tools
VII. Other criteria	
VII.1 Documentation	Very few and out-dated documentation and most of it in Korean language
VII.2 Training	
VII.3 Tool Maturity	Since 2007
VII.4 Commercial support	Available

results obtained by applying the rating method defined in Section 3.3 to the evaluation result of each BPMS.

The results shown in Table 9 are obtained by applying the proposed methodology.

Table 9 summarizes evaluation results and helps assess each BPMS, as well as comparing these suites in terms of criterion, group of criteria and general level [7]. The rest of this section states some of these results and represents them in Fig. 3 according to BPM phases.

Noting results, it is possible to conclude that: YAWL and jBPM offer the best support level for modeling criteria and jBPM for design criteria; Camunda and jBPM provide the best support for deployment group of criteria; YAWL provides the best support for the execution level; uEngine offers the best support for Monitoring and Control levels; and Bonita, ProcessMaker and Camunda provide the best support for the “other” group of criteria. This evaluation grouped by criteria provides fast and sufficient information for organizations, so they can decide which BPMS best supports their particular requirements.

Table 9 also represents the total support level of every BPMS to sort evaluated systems according to their level of support for all evaluation criteria as follows, i.e., the best support is provided by jBPM, followed by Camunda, YAWL, Bonita, Activiti, ProcessMaker and uEngine.

Finally, more information could be derived from our study. For example, modeling PPIs and process documentation are less supported by BPMSs, and other aspects (e.g., Analysis criteria) need to be supported in an improved manner. This kind of information is helpful to understand customer demands and guides academic research.

5. Conclusions and future work

Today's world economic situation is ruled by issues such as globalization, which involves relocation of companies, constant search for lower costs to maximize profits by continuous mergers and acquisitions, and continuous motivation for improving quality and optimizing all BP. Thus, organizations should strengthen their mechanisms to be more competitive in this economic context.

In this context, at any organization, a proper business processes management can contribute to achieve the organizational objectives outlined above. In this way, organizations should implement their BP with mature management models based on tools in order to maintain and/or achieve an edge over their competitors.

BPM commonly states for business processes management. It identifies a management strategy and includes methods, techniques and tools to support the BP lifecycle, which includes design, enactment, management and analysis of operational BP. BPM is a consolidated strategy to reduce costs and improve BP (through a cycle of continuous improvement) in many organizations. In fact, adopting BPM can be grouped into three main needs: understanding and assimilating the organization's know-how; knowing the employees' performance during the execution processes; and finally, monitoring and measuring processes. Controlling these needs improves the ROI (Return on investment) parameter through reducing production costs. These objectives cannot be achieved by carrying out manual tasks because personnel costs and management can be very high and prohibitive.

However, there is a wide range of BPMS (both commercial and open source) to automate the process management. Each one has a wide assortment of prices and functionalities. In this context, the selection of a concrete solution may be quite a difficult, laborious and complex undertaking. A selection requires a complete analysis of the most popular available solutions. Otherwise, it may lead to choosing an inadequate workflow product which will not support efficiently BP in an organization.

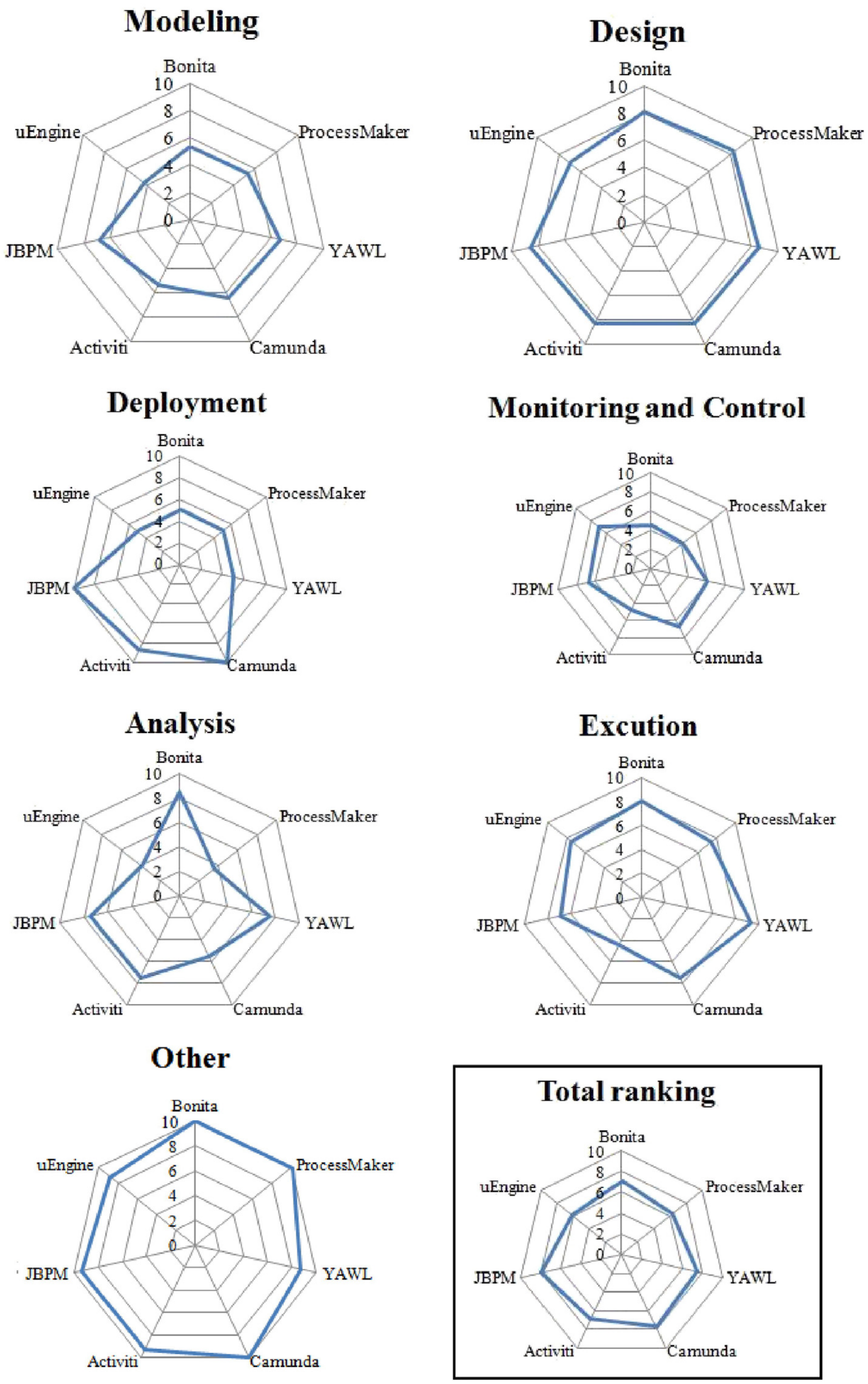
This paper shows a formal survey which combines well-known techniques such as SLR [23] and quality frameworks [7]. Thanks to this paper, organizations can formally plan, manage and perform a comparative on BPMSs and, later, analyze achieved results. This survey is also based on a quality model what is very useful for organizations and researchers because they can choose which BPMS better meets their requirements. It also allows researchers to identify the current characteristics of the existent open source BPMS, which could guide the future investigations.

Finally, after applying the evaluation method of our survey on open source BPMS, we could answer research questions mentioned in Section 3.1.1. Table 1 answers the first part of the RQ1 because it lists the open source BPMSs found and shows its fulfilment for our quality assessments, whereas Table 9 answers the second part of the RQ1.

Moreover, Sections 4.3 and 5 analyze the evaluation results to answer RQ2 and detail some areas for improvement. The evaluation methodology and its application results provide many conclusions for different stakeholders:

Table 9
Evaluation summary of all the BPMSs.

	Bonita	ProcessMaker	YAWL	Camunda	Activiti	jBPM	uEngine
I Modeling criteria: $PS_1 = \frac{\sum_{i=1}^{7} I_i}{28}\cdot 10$	5.36	5.36	6.79	6.43	5.36	6.79	4.29
I.1 Supported BP modeling languages	4	4	4	4	4	4	4
I.2. Interoperability and compatibility	4	0	4	4	4	4	0
I.3 Reuse BP models	0	4	4	0	0	4	0
I.4 Modeling views	4	4	4	4	4	4	4
I.5 Modeling of business rules	3	3	3	3	3	3	4
I.6 Modeling of PPIs	0	0	0	3	0	0	0
I.7 Generate process documentation	0	0	0	0	0	0	0
II Design criteria: $PS_2 = \frac{\sum_{i=1}^{9} II_i}{36}\cdot 10$	8.06	8.33	8.61	8.33	8.33	8.61	6.94
II.1 Supported programming languages	4	4	4	4	4	4	4
II.2 Designing user interface (UI)	4	4	4	4	4	4	4
II.3 Way of describing roles	3	4	4	3	3	4	3
II.4 Support for importing organizational structure	4	4	4	4	4	4	0
II.5 Support for assigning a role to a user	3	3	3	3	3	3	3
II.6 Support SLA	0	0	0	0	0	0	0
II.7 Support exception handling and transaction control	3	3	4	4	4	4	3
II.8 Translation into executable models	4	4	4	4	4	4	4
II.9 Supported BP execution languages	4	4	4	4	4	4	4
III Deployment Criteria: $PS_3 = \frac{\sum_{i=1}^{2} III_i}{8}\cdot 10$	5.00	5.00	5.00	10.00	8.75	10.00	5.00
III.1 Support for distributed execution	0	0	0	4	3	4	0
III.2 Support for integration	4	4	4	4	4	4	4
IV Execution & operation criteria: $PS_4 = \frac{\sum_{i=1}^{4} VI_i}{16}\cdot 10$	8.13	7.50	9.38	7.50	4.38	6.88	7.50
IV.1 Version management of BP models	4	0	4	4	0	4	4
IV.2 Support for calendar management	1	4	4	2	0	0	0
IV.3 Support for informing users	4	4	3	4	3	3	4
IV.4 Document management	4	4	4	2	4	4	4
V Monitoring & control criteria: $PS_5 = \frac{\sum_{i=1}^{10} V_i}{40}\cdot 10$	4.50	4.25	6.00	6.75	4.75	6.75	7.00
V.1 Support for technical monitoring	4	0	0	0	0	0	0
V.2 Support for business monitoring BAM	3	3	4	4	0	4	4
V.3 Change the role or resources	4	4	4	4	4	4	4
V.4 Support for changing business rules	0	0	3	2	0	0	0
V.5 Support for optimized execution	0	0	3	0	0	0	0
V.6 Ability to deal with failures	3	3	3	3	3	3	4
V.7 Support for changing the workload balance	4	4	4	4	4	4	4
V.8 Support dashboards and reports	0	3	0	4	4	4	8
V.9 Support for detail levels	0	0	0	3	0	4	0
V.10 Support for different views of monitoring	0	0	3	3	4	4	4
VI Analysis criteria: $PS_6 = \frac{\sum_{i=1}^{5} VI_i}{20}\cdot 10$	8.50	3.50	7.50	5.50	7.50	7.50	4.00
VI.1 Support for process verification	4	0	4	4	4	4	0
VI.2 Support for process simulation	4	0	4	0	4	4	0
VI.3 Support historical data	3	4	4	4	4	4	4
VI.4 Support for suggestions on improvement	3	0	0	0	0	0	0
VI.5 Support for BI and process mining tools	3	3	3	3	3	3	4
VII Other criteria: $PS_7 = \frac{\sum_{i=1}^{4} VII_i}{16}\cdot 10$	10.00	10.00	8.75	10.00	9.38	9.38	8.75
VII.1 Documentation	4	4	3	4	4	4	3
VII.2 Training	4	4	3	4	3	3	3
VII.3 Tool maturity	4	4	4	4	4	4	4
VII.4 Commercial support	4	4	4	4	4	4	4
Final score: $FS = \frac{\sum_{i=1}^{7} PS_i}{70}\cdot 10$	7.08	6.28	7.43	7.79	6.92	7.99	6.21



Modeling

Design

Deployment

Monitoring and Control

Analysis

Excution

Other

Total ranking

Fig. 3. Summary of the BPMS evaluation per BPM phase.

Organizations can use this evaluation to support their selection processes of the BPMS that best satisfy their requirements. For example, Table 9 and Fig. 3 detail the strengths and weaknesses of each BPMS in different levels (i.e. feature level, phase level, and general evaluation).

In addition, the quality model and the methodology proposed in this paper provide a starting point to perform customized evaluations (e.g., adding/removing criterion, phases, or evaluate other BPMSs).

BPMS providers could also use results of this study to compare their products with products of other providers in order to know its strength and weakness. This knowledge is important to guide their improvement plans.

Furthermore, our results highlight many improvement points, for instance: the definition and integration of the PPIs with the process model; linking of PPIs with service level agreements; the necessity to include native Business Rule Management; the generation of the process documentations and the support of suggestions when monitoring service discover an issue which affects the execution of the process (e.g. user overloading, bottlenecks, etc.), among others.

During the evaluation, we observed that every BPMS provider used its own terminology to describe BP concepts (e.g. join and fork elements, exception handling, events, etc.). Unifying the terminology could help to improve interoperability and portability between the BPMSs.

In addition, the evaluation shows BPMSs could be classified into two families. The first one is oriented to normal users (e.g. Bonita and ProcessMaker), and the second one is platform-oriented to developers and expert users (e.g. Activiti, Camunda and jBPM). This classification could help both organizations and providers. The first ones could select its best BPMSs and the second ones could focus on specific customers and their requirements.

Researchers also could benefit from this evaluation by focusing their investigation on improving the identified weakness points mentioned above (e.g. PPIs integration, the terminology, implementing Business Rule Management, improving the response of the BPMS when discovering possible issue during the execution of the process instances, etc.).

For future work, we plan to extend this evaluation criterion to include new groups of criteria such as security, cloud and mobile devices support, and social media integration, among others.

Future research will also focus on covering less supported criteria shown in our results. For example, it is interesting to research how to integrate PPI lifecycle (derive, model, design, verification, gathering and analysis) into BP lifecycle. In addition, process mining, business intelligent research and tools should be concerned in our future research.

Acknowledgments

This research has been supported by the MeGUS project (TIN2013-46928-C3-3-R) and by the SoftPLM Network (TIN2015-71938-REDT) of the Spanish Ministry of Economy and Competitiveness. This paper has also been supported by the Universia foundation thanks to a grant addressed to PhD students.

References

[1] E.M. Bahsi, E. Ceyhan, T. Kosar, Conditional workflow management: a survey and analysis, Sci. Program. 15 (4) (2007) 283–297.

[2] P. Brereton, B.A. Kitchenham, D. Budgen, M. Turner, M. Khalil, Lessons From Applying the Systematic Literature Review Process Within the Software Engineering Domain, JSS 80, (2007) 571–583.

[3] CMMI Official site: http://www.sei.cmu.edu/cmmi/. Carnegie Mellon University Software Engineering Institute. Retrieved July 2013.

[4] S. Craggs, Comparing BPM from Appian, Oracle and IBM - taking a high-level look at BPM solutions from three leading vendors, in: Lustratus Report, 2011.

[5] F.Q.B. Da Silva, A.L.M. Santos, S. Soares, A.C.C. França, C.V.F. Monteiro, F.F. Maciel, Six years of systematic literature reviews in software engineering: an updated tertiary study, Inf. Softw. Technol., vol. 53, Issue (9), September 2011, Pages 899–913, http://dx.doi.org/10.1016/j.infsof.2011.04.004, (ISSN 0950-5849).

[6] A Del-Río-Ortega, M Resinas A Ruiz-Cortés 2010. Defining process performance indicators: an ontological approach. On the Move to Meaningful Internet Systems: OTM 2010.

[7] F.J. Domínguez-Mayo, M.J. Escalona, M. Mejías, QuEF (Quality Evaluation Framework) for model-driven web methodologies. 10th International Conference on web engineering (ICWE 2010), LNCS 6385 (2010) 571–575.

[8] Escalona, M.J., García-García, J.A., Dominguez-Mayo, F.J., & Ramos, I. Technical tool surveys and comparative studies: a systematical approach. In V. Strahonja, N. Vrček, D. Plantak Vukovac, C. Barry, M. Lang, H. Linger, & C. Schneider (Eds.), Information Systems Development: Transforming Organisations and Society through Information Systems (ISD2014 Proceedings). Varaždin, Croatia: Faculty of Organization and Informatics. ISBN: 978-953-6071-43-2. 2014.

[9] M.J. Escalona, F.J. Domínguez-Mayo, J.A. García-García, N. Sánchez, J. Ponce, Evaluating enterprise content management tools in a real context, J. Softw. Eng. Appl. 8 (08) (2015) 431.

[10] Fuggetta, Software Process: A Roadmap. Proceeding ICSE '00 Proceedings of the Conference on The Future of Software Engineering, ACM 2000, pp. 25–34, http://dx.doi.org/10.1145/336512.336521.

[11] R. Garcs, T. de Jesus, J. Cardoso, P. Valente, Open Source Workflow Management Systems: A Concise Survey. 2009 BPM & Workflow Handbook, (2009) 333–346.

[12] GM Giaglis. 2001. A taxonomy of business process modeling and information systems modeling techniques. Int. J. Flex. Manuf. Syst. 2001, vol. 13, Issue (2), pp 209–228, Springer.

[13] Havey, M. Essential Business Process Modelling. O'Reilly Media, 1st ed., ISBN 13: 978-0596008437, 2005.

[14] J.B. Hill, J. Sinur, D. Flint, M.J. Melenovsky, Gartner's position on business process management, in: C.T. Stamford (Ed.), Business Issues Gartner, 2006.

[15] Hill, J.B., Cantara, M., Deitert, E. and Kerremans, M. Magic quadrant for business process management suites, Vol. ID No. G001252906, Gartner Research, Stamford, CT. 2007.

[17] D. Hollingsworth, The workflow reference model, in: Workflow Management Coalition, 1995.

[18] ISO/IEC, Software Engineering – Product Quality, 2001.

[19] ISO/IEC. ISO 9001:2008 Quality management systems—requirements. International Organization for Standardization, 2008.

[20] B.A. Kitchenham, Procedures for Performing Systematic Reviews. Keele University, Technical Report TR/SE-0401 and NICTA Technical Report 0400011T.1, 2004.

[21] B. Kitchenham, S. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering. Version 2.3., Department of Computer Science, University of Durham, Durham, UK, 2007 (EBSE2007-01).

[22] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey and S. Linkman, Systematic literature reviews in software engineering – a systematic literature review, Inf. Softw. Technol., Vol. 51, Issue (1), Pages 7–15, http://dx.doi.org/10.1016/j.infsof.2008.09.009, 2009, (ISSN 0950-5849).

[23] B. Kitchenham and P. Brereton, A systematic review of systematic review process research in software engineering, Inf. Softw. Technol., Vol. 55, Issue (12), December 2013, Pages 2049–2075, http://dx.doi.org/10.1016/j.infsof.2013.07.010, (ISSN 0950-5849).

[24] Laguna M., Marklund J. Business Process Modeling, Simulation and Design, Second Edn. Editor Chapman and Hall/CRC. ISBN 9781439885253. 2013.

[25] I. Markovic, A.C. Pereira, Towards a formal framework for reuse in business process modeling, in: I. Markovic, A.C. Pereira (Eds.), Business Process Management Workshops, Springer, 2008.

[26] S. Meilin, Y. Guangxin, X. Yong, W. Shangguang, Workflow management systems: a survey, in: International Conference on Communication Technology Proceedings, 1998. ICCT'98, IEEE, 1998, October, p. 6.

[27] M. Murray, Strategies for the successful implementation of workflow systems within healthcare: a cross case comparison, in: 36th Annual Hawaii International Conference on System Sciences (HICSS'03), IEEE Computer Society Washington, DC, USA, 2003.

[28] OASIS. Web Services Business Process Execution Language Version 2.0. Organization for the Advancement of Structured Information Standards. URL: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html. 2007.

[29] OMG. BPMN, Business Process Modelling Notation, Version 2.0. Object Management Grouphttp://www.omg.org/spec/BPMN/2.0/; 2011.

[30] OMG. UML2.5, Unified Modelling Language. Object Management Group, Formal/2012-10-24, 2012.

[31] PMI, Project Management Institute Inc. 2008. A Guide to the Project Management Body of Knowledge, Fourth Edition (PMBOK® Guide). ISBN13: 978-1933890517.

[32] C. Richardson, D. Miers, The Forrester Wave, BPM Suites, Q1 2013, in: Forrester's Report, 2013.

[33] Ryan K.L., Stephen S.G., Eng Wah Lee. Business process management (BPM) standards: a survey, Bus. Process. Manag. J., Vol. 15 Iss: (5) pp. 744–791. 2009.

[34] SEG (Software Enginnering Group), Guidelines for Performing Systematic Literature Reviews in Software Engineering Version 2.3., 2007.

[35] J. Sinur, W.R. Schulte, J.B. Hill, T. Jones, Magic Quadrant for Intelligent Business Process Management Suites. Gartner RAS Core Research G00224913, 2012.

[36] J. Sinur, W.R. Schulte, J.B. Hill, Magic Quadrant for Business Process Management Suites. Gartner RAS Core Research G00205212, 2010.

[37] H Smith, P Fingar 2003 [BOOK]. Business Process Management: The Third Wave ISBN 0929652339.

[38] SOB, Stationery Office Books. 2009. Managing Successful Projects with PRINCE2. Stationery Office Books. ISBN13: 978-0113310593. 2009.

[39] Stallman, M. Richard, Software libre para una sociedad libre, Ed. Tratantes de Sueños, Spain 2004, p. 99.

[40] K. Stoilova, T. Stoilo, Comparison of workflow software products, in: International Conference on Computer Systems and Technologies - CompSysTech'2006, Veliko Tarnovo, Bulgaria, 2006.

[41] K.P. Stoilova, T.A. Stoilov, Evolution of the workflow management systems, in: Scientific Conference on Information, Communication and Energy Systems and Technologies-ICEST, 2006, pp. 225–228.

[42] TDG, Open Source Software: Case Studies Examining Its Use, The Dravis Group, 2003.

[43] Tuomi, I. The future of open source. How Open is the Future. Marleen Wynants & Jan Cornelis (Eds), pp 429–459. ISBN 90-5487-378-7. 2005.

[44] W.M.P. Van der Aalst, A.H. ter Hofstede, et al., Business process management: a survey, in: Business Process Management, 2003, p. 1019.

[45] W.M.P. Van-der-Aalst, Business process management demystified: a tutorial on models, systems and standards for workflow management, Lect. Notes Comput. Sci 3098 (2004) 1–65, Lectures on Concurrency and Petri Nets.

[46] W.M.P. Van Der Aalst, A.H. Ter Hofstede, YAWL: yet another workflow language, Inf. Syst. 30 (4) (2005) 245–275.

[47] W.M.P. van der Aalst, Business process management: a comprehensive survey, in: ISRN Software Engineering, 2013.

[48] G. Von Krogh, S. Spaeth, K.R. Lakhani, Community, joining, and specialization in open source software innovation: a case study, Res. Policy 32 (7) (2003) 1217–1241, http://dx.doi.org/10.1016/S0048-7333(03)00050-7.

[49] W. Bandara, G.G. Gable, M. Rosemann, Factors and measures of business process modelling: model building through a multiple case study, Eur. J. Inf. Syst. 14 (2005) 347–360.

[50] M. Weske, W.M.P. van der Aalst, et al., Advances in business process management, Data Knowl. Eng. 50 (1) (2004) 1–8.

[51] C.Wohlin and R. Prikladnicki, Systematic literature reviews in software engineering, Inf. Softw. Technol., Volume 55, Issue 6, June 2013, pp. 919–920, http://dx.doi.org/10.1016/j.infsof.2013.02.002, (ISSN 0950-5849).

[52] H. Zhang, M.A. Babar, P. Tell, Identifying relevant studies in software engineering, Inf. Softw. Technol., Volume 53, Issue 6, June 2011, pp. 625–637, http://dx.doi.org/10.1016/j.infsof.2010.12.010, (ISSN 0950-5849).

[53] H. Zhang and M.A. Babar, Systematic reviews in software engineering: an empirical investigation, Inf. Softw. Technol., Vol. 55, Issue 7, July 2013, pp. 1341–1354, http://dx.doi.org/10.1016/j.infsof.2012.09.008, (ISSN 0950-5849).

[54] W.A. Shewhart, W.E. Deming (Eds.), Statistical Method from The Viewpoint of Quality Control, Dover Publications, New York, 1939/1986.