Functional Requirements for Business Rules Management Systems

Koen Smit
Digital Smart Services, HU University of Applied Sciences Utrecht, Utrecht, The Netherlands
koen.smit@hu.nl

Martijn Zoet
Optimizing Knowledge-Intensive Business Processes, Zuyd University of Applied Sciences, Sittard, The Netherlands
martijn.zoet@zuyd.nl

Matthijs Berkhout
Digital Smart Services, HU University of Applied Sciences Utrecht, Utrecht, The Netherlands
matthijs.berkhout@hu.nl

Abstract

Business Rules Management (BRM) is a method designed to transform legal requirements into executable business decisions and business logic. In the last few years, the BRM capabilities have been increasingly supported by a set of software technologies, which are bundled together in a so-called BRM-systems. The aim of this study is to develop a functional requirements themes for BRM solutions. To be able to do so, our data collection and analysis consisted of the collection and analysis of secondary data. With the collaboration of four Dutch governmental agencies, we collected 759 functional requirements with regards to BRM systems. Findings of our analysis show that several essential functional BRM themes emerge, which should be taken into account when selecting or constructing the actual BRM systems. Future research should focus on further validation of the functional requirement themes in both the governmental context as well as the context of commercial industries.

Keywords


Introduction

An organization’s performance depends upon its ability to manage its business decisions and business logic (Blenko, Mankins, & Rogers, 2010). To get a grip on business decisions and business logic, organizations apply a systematic and controlled approach to support the elicitation, design, specification, verification, validation, deployment, execution, governance, and monitoring of both, see Figure 1. The overall method to describe each step is defined as Business Rules Management (BRM), which is a combination of methods, techniques, and tools (Bajec & Krisper, 2005; Ross, 2003). The actual realization in of each capability depends on the type of business logic applied to define the business decisions. For example, to specify predictive models, different activities are executed in comparison when business rule statements are specified. In this research, the focus is on the latter.

More and more software systems to support one or more of the nine BRM capabilities in relationship to business rule statements has become available. Examples of software systems are IAM4, Cognitatie, DecisionFirst Modeler, BizzDesign, Trisotech, Usoft, Camunda, Avola, Pega Systems, Blueriq and Sapiens Decisions. Although the previously mentioned software systems are all labeled as business rules management systems, the actual functionality of each system differs. Previous research has focused on a...
classification of such systems based on a literature review of 166 articles (Liao, 2004). They define multiple categories like rule-based, knowledge-based, case-based reasoning, neural network systems and describe, on a high level, each type of system. Additionally, research has focused on defining the system architecture of business rules management systems as well as the application of business rules components in software architectures (Ly, Rinderle, & Dadam, 2008). In this paper, we adhere to the following definition of a business rule: “a statement that defines or constrains some aspect of the business intending to assert business structure or to control the behavior of the business” (Morgan, 2002).

![Fig. 1. BRM capability overview](image)

Figure 1. BRM capability overview

Current BRM research focused on technical implementations such as rule mining tools (Nelson, Peterson, Rariden, & Sen, 2010), different chaining mechanisms (backwards, forwards, hybrid) (Zoet, 2014), rule software architectures (Paschke & Bichler, 2008; Xiao & Greer, 2009), and the application of rules in software architectures (Ly et al., 2008; Min, Kim, Kim, Min, & Ku, 1996). This is also recognized by Schlosser, Baghi, Otto and Oesterle (2014), which describe that “companies are unsure about what they need to consider when dealing with BRM. Literature hardly provides answers to this question.” In their research, they create a process, functional architecture, and goal perspective, on a high abstraction level. This study extends the understanding of functional requirements, in the context of BRM, by exploring the required functional requirements for Business Rules Management Systems in more detail.

Similar to previous research, we consider the nine BRM capabilities as the foundation to define the functionalities for business rules management systems. In contrast to previous research, we do not analyze the business rules management systems from a meta-survey, academic literature or business literature perspective, but analyze end-user functional requirements for such systems. With this premise, the specific research question addressed is: “Which functional requirements should be taken into account with regards to the different capabilities as part of BRM?” We aim to answer this research question with the goal to add to the body of knowledge a framework of functional requirement themes that are derived from inductive research rather than deductive research. Additionally, we want to provide organizations, especially in the governmental context, with a collection of functional requirement themes that can guide the process of selection and development of BRM solutions.

The remainder of this paper is organized as follows: First, we provide insights into BRM capabilities and how functional requirements are related to those capabilities. This is followed by the research method used to derive the functional requirement themes. Furthermore, the collection and analysis of our research data, with regards to case study research and three rounds of coding, are described. Subsequently, our results which led to the collection of functional requirement themes are presented. Finally, we discuss which
conclusions can be drawn from our results, followed by a critical view of the research methods utilized and results of our study and propose possible directions for future research.

Background and Related Work

Business decisions and business logic are an important part of an organization’s daily activities. A business decision is defined as: “A conclusion that a business arrives at through business logic and which the business is interested in managing” (OMG, 2016b). Moreover, business logic is defined as: “a collection of business rules, business decision tables, or executable analytic models to make individual business decisions” (OMG, 2016a). To create added value with business decisions and business logic, several concepts are utilized in theory and practice. For example, a business vocabulary, fact models, a rulebook, and rule requirements (Bajec & Krisper, 2005). However, as our focus in this paper is not to define these different concepts that are utilized in a variety of ways by organizations, we adhere to these concepts as artifacts in a general sense. See, for a detailed description of each of the concepts to design, specify, and execute business decisions and business logic in the work of (Smit & Zoet, 2016).

As stated in the previous section, BRM consists of nine capabilities. In this paper, a capability is defined as an ability that an organization, person, or system, possesses (OMG, 2016a). A detailed explanation of each capability can be found in (Smit & Zoet, 2016). However, to ground our research, a summary of the elicitation, design, specification, verification, deployment, execution, governance and monitoring capabilities is provided here, see also Figure 1.

The purpose of the elicitation capability is twofold. First, the purpose is to determine the knowledge that needs to be captured from various legal sources to realize the value proposition of the business rules (Graham, 2007). Different types of legal sources from which knowledge can be derived are, for example, laws, regulations, policies, internal documentation, guidance documents, parliament documents, official disclosures, implementation instructions, and experts. Depending on the type of knowledge source(s), for example, documentation versus experts, different methods, processes, techniques and tools to extract the knowledge are applied (Liao, 2004). The second purpose is to conduct an impact analysis is if a business rule architecture is already in place. When all relevant knowledge is captured, the business decisions need to be designed in the design capability. The purpose of the design capability is to establish a business rules architecture, which contains the business decisions and how the business decisions are derived to deliver the value proposition (Von Halle & Goldberg, 2009). After the business rule architecture is designed, the contents of the business decisions need to be specified in the specification capability. The purpose of the specification capability is to write the business logic and create the fact types needed to define or constrain some particular aspect of the business. After the business logic is created, it is verified and validated. The purpose of the verification capability is to determine if the business logic adheres to predefined criteria and are logically consistent (to check for semantic / syntax errors). When no verification errors are identified, the created value proposition is reviewed in the validation capability. The purpose of the validation capability is to determine whether the verified value proposition holds to its intended behavior (Zoet & Versendaal, 2013). When no validation errors are identified the context is approved and marked for deployment. The purpose of the deployment capability is to transform the verified and validated value proposition, which is formulated in an implementation-independent language, to implementation-dependent executable business decisions and business logic. An implementation-independent language is defined as: “a language that complies with a certain level of naturalness but has a delimited predefined expressiveness and is not tailored to be applicable to a specific automated information system” (Zoet & Versendaal, 2013). In contrast, an implementation-dependent language is defined as: “a language that complies with a specific software formalism, has a delimited predefined expressiveness and is tailored to be interpreted by a particular information system” (Zoet & Versendaal, 2013). However, this does not necessarily imply that the actor that utilizes the value proposition is a system, as the value proposition could also be used by subject-matter experts (Zoet, 2014). An implementation-dependent value proposition can be source code, handbooks or procedures (Morgan, 2002). The output of the deployment capability is then executed in the execution capability, which delivers the actual value proposition. To realize the added value, human or information system actors execute the business decisions and business logic. Overall, covering the full range of capabilities described earlier, two more capabilities are of importance; governance and monitoring. The governance capability consists of three sub-captibilities; version management, traceability management, and validity management (Boyer & Mili, 2011; Morgan, 2002; Smit, Zoet, & Berkhout, 2016).
The goal of the version management capability is to capture and keep track of version data regarding the elements created or modified in the elicitation, design, specification, verification, validation, deployment and execution capabilities. The traceability management capability is utilized to create relationships between specific versions of elements used in the value proposition. The goal of the traceability management capability is to make it possible to trace created elements, as parts of the value proposition, to the corresponding laws and regulations on which they are based. Another goal of the traceability management capability is the foundation it forms for impact analysis when new or existing laws and regulations need to be processed into the value proposition. The third sub-capability comprises validity management. The goal of validity management is to be able to provide, at any given time, a specific version of a value proposition. Lastly, the monitoring capability observes, checks and keeps record of not only the execution of the value proposition but also the full range of activities in the previously explained BRM capabilities that are conducted to realize the value proposition. The goal of the monitoring capability is to provide insights into how the BRM capabilities perform and, additionally, suggest improvements (Bajec & Krisper, 2005). To realize the summarized capabilities, functionalities are needed that support the actual execution of the capabilities.

A method to formulate functionalities in software engineering is requirements engineering. Requirements engineering, in general, is a systematic approach to specifying requirements and consists of four stages 1) requirements elicitation, 2) requirements analysis, 3) requirements specification, and 4) requirements validation (Kotonya & Sommerville, 1998). Different types of requirements exist, for example, functional requirements, non-functional requirements, and constraints (Sommerville & Sawyer, 1997). In this paper, we solely focus on functional requirements with regards to BRM systems as a functional requirement emphasizes what is required, and not how. This is in line with the notion of a capability, which also focuses on what (value) an organization can deliver, but not how the value is delivered.

Different methods to formulate functional requirements exist, for example, use cases, personas, mockups, wireframing, user stories (Schön, Thomaschewski & Escalona, 2017). The latter is increasingly being adopted. User stories are comprehensible by, for example, both developers and customers and support participatory design by all stakeholders as they are all able to design the behavior of the system. The agile community, in addition to user stories, also distinguish epic’s and themes. An epic is a large user story while a theme is a collection of user stories. Furthermore, the utilization of user stories enables empirical design by enabling the designers to make decisions by studying prospective users in typical situations (Cohn, 2004). The organizations analyzed all defined their functional requirements by means of user stories. Therefore, in our study, the unit of analysis is user stories.

**Research Method**

The goal of this research is to identify BRM functional requirement themes for the development of BRM solutions in the governmental context. To be able to do so, qualitative research is chosen as our research methodology. To instantiate this, case study research is identified as the most suitable strategy for this research.

Case study research is selected so that the researchers were able to gather functional requirements for BRM solutions in the Dutch governmental context. Therefore, the case studies are exploratory of nature. The organizations are selected from a pool of Dutch governmental institutions that provide public administration services based on laws and regulations that are provided by the Dutch legislative governmental branches. Our study comprised a holistic case study approach, see also the work of (Yin, 2013), featuring one context, the BRM solutions requirements phase, and four cases within this context. The unit of analysis is the BRM solution-related set of functional requirements of the participated organizations. The data collection consisted of secondary data, which is a form of third-degree data collection. According to Runeson & Höst (2009), third-degree data collection is specifically suitable when data such as requirements specification documents are studied, which is the case in our study.

**Data Collection and Analysis**

Data for this study was collected over a period of three months, between November 2016 and January 2017, through case studies at four organizations. The selection of the participants should be based on the group of individuals, organizations, information technology, or community that best represents the phenomenon
studied (Strauss & Corbin, 1990). In the context of this study, this means that the phenomenon studied is represented by organizations and individuals within these organizations which deal with the formulation or collection of BRM solutions-related requirements. Such organizations are often financial and government institutions. As stated previously, several Dutch governmental agencies were invited to collaborate in this study. All invited Dutch governmental agencies have in common that they are all executive governmental branches of the Dutch government. These type of governmental organizations are responsible for the execution of a large variety of services like the application, assessment, and notification regarding benefits, subsidiaries, visa’s, permits, tax returns, vouchers, loans, grants, screenings, etc. Combined, the participated organizations serve approximately 17 million citizens and organizations in the Netherlands. The governmental agencies are similar in nature in terms of business processes and how law and regulations must be implemented, which is imposed by legislative governmental branches. Due to requests of the participated organizations to be reported anonymously, the four governmental agencies that participated in this research are, from here on, labeled as organization A, B, C, and D. The four participated organizations were invited to gather and send all their BRM solutions-related requirements documents to the research team. The BRM solutions-related requirements were defined by teams within the organizations. Each team minimally existed out of an enterprise architect, business rules analyst, policy or legal expert. Additionally, per individual organization the teams were supported by a Procurement Officer, BRM project manager, business consultant, IT architect and/or external advisors.

Based on the data received, the research team analyzed and structured the functional requirements. As stated in the background and related work section, the functional requirements were already expressed in a user story format, in the form of natural language (in Dutch). To the knowledge of the authors the participated organizations employ the format of user stories as it allows them to work with functional requirements in a practical way, also due to the fact that all stakeholders can understand the functional requirements. The data analysis was conducted in three cycles of coding, following Strauss and Corbin’s process of 1) open coding, 2) axial coding, and 3) selective coding (Strauss & Corbin, 1990). The first coding cycle, open coding, consisted of the identification of functional requirements from the secondary data and the registration of meta-data with regards to the functional requirements. Each functional requirement was already registered as a user story, but to ensure optimal analysis we numbered each user story with a unique ID. Furthermore, for each functional requirement we registered their responsible role or owner, feature (what does the role exactly wants with the functionality), feature outcome (what is the benefit of using the particular functionality), organization, and organization’s ID (for traceability of the functional requirement towards the original documentation of the case organizations). In this process, two situations occurred: 1) The functional requirements were explicitly documented and could be documented by registering the organization and organization ID, see for example Table 1, or 2) the functional requirements were implicitly stated in other functional requirements (nested requirements) or plain text. A simplified example of a nested requirement is as follows: “I want to be able to import a source, via MS Word, MS Excel, but also in XML format and from the eur-lex.europa.eu platform.” This particular requirement is not properly normalized and actually consists of four individual functional requirements.

The open coding was followed by axial coding. The axial coding round was applied to structure the functional requirements to the BRM capabilities as proposed by (Smit & Zoet, 2016): the elicitation, design, specification, verification, validation, deployment, execution, monitoring, and governance, which is the coding scheme in this round. Furthermore, the category overall was added to ensure that all functional requirements that could not be assigned to the existing BRM capabilities or where applicable to all capabilities could be coded as well. For example, see Table 1, where both functional requirements, which were identified in the previous coding round, focus on the elicitation of knowledge from sources. Therefore, both functional requirements were coded as elicitation functional requirements.

<table>
<thead>
<tr>
<th>ID</th>
<th>Role</th>
<th>Feature</th>
<th>Outcome</th>
<th>Organization</th>
<th>Organization reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>44</td>
<td>Policy advisor</td>
<td>I want to be able to select text to link sources.</td>
<td>So that I am able to target parts of text that are decisions</td>
<td>B</td>
<td>PR13_UR_A_24</td>
</tr>
<tr>
<td>67</td>
<td>BR analyst</td>
<td>I want to be able to create blocks of text from law.</td>
<td>So that I am able to select and store artifacts</td>
<td>C</td>
<td>BLIKZT-1864</td>
</tr>
</tbody>
</table>
Table 1. Example functional requirements from the data set

Lastly, the third round of coding, selective coding, was conducted. Selective coding consisted of the identification of themes within the selection of functional requirements which were assigned to the BRM capabilities in the axial coding. As both functional requirements in Table 1 describe the annotation of sources, we coded them as *annotate sources*.

### Results

In this section, we present the results of our data collection through the presentation of the BRM functional requirement themes. As described in the previous section, three rounds of coding were conducted. We first provide the descriptive statistics with regards to the results of our coding processes, which is followed by the description of the derived functional requirement themes. The first round of coding, open coding, resulted in the registration of 759 unique functional requirements, which originate from four organizations, see Table 2.

<table>
<thead>
<tr>
<th>Organization</th>
<th>Total number of functional requirements identified</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>241</td>
</tr>
<tr>
<td>B</td>
<td>169</td>
</tr>
<tr>
<td>C</td>
<td>146</td>
</tr>
<tr>
<td>D</td>
<td>203</td>
</tr>
</tbody>
</table>

Table 2. Breakdown of functional requirements received from the case organizations

From this sample, 224 functional requirements (29.5%) were identified and registered as nested functional requirements in the data set, see for example the nested functional requirement provided in the previous section. Subsequently, the second round of coding consisted of the assignment of the functional requirements to ten BRM capabilities as described in the previous section. The results of the second round of coding (in amounts and percentage of the total sample of requirements) are presented in Table 3.

<table>
<thead>
<tr>
<th>Capability/Case</th>
<th>Organization A</th>
<th>Organization B</th>
<th>Organization C</th>
<th>Organization D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elicitation</td>
<td>12</td>
<td>42</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>Design</td>
<td>52</td>
<td>14</td>
<td>25</td>
<td>8</td>
</tr>
<tr>
<td>Specification</td>
<td>62</td>
<td>67</td>
<td>22</td>
<td>122</td>
</tr>
<tr>
<td>Verification</td>
<td>20</td>
<td>10</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Validation</td>
<td>13</td>
<td>4</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>Deployment</td>
<td>7</td>
<td>1</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>Execution</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Governance</td>
<td>24</td>
<td>5</td>
<td>42</td>
<td>12</td>
</tr>
<tr>
<td>Monitoring</td>
<td>1</td>
<td>11</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>50</td>
<td>15</td>
<td>24</td>
<td>41</td>
</tr>
</tbody>
</table>

Table 3. Breakdown of functional requirements per BRM capability

The third round of coding resulted in the identification of 37 functional requirement themes, see Figure 2. Due to space constraints, the themes derived from our data collection and analysis are presented, per capability, by their name and briefly described upon.

### Elicitation themes

The knowledge needed to create business decisions and business logic is elicited from a variety of different sources, i.e. laws, regulations, policies, internal documentation, guidance documents, parliament documents, expert hearings, implementation instructions, and official disclosures. Each source has a different format and is published by a different organization. Therefore, the functionality to support the *import sources* is one of the main themes within the elicitation capability, either manually or automatically. Additionally, the theme *annotate sources* was identified, which concerns either the manual, but preferably automatic annotation of sources to artifacts used to create business decisions and business logic, i.e. derivation structures, terms, or roles. When all sources are imported and classified, the user must be able to *generate overviews* of (related) sources in order to determine the business decision of the added value that has to be created. Additionally, it is deemed useful that important information, i.e. interpretations,
design decisions, and tips, can be captured with regards to the sources. Lastly, the possibility to perform impact-analysis is addressed, as it allows the user to determine the impact of modified sources with regards to already implemented business decisions and business logic.

**Figure 2. Framework of functional requirement themes**

**Design themes**

When the relevant knowledge is delivered from the elicitation capability, the user needs to analyze the knowledge in order to create business decisions. Additionally, when business decisions are created, the user needs to be able to create relationships to the different business decisions by linking them to each other and specify the relationship type between them. The cohesion between business decisions can be depicted in diagrams that provide the user with the possibility to create overviews of the linked business decisions. Lastly, one important theme was the ability to reuse business decisions and structures of business decisions, which enable users to work more efficiently by reusing existing artifacts or use them as design templates.

**Specification themes**

Based on the business decisions created and related to each other in the design capability, the user needs to be able to define business logic per business decision. Additionally, the possibility to add meta-data to artifacts created in the specification capability is deemed important. For example, with regards to business rules, users want to be able to add traceability links or validity dates. Also, all artifacts need to be related to each other in the specification capability. Therefore, create relationships is of importance as it supports the user in the specification processes as well as that it forms a fundamental basis for impact-analysis and traceability over all business logic artifacts.

**Verification themes**

Verification is related to the creation and modification of artifacts in the elicitation, design and specification activities. In the data, it was found important that a user is able to perform verification on request, which is classified as a detective form of verification, see (Smit, Versendaal, & Zoet, 2017). However, the system also must be able to perform verification in a preventive manner so that it is nearly impossible to implement errors in the business decisions and business logic, see (Smit et al., 2017). According to the data, both forms of verification must be available in the elicitation, design and specification processes. For example, verification in the context of the specification of business logic means that the quality of business rules is controlled by checking and notifying users of errors or enforcing certain business rule patterns, for example, see (Smit et al., 2017). Lastly, to incorporate changes into the verification functionality, a user must be able to customize the meta-model.
Validation themes

As part of validation, a user must be able to mark for validation, when an artifact succeeded the verification processes. Furthermore, validation must be supported by enabling a user to perform validation, where the user is able to test all relevant combinations of artifacts, i.e. business rules, fact types, and fact values. An important theme within validation is the dependence of cohesion between all the business decisions and business logic artifacts that need to be validated, thus requiring the possibility to create overviews to support validation of artifacts. Moreover, the system should assist the user by the automatic generation of test cases of the artifacts that need to be validated. Based on the results of the validation, a user (manually) or the system (automatically) must be able to generate a review report.

Deployment themes

When the new or modified business decisions and business logic passed both the verification and validation processes, a user marks both for deployment. Business decisions and business logic can be transformed into various type implementation-dependent languages, i.e. in different information systems, but also into work instructions, manuals, and procedures (Smit & Zoet, 2016). An important theme with regards to deployment is the possibility to deploy artifacts. With regards to the deployment of artifacts, a user must be able to deploy an individual artifact as well as a collection of artifacts. Furthermore, a user must be able to add meta-data (i.e. version number, user, date of deployment, and validity range) of deployed artifacts. However, this could also be performed in an automatic manner by the system, which is preferred as it eliminates manual input errors by users. Lastly, the publication of business decisions and business knowledge is an important theme. All Dutch governmental organizations are forced by law to provide transparency with regards to how a decision is made. Additionally, Dutch organizations and other stakeholders are dependent on the publications of the Dutch governmental agencies as well. However, it is not always desirable to publish all information regarding business decisions and business logic. In addition, the ability to publish selected sections of business decisions and business logic is desired within this context.

Execution themes

No specific themes were identified for the execution capability.

Governance themes

The governance capability is one of two overarching BRM capabilities that supports all capabilities, with the exception of the monitoring capability. Governance exists of three sub-capabilities; traceability management, version management, and validity management. With regards to traceability management, a user must be able to select an artifact and examine traces of the artifact in a backward and forward direction. For example, a business rule is part of a business rule set, which is part of a business decision that is based upon a collection of sources (backward direction). However, the same business rule is used in implemented products or services, in the form of a letter, source code and user instruction (forward direction). Additionally, a user wants to be able to create traces between artifacts.

With regards to version management, the system must log CRUD-activities from a user, accompanied with a timestamp, preferred in an automatic manner. For example, when an individual removed version 2.1 of a certain business rule and created a subsequent version, or where the system must log status changes of an artifact (i.e. in progress, to be reviewed, to be deployed). Furthermore, a user must be able to, for all artifacts, view version history, and be able to retrieve (previous) version. Also, a user must be able to manage version statuses known by the system (i.e. add an extra status or modify the label of a status). Lastly, to facilitate effective collaboration between users, a specific user must be able to check-out artifact so that other users cannot work on the same artifact.

With regards to validity management, a user must be able to define validity variables of an artifact (i.e. this business decision’s validity period starts at 03-01-2017 and ends at 08-12-2017). A user must be able to perform this manually, but preferably it is supported by the system in an automated manner as it should be able to analyze the sources in the elicitation processes. However, a user must always have the possibility to override the validity data derived from a source by the system.
Monitoring themes

Monitoring is one of two overarching BRM capabilities and is applied with regards to each of the other eight capabilities to support users with various activities. For example, within the context of verification, a user wants to examine the number of verification errors identified in a given time period. A similar example holds for validation, where a user wants to examine the amount of rejected artifacts that did not meet the criteria and needed further reiteration in the elicitation, design or specification processes. With regards to the monitoring of all the eight capabilities, a user wants to be able to execute standard reports as well as be able to define included variables of a report in a manual way. To boost the effectiveness of searching for specific information in reports, a user must be able to apply filters, either standard or self-defined (i.e. creation date or updates per artifact). Furthermore, a user must be able to apply sorting options. Lastly, it is deemed important that all reports can be stored in a wide variety of common formats (i.e. .csv, .xlsx, .pdf, and .docx).

Discussion and Conclusion

The goal of this research is to identify BRM functional requirement themes for the development of BRM solutions in the governmental context. To be able to do so, we addressed the following research question: “Which functional requirements should be taken into account with regards to the different capabilities as part of BRM?” In order to answer this question, we utilized case study research and conducted three rounds of coding, involving 759 functional requirements specified by four large Dutch governmental agencies. From a research perspective, our study provides a fundament for the development of functional requirements and situational factors regarding the application of such requirements. From a practical perspective, organizations, especially in the governmental context, could benefit from the presented BRM functional requirement themes that guide the process of selection and development of BRM solutions.

Several limitations may affect our results. The first limitation is the sampling and sample size. The sample group of case organizations is drawn from organizations only in the governmental domain. While we believe that government institutions are representative for organizations selecting and applying BRM solutions, further generalization towards non-governmental organizations is recommended. Furthermore, our sample size of 759 functional requirements from four organizations is limited, however, appropriate for research studies at the current maturity stage of the BRM domain. With regards to the sample, examination of the coverage statistics presented in Table 3 shows that there is an anomaly between the number of functional requirements per capability, per participated organization. This phenomena is likely caused by the different role compositions per team of individuals at each organization. Following this, we recommend future studies to incorporate larger amounts of functional requirements, preferably from a mix of different industries to further validate the current set of functional requirement categories as well as to compare between different industries with the goal to provide situational sets of functional requirements. Lastly, in this study, we solely take into account the functional requirements related to BRM solutions. While this scope is appropriate for this particular study, we believe that the focus of future studies should also be on the inclusion of, amongst other types of requirements, non-functional requirements in the context of BRM solutions.

REFERENCES

Functional Requirements for BRM-Systems


