Data Quality in Big Data Applications - Derivation of a Taxonomy based on Standards and Literature Review

MASTERTHESIS

To attain the academic title „Master of Science“ in the degree course Information Systems

Submitted to
Univ.-Prof.
Dr. Ruth Breu
Department of Computer Science
Faculty of Mathematics, Computer Science and Physics
at the Leopold-Franzens-University of Innsbruck

by
Bernhard Brugger

Innsbruck, April 2019
Abstract

Big Data is currently in the spotlight of research and industry. It has its roots in technological innovations, like Internet of Things (IoT), which facilitate novel capabilities for knowledge creation through the provision of unprecedented amounts of data. The intrinsic value that is hidden in the torrent of data has already caught the attention of many industries including Finance, Retail, or Healthcare. Those companies use Big Data applications to enable the extraction of decisive information that create deep understanding for the purpose at-hand, resulting in competitive edge. Whilst efforts in advancing technological capabilities seem to be highly focused in research and industry, however, the quality of these huge data masses, on which decisions are based, is often neglected. Generally, research on data quality is firmly established since decades. The problem is that focus has changed and Big Data dares existing quality practices with new challenges embodied by its key-characteristics Volume, Variety, Velocity and Veracity. Satisfying those characteristics requires a broader focus, which means, that Big Data quality is not just about data itself. Rather more it affects data as well as system design, like storage, processing, format, transformation, and others. Due to these peculiarities, research on Big Data quality is just in its infancy and craves for attention. Up to date, there is no comprehensive classification on Big Data quality that focuses especially on design quality characteristics of Big Data applications. To fill this gap and to provide a starting point for future research, this work presents the first taxonomy for data quality in Big Data applications. Based on an in-depth review of standards, scientific as well as business literature, the proposed taxonomy includes 19 quality design attributes, assorted into the five quality dimensions Usability, Security, Privacy, Persistence, and Validity. Supplementary, for the purpose of thoroughness, the classification also introduces inherent quality dimensions of Big Data. The taxonomy is evaluated on two distinct Big Data applications using a Goal-Question-Metric (GQM-) approach and corresponding visualizations. Results from the evaluation prove the value of the proposed taxonomy by illustrating contextual quality profiles of distinct Big Data applications.

Principle of equality

For better readability, this work does not apply any gender-specific formulations. However, if in any case the masculine form is used, both genders are addressed.
# Table of Contents

List of Abbreviations ........................................................................................................ IV
List of Figures ...................................................................................................................... V
List of Tables ....................................................................................................................... VI
1. Introduction ...................................................................................................................... 1
   1.1. Motivation ................................................................................................................... 1
   1.2. Scope and Boundaries ................................................................................................. 4
      1.2.1. Design-Science Research ................................................................................... 4
      1.2.2. Taxonomy based on Literature Review ............................................................... 5
      1.2.3. Criteria Catalogue using a Goal-Question-Metric (GQM-) Approach ............... 6
      1.2.4. Evaluation Case Specifications ........................................................................... 7
   1.3. Research Question ...................................................................................................... 8
   1.4. Thesis Structure ......................................................................................................... 8
2. Theoretical Background .................................................................................................. 10
   2.1. The Concept of Big Data ............................................................................................ 10
      2.1.1. Definition of Big Data ......................................................................................... 11
      2.1.2. Big Data Analytics .............................................................................................. 13
      2.1.3. Big Data Applications .......................................................................................... 14
   2.2. The Concept of Quality ............................................................................................. 15
      2.2.1. Definition of Data Quality ................................................................................... 15
      2.2.2. Data Quality Perspectives .................................................................................. 16
3. Research Method ............................................................................................................. 20
   3.1. Material Sources ........................................................................................................ 20
   3.2. Search Strategy ......................................................................................................... 21
   3.3. Inclusion and Exclusion Criteria ............................................................................... 23
   3.4. Data Extraction and Classification ........................................................................... 23
4. Results ............................................................................................................................. 26
   4.1. Quality of Conformance ............................................................................................ 28
   4.2. Usability .................................................................................................................... 30
      4.2.1. Contextual-Requirements Completeness ............................................................. 30
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.2.</td>
<td>Metadata Provision</td>
<td>33</td>
</tr>
<tr>
<td>4.2.3.</td>
<td>Data Fitness</td>
<td>34</td>
</tr>
<tr>
<td>4.2.4.</td>
<td>User Acceptance</td>
<td>36</td>
</tr>
<tr>
<td>4.3.</td>
<td>Security</td>
<td>38</td>
</tr>
<tr>
<td>4.3.1.</td>
<td>Confidentiality</td>
<td>38</td>
</tr>
<tr>
<td>4.3.2.</td>
<td>Integrity</td>
<td>40</td>
</tr>
<tr>
<td>4.3.3.</td>
<td>Availability and System Immunity</td>
<td>42</td>
</tr>
<tr>
<td>4.3.4.</td>
<td>Accountability</td>
<td>44</td>
</tr>
<tr>
<td>4.4.</td>
<td>Privacy</td>
<td>46</td>
</tr>
<tr>
<td>4.4.1.</td>
<td>Non-disclosure</td>
<td>46</td>
</tr>
<tr>
<td>4.4.2.</td>
<td>Pre-serving</td>
<td>47</td>
</tr>
<tr>
<td>4.4.3.</td>
<td>Provenance</td>
<td>49</td>
</tr>
<tr>
<td>4.4.4.</td>
<td>Conformance</td>
<td>50</td>
</tr>
<tr>
<td>4.4.5.</td>
<td>Compliance</td>
<td>52</td>
</tr>
<tr>
<td>4.5.</td>
<td>Persistence</td>
<td>54</td>
</tr>
<tr>
<td>4.5.1.</td>
<td>Scalability</td>
<td>54</td>
</tr>
<tr>
<td>4.5.2.</td>
<td>Automation</td>
<td>56</td>
</tr>
<tr>
<td>4.5.3.</td>
<td>Interoperability</td>
<td>58</td>
</tr>
<tr>
<td>4.5.4.</td>
<td>Extensibility</td>
<td>61</td>
</tr>
<tr>
<td>4.5.5.</td>
<td>Agility/Flexibility</td>
<td>63</td>
</tr>
<tr>
<td>4.6.</td>
<td>Validity</td>
<td>65</td>
</tr>
<tr>
<td>4.6.1.</td>
<td>Requirements Validity</td>
<td>65</td>
</tr>
<tr>
<td>4.7.</td>
<td>Discussion</td>
<td>68</td>
</tr>
<tr>
<td>5.</td>
<td>Criteria Catalogue</td>
<td>69</td>
</tr>
<tr>
<td>5.1.</td>
<td>Basic Structure</td>
<td>69</td>
</tr>
<tr>
<td>5.2.</td>
<td>Calculation</td>
<td>69</td>
</tr>
<tr>
<td>5.3.</td>
<td>Visualization Tool</td>
<td>71</td>
</tr>
<tr>
<td>6.</td>
<td>Evaluation</td>
<td>73</td>
</tr>
<tr>
<td>6.1.</td>
<td>Usability</td>
<td>73</td>
</tr>
<tr>
<td>6.2.</td>
<td>Security</td>
<td>76</td>
</tr>
<tr>
<td>6.3.</td>
<td>Privacy</td>
<td>78</td>
</tr>
<tr>
<td>6.4.</td>
<td>Persistence</td>
<td>80</td>
</tr>
<tr>
<td>6.5.</td>
<td>Validity</td>
<td>83</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>6.6.</td>
<td>Result Visualization</td>
<td>84</td>
</tr>
<tr>
<td>6.7.</td>
<td>Costs and Time</td>
<td>85</td>
</tr>
<tr>
<td>7.</td>
<td>Limitations</td>
<td>87</td>
</tr>
<tr>
<td>8.</td>
<td>Conclusion and Future Research</td>
<td>89</td>
</tr>
<tr>
<td>Appendix A:</td>
<td>Completed Questionnaire for the Evaluation (Chapter 6)</td>
<td>92</td>
</tr>
<tr>
<td>Appendix B:</td>
<td>Visualization Code</td>
<td>104</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>106</td>
</tr>
</tbody>
</table>
# List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AABA</td>
<td>Architecture-centric Agile Big Data Analytics</td>
</tr>
<tr>
<td>ABAC</td>
<td>Attribute-based Access Control</td>
</tr>
<tr>
<td>BDRAs</td>
<td>Big Data Reference Architectures</td>
</tr>
<tr>
<td>CapBAC</td>
<td>Capability-based Access Control</td>
</tr>
<tr>
<td>DBMSs</td>
<td>Data Base Management Systems</td>
</tr>
<tr>
<td>DMBOK</td>
<td>Data Management Body of Knowledge</td>
</tr>
<tr>
<td>e.g.</td>
<td>exempli gratia</td>
</tr>
<tr>
<td>EA</td>
<td>Enterprise Architecture</td>
</tr>
<tr>
<td>et al.</td>
<td>et altera</td>
</tr>
<tr>
<td>etc.</td>
<td>et cetera</td>
</tr>
<tr>
<td>EU</td>
<td>European Union</td>
</tr>
<tr>
<td>GDPR</td>
<td>General Data Protection Regulation</td>
</tr>
<tr>
<td>GQM</td>
<td>Goal-Question-Metric</td>
</tr>
<tr>
<td>HDFS</td>
<td>Hadoop Distributed File System</td>
</tr>
<tr>
<td>i.e.</td>
<td>id est</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>IS</td>
<td>Information System</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>MDA</td>
<td>Model Driven Architecture</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>OMG</td>
<td>Object Management Group</td>
</tr>
<tr>
<td>PII</td>
<td>Personal Identifiable Information</td>
</tr>
<tr>
<td>QoC</td>
<td>Quality of Conformance</td>
</tr>
<tr>
<td>QoD</td>
<td>Quality of Design</td>
</tr>
<tr>
<td>RBAC</td>
<td>Role-based Access Control</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td>TOGAF</td>
<td>The Open Group Architecture Framework</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1: Hierarchical Structure of the GQM-Approach (Yahya et al., 2015) ........................................ 6
Figure 2: Data Variety (Batini, et al.2009) .......................................................................................... 11
Figure 3: The Five V’s of Big Data after Serrao and Moura (2016) ......................................................... 12
Figure 4: Business Intelligence versus Advanced Analytics (Minelli et al., 2012) ............................ 13
Figure 5: Big Data Value Generation across different Sectors (McKinsey, 2011) .............................. 14
Figure 6: Conceptual Data Quality Framework (Wang and Strong, 1996) ........................................... 17
Figure 7: Years of Publication ............................................................................................................. 24
Figure 8: Venue Types .......................................................................................................................... 25
Figure 9: The Quality-driven Big Data Application Taxonomy .............................................................. 28
Figure 10: Quality of Conformance .................................................................................................... 29
Figure 11: Usability ............................................................................................................................... 30
Figure 12: Quality Life-cycle (Caballero et al., 2014) ...................................................................... 32
Figure 13: Quality Metadata Creation across the Big Data Pipeline (Immonen et al., 2015) .......... 34
Figure 14: Security ............................................................................................................................... 38
Figure 15: Privacy ................................................................................................................................. 46
Figure 16: Persistence ............................................................................................................................ 54
Figure 17: Data Quality Rule Discovery Model (Taleb and Serhani, 2017) .............................................. 57
Figure 18: Validity ................................................................................................................................. 65
Figure 19: Radar-Chart Example .......................................................................................................... 72
Figure 20: Predictive Maintenance ....................................................................................................... 74
Figure 21: Recommendation Systems ................................................................................................. 84
List of Tables

Table 1: Data Extraction ............................................................................................................. 22
Table 2: Requirements-centric Classification of Literature ..................................................... 26
Table 3: Differences in Big Data Quality Dimensions............................................................... 31
Table 4: Classification of Pre-processing Techniques ............................................................... 35
Table 5: Classification of Confidentiality Models and Techniques ........................................... 39
Table 6: Classification of Integrity Models and Techniques ..................................................... 41
Table 7: Classification of Availability and System Immunity Models and Techniques .......... 43
Table 8: Anonymization/De-Identification Techniques ............................................................. 48
Table 9: Key Big Data Reference Architectures ....................................................................... 60
1. Introduction

1.1. Motivation

“Water, water, everywhere, nor any drop to drink “

Samuel Tayler Coleridge, 1834

This quotation of the famous poem *Rime of the Ancient Mariner* from Samuel Tayler Coleridge could not be more appropriate to centralize the problem of this topic:

Since the beginning of the 21st century, the world has faced a number of significant changes in Information Technology (IT), like Internet of Things (IoT), cloud computing or social networking (Cai and Zhu, 2015). Those technological innovations are new key-enablers for competitive edge, because they facilitate skyrocketing processing power, communication velocity and ubiquitous device connections (Caballero et al., 2014). And all these rising innovations have one thing in common: They produce enormous amounts of data (Gudivada et al., 2015). The reason is called “Datafication”, which considers the ability to transform aspects of the daily life into data for the purpose of maximizing utilization (Ardagna et al., 2018). Data growth has already exceeded Moore’s Law (Chen and Zhang, 2014) and will grow from 4.4 Zettabytes in 2013 to 44 Zettabytes in 2020 according to IDC’s digital universe forecast (IDC, 2014). This new phenomenon, which seems to be only characterized by its tremendous amount of data, is called “Big Data”:

Generally, the term “Big Data” gives the impression to be rather abstract. Mostly, it is related to something very big and informational (Ohlhorst, 2012). But Big Data is more than just large amounts of data. Gartner Inc. (2019) define Big Data as following: “Big data is high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation”.

The immanent value that is buried in those vast data masses experiences growing attention in industries, like Finance, Healthcare, Retail and many more (Marr, 2016a). In 2017, *The Economist* entitles Big Data as the new oil of the digital era (The Economist, 2017). Its sometimes-unbelievable power has already been demonstrated in some real-world applications.
For example, many people know the story about Andre Pole, a statistician of the company Target, who developed a customer tracking method that predicted the pregnancy of a young girl before her father knew, only by looking at the product purchase behavior (Lubin, 2012). A more recent controversy discussion was also made about the data-driven manipulation in Donald Trump’s presidential campaign where the data analysis company “Cambridge Analytica” analyzed massive datasets from Facebook surveys and proprietary databases to profile and target potential voters (Kirchgaessner, 2017). In contrast, Big Data also demonstrates its value through promising opportunities, like the pattern recognition software “IBM Watson” that supports clinical diagnosis and has been proven as probably even more effective as human diagnosis in searching and spotting cancer (Marr, 2016b).

What this all amounts to is that Big Data has the potential for ground-breaking business value as it serves competitive advantage for various industries. The big problem is (and therefore turning back to the opening quotation of Samuel Tayler Coleridge) that nowadays, data is everywhere. As a result, it is highly challenging to distill valuable information from this data flood. Data can be outdated, noisy, biased or incorrect (Ardagna et al., 2018), which could make it hard to use it appropriately.

Typically, data-driven decisions are solely as meaningful as the data (or information), on which the decision is based on (Hazen and Boone, 2014). In other words, an inevitable prerequisite for knowledge creation is that data have a certain level of quality (Cai and Zhu, 2015), otherwise it might result in monetary losses. As a proving example, according to a survey of The Data Warehousing Institute (TDWI), the United States economy incurs a loss of more than 600 billion dollars per year due to data quality problems (Abdullah et al., 2015). This problem becomes even more severe within Big Data, because V-characteristics strongly affect the quality of data (Taleb et al., 2015). Challenges of Big Data quality include diverse data sources that produce heterogeneous data types, massive data volume that leads to difficulties in judging data quality in reasonable time, very short timeliness of data and missing standards on data quality (Cai and Zhu, 2015).

Today, research on Big Data quality lacks (Cai and Zhu, 2015). Definitions of data quality evolved from “fitness for use” (Wang and Strong, 1996, p. 6) in traditional settings to “fitness for purpose” (Caballero et al., 2014, p. 67) in a Big Data-context. Usually, data quality can be described as a hierarchical, multi-dimensional concept consisting of a set of characteristics or properties (so-called attributes) that partly describe the nature of qualitative data (Scannapieco and Cartaci, 2002; Wang and Strong, 1996). Common data quality attributes include accuracy,
consistency or completeness (Scannapieco and Cartaci, 2002). But on grounds of Big Data peculiarities the focus has changed, because Big Data does not solely rely on the data itself. Rather more, it is a holistic framework that includes data and corresponding technological innovations focusing on data storage, provision, formats, processing as well as analytics (Caballero et al., 2014). This means, Big Data is not only dependent on data but more on technologies and techniques that serve the ability to handle large-scale, heterogeneous and rapid data (Abdullah et al., 2015). Applying the right technologies will foster the facilitation of data quality (Loshin, 2014). Consequently, system quality must be addressed, otherwise Big Data will lead to general dissatisfaction and lost revenues (Abdullah et al., 2015). Example technologies in this regard are NoSQL databases, Hadoop framework or cloud computing (Hashem et al., 2015). The importance of the interrelationship between Big Data application quality, Big Data V’s and traditional data quality attributes has already been proven by Noorwali et al. (2016). Another vital aspect is the way data travel through different phases in the Big Data life-cycle, including data generation, acquisition, storage, and analytics. Data quality must be addressed in each of those phases (Taleb et al., 2015). Existing research challenges considering Big Data application quality include scalability, availability, privacy, security, integrity, heterogeneity and others (Hashem et al., 2015). For instance, looking at privacy challenges, the recent privacy breach on Facebook where around 87 million user accounts were abused (Badshah, 2018) shows that privacy is far from perfect.

The motivation to conduct research in this area is that up to date, there is a plain gap in providing a comprehensive set of data quality-specific requirements for Big Data applications. Rather more, those quality requirements are spread across various standards, scientific as well as business literature. This master thesis aims to fill this research gap by providing a holistic classification of quality-centric Big Data application requirements and by demonstrating its usefulness.
1.2. **Scope and Boundaries**

In order to introduce the reader to the work at hand, this chapter clearly defines the scope and boundaries. In particular, it specifies the area of research, the conducted research method for the artifact development, the approach used for the evaluation, and the case specifications, on which the artifact evaluation is based. Generally, the work is not intended to explore technical issues of Big Data applications. Instead, it sets its focus on identifying requirements that go beyond technical questions. Such requirements are denoted as non-functional requirements. Non-functional requirements are quality attributes like interoperability, reliability, usability, security, or others.

1.2.1. **Design-Science Research**

This master thesis is based on design-science research, which “*seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts*” (Hevner et al., 2004, p.75). The developed artifact is embodied by a non-functional requirements specification framework for data quality in Big Data applications that extends existing boundaries of applying Big Data in enterprises by supporting a quality-aware implementation. Consequently, the different sections follow the design-science guidelines presented by Hevner et al. (2004):

- **Design as an artifact:** The key-contribution is the development of the non-functional requirements specification framework that describes data-specific quality requirements for Big Data applications.
- **Problem relevance:** The relevance of the problem is already discussed in chapter 1.1.
- **Design evaluation:** The developed artifact is evaluated on two case examples, which are described in chapter 1.2.4.
- **Research contribution:** This master thesis breaks down the design processes of Big Data applications that should assist quality-aware decision-making activities regarding Big Data application initiations.
- **Research rigor:** The artifact rests upon a review of relevant literature in multiple fields. The literature review is addressed in chapter 1.2.2 and more detailed in chapter 3.

- **Design as a search process:** The artifact is developed by successively derive non-functional requirements of Big Data application quality found in literature. Identified quality attributes are hierarchically structured and integrated into quality dimensions.

- **Communication of research:** The resulting taxonomy is valuable for both, technological and management audiences, by delivering aspects of non-functional requirements as well as strategic decision support for Big Data application quality.

### 1.2.2. Taxonomy based on Literature Review

The taxonomy, i.e. the non-functional requirements specification framework, builds on an in-depth review of relevant literature in various fields, like data quality, security or privacy. Standards, scientific as well as business literature are used to extract relevant quality requirements. A detailed description of the literature review conducted is provided in chapter 3. Generally, a literature review is a valuable means for establishing a sturdy foundation for progressive knowledge by uncovering new areas for future research (Webster and Watson, 2002). A key-challenge in conducting literature reviews is to stay focused on the topic and define a clear scope. All too often, a literature review includes so many references that it looks more like a telephone book but comprises only little plot. Thus, it is important to balance the quantity of reviewed literature with the quality of the research itself (Webster and Watson, 2002).

According to Webster and Watson (2002), a good literature review is concept-centric and uses visualizations to communicate findings. Moreover, the review should not be too overcritical and written in present tense.
1.2.3. Criteria Catalogue using a Goal-Question-Metric (GQM-) Approach

In order to evaluate the developed taxonomy, a criteria catalogue is established. This catalogue is epitomized in the form of a questionnaire. Additionally, to make quality criteria tangible and measurable, metrics are defined for each question. One way to define questions and resulting metrics is the Goal-Question-Metric (GQM-) approach after Basili and Weiss (1984).

GQM uses a top-down, goal-oriented way to select and derive metrics for specific tasks. As the name implies, a GQM-approach consists of goals, questions and metrics within a hierarchical structure. Top-down means starting with the definition of particular goals. These goals are made conceptually. They get on an operational level by defining questions that should support reaching the goals. In a last step, metrics are defined for measuring purposes. Metrics are quantitative to assign numbers to the qualitative attributes. Such a top-down approach has the advantage of a more precisely selection of the relevant metrics for achieving a specific goal (Koziolek, 2008).

![Hierarchical Structure of the GQM-Approach](image)

*Figure 1: Hierarchical Structure of the GQM-Approach (Yahya et al., 2015)*

With the help of metrics, I afterwards visualize the results in form of a chart to improve comprehensibility as well as readability of the outcomes from the criteria catalogue.
1.2.4. Evaluation Case Specifications

The evaluation is done with the aid of two real-world examples. In particular, I selected “Predictive Maintenance” and “Recommendation Systems” as possible Big Data applications to assess the developed artifact. Those cases are largely taken from Marr (2016a) who describes 45 examples of successful Big Data applications.

Predictive Maintenance at Rolls Royce

Rolls-Royce, which is probably better known for its superior cars, produces massive engines for airlines as well as armed forces. Producing such engines consists of high-complex processes that must be strongly addressed since one mistake can cost human lives. To cope with that, Rolls-Royce implemented Big Data analytics in all of their manufacturing processes including design, manufacturing and after-sales. The focus will rely on after-sales support. After-sales support uses Big Data to predict possible needs for engine maintenance. To do so, every engine sold is equipped with hundreds of different sensors that continuously collect data from the engine. The data are sent to analysts who assess them using advanced analytical methods. In cases of deviations within the data, the analysts can make predictions of possible maintenance needs and notify the customer (Marr, 2016a).

Recommendation Systems at Amazon

Retail is also a high-potential area for applying Big Data due to the tremendous data produced. Large retail companies like Amazon try to comprise all items a customer needs “under one roof”. Having as much items as possible in a supermarket, a customer gets quickly crushed by the enormous supply. To deal with that problem, Amazon started collecting data about their customers. They collect information about what they buy, what they are looking for, where they buy, and many more. Afterwards, they create “360-degree views” of the customers and offer them individual tailored products or items they might be interested in. Amazon was the first company that used recommendation systems at large-scale (Marr, 2016a).

In summary, both seem to have basically similar intentions. They analyze data to make suggestions, either for potential item purchases or for recommended maintenance dates. However, they have some peculiarities that will be analyzed in chapter 6.
1.3. Research Question

According to the defined scope and boundaries, the following research question can be defined:

“What are data quality-specific characteristics of Big Data applications under consideration of potential interdependencies between data quality and other relevant topics, towards a Non-functional Requirements-Specification Framework?”

1.4. Thesis Structure

The thesis consists of eight chapters. In the following, each chapter is shortly described:

Introduction (Chapter 1): Chapter 1 starts with a motivation that should get the reader’s attention on the topic and the problem at-hand. Afterwards, scope and boundaries are clearly defined to set the focus of this work. With that scope in mind, the research question is specified. This chapter ends with an overview of the thesis’ structure.

Theoretical Background (Chapter 2): The theoretical background describes the key concepts, which are critical for this topic. In particular, the concept of Big Data as well as the concept of data quality are explained.

Research Method (Chapter 3): Chapter 3 outlines the research method used. It specifies material sources, search strategy, inclusion and exclusion criteria, as well as data extraction and classification of the identified literature.

Results (Chapter 4): The fourth chapter represents the main part of the thesis and successively collects quality attributes for Big Data applications based on literature review.

Criteria Catalogue (Chapter 5): Chapter 5 defines the basic structure of the questionnaire based on a GQM-approach. Additionally, the calculation of the quality scores and the development of the visualization (i.e. radar-chart) are explained.

Evaluation (Chapter 6): This chapter evaluates the developed artifact by answering the criteria catalogue for two different Big Data applications, which are specified in chapter 1.2.4. This is done to reveal potential differences in quality profiles of the applications.
Limitations (Chapter 7): Chapter 7 states possible limitations of this work.

Conclusion and Future Research (Chapter 8): The last chapter reviews the work, gives resulting conclusions and set directions for future research.
2. Theoretical Background

The theoretical background should support the reader’s comprehension on the most important notions of this topic. The first part gives a general view over Big Data. It defines the term and specifies its characteristics. Additionally, it describes Big Data analytics and Big Data applications to get a clear sense of how, where and why Big Data is applied. The second part presents the concept of quality in this context. In particular, data quality and its related perspectives are explained.

2.1. The Concept of Big Data

Data have influenced our daily life. We can experience data at work, during our leisure time, school or study. Mostly, we have no idea where these data streams come from, how they are used or what they really do. Some people only know that there is some kind of IT running in the background (Mainzer, 2014). Against all odds, this phenomenon accompanies us much longer we would suppose:

If we take the evolution of humanity, we can identify that we were always concerned with Big Data. Every day, the human body is faced with a tremendous number of signals, like sounds, lights, shadows, pressure, heat, coldness and many more. During millennia of evolution, humans developed several organs and functions to automatically process all those signals. Additionally, the body comprises different filters that allow only essential signals being recognized. Searching for patterns in the flood of signals is vital (Mainzer, 2014).

Transferring this example into our context, we can identify similarities. Today, the world produces vast amounts of data every day (represented by the signals in our example). These data origin from different sources, like smart devices (e.g. smartphones), sensors, social media, company-specific Information Systems (IS) and others (Gudivada et al., 2015; Kwon et al., 2014; Fang et al., 2017). Additionally, the data exhibit different types (e.g. videos, documents, audio files, etc.) (Chen et al., 2014b). Like the human body that instantly processes incoming signals, data must be also recognized and processed forthwith (Chen et al., 2014b). Consequently, this new phenomenon of high volume, high variant and fast-paced data production has its own characteristics and is therefore denoted by a new term, namely “Big Data”.
2.1.1. Definition of Big Data

Although, the value of Big Data has been broadly recognized, there are still different opinions regarding its definition (Chen et al., 2014b). A broad consensus describes Big Data by its three core-V’s Volume, Variety and Velocity (Russom, 2011), which were first mentioned by Doug Laney back in 2001 (Laney, 2001). For example, Gartner Inc. define Big Data as follows:

“Big data is high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation” (Gartner Inc., 2019).

Other definitions emphasize additional V-characteristics. Chen et al. (2014b) as well as Cai and Zhu (2015) define Big Data with an additional V, namely Value. Abdullah et al. (2015) or Demchenko et al. (2013) use five V’s for Big Data characterization. The fifth V is called Veracity. The characteristics can be described as follows:

**Volume**: As the name implies, Big Data will come in only one size: big. Companies are downright flooded with data, often accumulated to Terabytes (Ohlhorst, 2012). Problems that come along with Volume are that those quantities often exceed the storage abilities of traditional Database Management Systems (DBMSs) (Klein et al., 2013). The size of the collected data can be unlimited whereas the power of a computer is limited (Tole, 2013). As an example, Facebook generates 650.000 contents per minute that must be stored and processed continuously (Klein et al., 2013).
**Velocity:** Big Data is time-sensitive and must often be processed straightaway as the data enter a company for the purpose of value maximization (Ohlhorst, 2012). The high number of devices people use nowadays (smart phones, tablets, laptops, etc.) imply large data stream requests. Once again, the size of requests can be unlimited, the power of a computer is limited (Tole, 2013).

**Variety:** Data Variety explains the various data types a computer should store, use and analyze. Those different types of data are mostly semi- or unstructured (e.g. maps, video files, etc.) (Tole, 2013; Demchenko et al., 2013). Challenges emerging with data Variety are mainly storage problems. Traditional DBMSs store data according to relations. In doing that, data must be structured (i.e. data must have similar attributes). Due to the forthcoming data that is not structured, traditional DBMSs are quickly overwhelmed (Klein et al., 2013).

**Value:** Value points out that Big Data is only meaningful, if the data are valuable. If companies have access to large amounts of data is trivial unless they can get value out of them (Abdullah et al., 2015).

**Veracity:** Veracity refers to the key words consistency and trustworthiness of the included data. Data consistency defines how reliable the data is if it gets transmitted often. Trustworthiness should make sure that data is trusted, protected and not modified when receiving it (Demchenko et al., 2013).

*Figure 3: The Five V’s of Big Data after Serrao and Moura (2016)*
2.1.2. Big Data Analytics

Like the signals a human body is exposed to, the data that we produce also include important but also negligible information. Therefore, we need to extract data that is useful in a given context. To reach a set of valuable data, scientists concluded the more data the better the results (Ohlhorst, 2012). This means, increasing data, analysis and results. In doing that, researchers started integrating unstructured data, real-time data, related datasets and archival data into the analysis process (Ohlhorst, 2012). Consequently, a shift is done from transactional analytics (using mainly structured data) to behavioural analytics (also including unstructured data) (Ohlhorst, 2012).

Analyzing Big Data is mostly done by means of advanced analytics, which are able to deal with various data types. Applying advanced analytics on large datasets requires a number of different techniques to reach desired outcomes (Minelli et al., 2012). As figure 4 shows, there are several methods existent in analytics. Traditional business intelligence differs from advanced analytics, because it mostly uses basic statistic methods and focus more on current and past periods by compiling reports or queries related to important business questions (Minelli et al., 2012). Big Data evokes a number of new analysis techniques (i.e. advanced analytics) using high potential methods in order to put predictions on a new level. They promise new insights on large datasets by gathering more value out of them (Minelli et al., 2012). Hence, these methods help to extract the intrinsic value of the data for a desired purpose.

![Figure 4: Business Intelligence versus Advanced Analytics (Minelli et al., 2012)]
2.1.3. Big Data Applications

Big Data is becoming increasingly popular and there are already a number of enterprises out there experiencing the magic of Big Data (Marr, 2016a). Searching for reasons, we will find that Big Data applications are mostly used, because they promise increased performances that have the potential to rule out competitors (McKinsey, 2011). For example, Brynjolfsson et al. (2011) surveyed 179 large firms on their potential usage of data driven decision-making. They found that data driven decision-making leads to a productivity increase of five to six percent. Also, McKinsey (2011) studied possible benefits of Big Data applications in various fields. They examined five different areas, namely health care, public administration, retail, manufacturing and global personal location data. Results show that every sector experiences a growth in their productivity, as displayed in figure 5. Additionally, the quality of the produced outputs also raised due to customer specific tailoring of the products as well as cost and price reductions. In consequence of these perceptions, it can be said that Big Data applications implicate promising benefits that maximize the performance of an enterprise. Hence, Big Data is important, because it will transform each of these areas and industry should be prepared to exploit data in an optimized way.

![Figure 5: Big Data Value Generation across different Sectors (McKinsey, 2011)](image-url)
2.2. The Concept of Quality

In general, the term “Quality” is an atemporal concept (Juran and Godfrey, 1999). The provenance of mastering quality is hidden in ancient pasts, which means, human evolution has always been facing the problem of dealing with quality (Juran and Godfrey, 1999). For example, primitive food-gatherers had to understand which fruits are poisonous and which are edible, and hunters had to learn which wood is best for building hunting weapons. Valuable perceptions were afterwards passed further to the next generation. In this time, humans were responsible to only meet family needs. Efforts in meeting own needs and hence, achieving individual perceived quality, could be denoted as the foundation of technology. As humans evolve, people started to build villages and to allocate different roles and tasks for the purpose of general welfare. By diverging from meeting individual family needs and moving towards meeting the needs of larger populations, the foundation of technology also moved towards technology evolution (Juran and Godfrey, 1999).

According to Juran and Godfrey (1999), the meaning of “Quality” in terms of quality management is twofold:

1. “Quality” refers to product features that meet customer needs and therefore, deliver customer satisfaction. In this perception, quality refers to income. The purpose is to achieve high customer satisfaction, which, in turn, could lead to increased income. But satisfying customer needs will also include some kind of investment.

2. “Quality” addresses the freedom of errors, which requires iterative work of error elimination. Here, quality refers to costs. Identifying and removing errors can be a daunting task leading to customer dissatisfaction. Comparing costs of error elimination with investments of customer satisfaction, the second might be cheaper.

In summary, quality isn´t a new term and can be denoted as the engine of technology. It can be achieved by customer satisfaction and error elimination, but the first might be more sensible.

2.2.1. Definition of Data Quality

Academic efforts on data quality are largely based on three research areas, namely computer science, statistics and management. Statistics was the first research area addressing data quality problems in the 1960´s, mostly with the help of mathematical models. In the 1980’s,
management started investigating potential data quality issues. Focus relied on data manufacturing systems for quality problem elimination. Computer science was the last research area concerning about data quality since the 1990’s. They mainly focused on electronical data within databases (Scannapieco and Catarci, 2002).

Commonly, data quality is a hierarchical, multidimensional concept (Wang and Strong, 1996). This means, it consists of a set of characteristics or properties, so-called attributes, that partly describe the nature of qualitative data and are often split up into dimensions, resulting in a classification scheme (Scannapieco and Catarci, 2002). While there are many data quality classification schemes in literature, an overall conform set of attributes is not existent (Scannapieco and Catarci, 2002). However, defining the term “Data Quality”, we can recognize variances with respect to time and field (Sidi et al., 2012). Wang and Strong (1996) for example define data quality as “fitness for use” (p.6). Cappiello et al. (2004) state that data quality “refers to the degree to which data satisfy user requirements or are suitable for a specific process” (p. 68). According to Caballero et al. (2014, p. 67), data quality is defined as “fitness for purpose”.

2.2.2. Data Quality Perspectives

Data quality can largely be viewed from two perspectives, namely “Quality of Conformance” (QoC) and “Quality of Design” (QoD), e.g. Wang and Strong (1996), Redman (1996), or Wang (1998). According to Helfert and Heinrich (2003), QoD “refers to the degree of correspondence between user requirements and their concretion in specifications” (p. 3). QoC “enfolds the degree of correspondence between […] production processes and its products” (Helfert and Heinrich, 2003, p. 3). In other words, QoC indicates the degree of conformity between data values and related real-world counterparts. QoD is expressed by the level of compliance between the application specifications and the user needs within data handling processes.
A well-known example of a data quality classification, which includes both perspectives, is provided by Wang and Strong (1996). The value of this classification is that data quality attributes were collected from data consumer needs. The research is grounded on an empirical study by conducting two surveys on data users. The first survey was conducted to get a comprehensive list of data quality dimensions. The second survey explored the perceived importance of the identified data quality attributes from survey one. The authors received 179 different quality attributes from which 15 were finally selected. The classification can be viewed in figure 6.

![Figure 6: Conceptual Data Quality Framework (Wang and Strong, 1996)](image)

It consists of four dimensions, each including a number of data quality attributes. *Intrinsic data quality* describes inherent quality attributes of the data itself. *Contextual data quality* refers to quality attributes that must be defined for the task at-hand. For example, if data can be denoted as “on-time” might differ between two distinct processes. While the first two dimensions are data-centric, *representational data quality* as well as *accessibility data quality* emphasize system design importance, i.e. the system must be able to clearly represent data and make data accessible to the users.

The example above stresses the existence of data quality attributes that refer to the data itself while others should be addressed by system design. Below, I shortly present some important quality attributes for both perspectives:

Since there are many different data quality classifications, Scannapieco and Cartaci (2002) analyzed a number of classifications and developed a principal set of data quality attributes
which were addressed by the bulk of prior research, including accuracy, consistency, completeness and timeliness:

**Accuracy** refers to the level “the data are correct, reliable, and certified free of errors” (Wang and Strong, 1996, p. 31). For example, if an attribute of a record is “Name” and the value of this attribute is “Michael”, then this value is highly accurate, because it is correct according to English dictionaries of names. If the name would be “Micl”, the accuracy would be low (Scannapieco and Catarci, 2002).

**Completeness** gives indication to what “extent the data are of sufficient breadth, depth, and scope for the task at hand” (Wang and Strong, 1996, p. 32). As an example, take a record “Person” with the attributes “Name”, “Telephone Number” and “Date of Birth”. If there exists a record without a telephone number, then either the record is not complete, or the person does not have a telephone number (Scannapieco and Catarci, 2002).

**Consistency** is “the extent to which data are always presented in the same format and are compatible with previous data” (Wang and Strong, 1996, p. 32). For instance, consider the two attributes “Name” and “Sex”. If “Name” includes the value “John” and “Sex” has the value “Female”, then the values might have low consistency (Scannapieco and Catarci, 2002).

**Timeliness** simply defines “the extent to which the age of the data is appropriate for the task at hand” (Wand and Strong, 1996, p. 32). This dimension is mostly treated with other time-related dimensions called Currency (how current is the data) and Volatility (how long the data stays valid) (Scannapieco and Catarci, 2002).

Examples for design quality attributes are accessibility or access security according to the classification from Wang and Strong (1996):

**Accessibility** is defined as “the extent to which data are available or easily and quickly retrievable” (Wang and Strong, 2016, p.32). Accessibility is strongly associated to availability defined in ISO/IEC 27000 (International Organization for Standardization, 2016b). According to this standard, availability is the “property of being accessible and usable upon demand by an authorized entity” (p. 3).

**Access security** can be denoted as “the extent to which access to data can be restricted and hence kept secure” (Wang and Strong, 2016, p. 32). This is closely related to integrity in ISO/IEC 27000 (International Organization for Standardization, 2016b). They define integrity
as the “property of accuracy and completeness” (p. 7), which refers to the ability to protect data from unauthorized deletion or modification.

In summary, data quality is not just a unidimensional concept. It consists of (i) inherent quality attributes, which directly refer to the data (e.g. accuracy, timeliness, consistency, or completeness), and (ii) surrounding design quality attributes, which should satisfy user needs (e.g. accessibility). Combining chapter 2.1 and 2.2, the task of this work is to identify data quality design attributes for Big Data applications. Especially in a Big Data-context, quality design attributes indicate to have peculiarities due to the already described V-characteristics. As one might see within the explanations of accessibility or access security, traditional standards are valuable means to support the development of the right IS. Examples of relevant standards are DAMA Data Management Body Of Knowledge (DMBOK), The Open Group Architecture Framework (TOGAF), ISO/IEC 8000-61, etc. The problem is that those standards are intentionally built with respect to traditional IS. Therefore, such standards might miss important considerations regarding Big Data application design. Moreover, standards that directly focus on Big Data are rare. The main standards in this regard are provided by the National Institute of Standards and Technology (NIST), e.g. NIST (2015b). On that account, there is also a high need to base design efforts on various Big Data-related academic and business literature.

In the next section, I present the research method used for the collection of quality design attributes for Big Data applications.
3. Research Method

The literature review conducted to synthesize quality characteristics of data-intensive applications generally incorporates three types of literature: (i) academic papers, (ii) standards, and (iii) business articles. The search process for academic papers is based on automated search using online databases and follows the guidelines from Webster and Watson (2002) as well as some concepts from Kitchenham (2007) for conducting systematic literature reviews. Standards and business articles are largely grounded on literature provided by the supervisor and on traditional Google search. However, the main challenge in performing a literature review in the area of IS is the interdisciplinary nature of the field (Webster and Watson, 2002). As Webster and Watson (2002) state: “[...] constructing a review is a challenging process because we often need to draw on theories from a variety of fields” (p. xiii-xiv). In doing so, there is a need to loosen the strictness of analyzing only one research area by allowing to concede a meaningful level of freedom in the search process.

The next subchapters describe how the literature review was applied. In particular, the sources of material collection are shortly presented in the first part. Afterwards, a clear search strategy is defined in the second part. Additionally, inclusion and exclusion criteria are described to break down the returned results from the online database search into a meaningful number. Finally, data extraction and classification characterize the identified set of literature.

3.1. Material Sources

The sources of material used in this work are:

- Starting set provided by the supervisor and initial search for the thesis’ proposal
- Google Scholar
- IEEE Xplore Digital Library
- Traditional Google search

In supporting initial definition of scope, a starting set of scientific literature and standards were provided by the supervisor of the thesis. The starting set mainly consists of literature related to Big Data in general, (Big) data quality, (Big) data security and privacy as well as live experimentation for quality requirements validation. The focus areas of standards were data
quality management, enterprise architecture management, and security and privacy management.

Additionally, the two databases Google Scholar and IEEE Xplore Digital Library were used to extract a meaningful set of further scientific literature. The reason for using two databases is that one database might miss some important papers which are provided in the other one. Finally, traditional Google search was conducted for further identification of standards as well as business literature that should provide a practical view on the topic. Google search was also conducted for one online source.

3.2. Search Strategy

Regarding the search strategy for scientific literature, advanced search was used for both, Google Scholar and IEEE Xplore Digital Library. Search strings were constructed by using Boolean terms “AND” and “OR”. In doing so, keywords are defined prior to the search process. One major validity threat in this regard is that the quality of the literature review is highly dependent on the quality of defined search strings. If the quality of keywords is low, the literature review might be based on low quality papers, which, in turn, leads to a low-quality research in general. To counter this validity threat, two processes of key-word identification were used. Keywords are defined either by reading the papers provided in the starting set or by using PICO (Population-Intervention-Comparison-Outcome-) criteria suggested by Kitchenham (2007). Additionally, according to Webster and Watson (2002), it is valuable to review citations of the top articles (go backward) and look what articles cited the top articles (go forward), which was also applied.

Search strings have different syntax depending on the database the search is performed. However, an example pattern of the search strings used are provided in table 1.
Table 1: Data Extraction

<table>
<thead>
<tr>
<th>Search string pattern examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>“big data” AND (“data quality” OR “quality requirements” OR “quality attributes” OR “quality challenges” OR “quality assurance” OR “data quality challenges” OR “metadata management”)</td>
</tr>
<tr>
<td>“big data” AND (“user requirements” OR “user needs” OR “user acceptance” OR usability)</td>
</tr>
<tr>
<td>“big data” AND (“privacy issues” OR “security issues” OR “security challenges” OR “privacy challenges” OR “security opportunities” OR “privacy opportunities” OR “security solutions” OR “privacy solutions”)</td>
</tr>
<tr>
<td>“big data” AND (“enterprise architecture” OR “reference architecture” OR “quality architecture” OR “data architecture” OR “big data architecture” OR “enterprise architecture management” OR “architecture management”)</td>
</tr>
<tr>
<td>“big data” AND (“large scale experiments” OR “controlled experiments” OR “online controlled experiments” OR “live experiments” OR “continuous experiments”)</td>
</tr>
<tr>
<td>“big data” AND (scalability OR scalable)</td>
</tr>
<tr>
<td>“big data” AND (agility OR agile)</td>
</tr>
</tbody>
</table>

I have to note that if a topic was found especially important and further information was needed, more detailed search was conducted, which might also include other search strings than those presented above.

The search strategy for standards is mainly based on the focus areas described above. For each area, standards were searched using traditional Google search and selected according to their fitness and popularity. Here, the main validity threat is that standards are often not available for free. Thus, it may be the case that other valuable standards are missed in the identified set. However, some of the most important standards are provided by the supervisor through the starting set.

The search strategy for business articles also focus on traditional Google search. The reason for applying traditional Google search is that business articles often cannot be found in academic online databases. I selected those articles that best fit into my topic and supplement identified scientific papers.
3.3. Inclusion and Exclusion Criteria

Inclusion and exclusion criteria should support the identification of the most relevant papers. Applying inclusion and exclusion criteria helps to eliminate papers that are not specific to the research question(s). The identified inclusion and exclusion criteria are presented below:

**Inclusion criteria:**
- Literature in English
- Academic papers and business articles focusing on Big Data
- Standards on both, Big Data- and non-Big Data-related topics within the pre-defined scope
- Solely research papers and business articles published after 2012 are included due to the novelty of the field
- Within the Google Scholar search, results are restricted to title

**Exclusion criteria:**
- Literature not in English
- Literature not available for me because of missing access rights
- Scientific research and business articles that do not focus on Big Data
- Tutorial papers
- PhD or Master’s thesis
- Books

3.4. Data Extraction and Classification

From the starting set, the initial database search and based on inclusion and exclusion criteria, I identified 80 papers in Google Scholar. Four additional papers were found in IEEE Xplore Digital Library which were not available in Google Scholar. Applying “go forwards” and “go backwards” suggested by Webster and Watson (2002, p. xvi) lead to another six papers. One online article was used from the official homepage of the European Commission about the new General Data Protection Regulation (GDPR). The rest of the literature is embodied by standards (or literature on standards) as well as business articles and counts 20.

Overall, 111 papers, standards, and articles were identified for performing the systematic literature review.
Figure 7 shows the publication years of the identified literature. Traditional standards are excluded, because first, they are often older and therefore would distort the visualization and second, they are continuously renewed. Additionally, the online resource is also excluded, because no data was provided.

As visualized in figure 8, the identified set of literature is primarily published in scientific journals or in international conferences. Four papers are grounded on workshops. One paper is based respectively on symposia and forums. “Others” refers to standards or business articles which cannot be classified in one of the other types.
The next chapter represents the main part where literature is classified according to the identified quality attributes and each of the attributes is described in detail.
4. Results

This section presents the results from the literature review. In particular, the identified quality dimensions and their corresponding quality attributes are successively elucidated. Table 2 delivers a non-functional requirements-centric classification of the identified literature. The table gradually displays each quality dimension and its comprised attributes. Additionally, related papers and paper counts are included. Papers are stated within an attribute if they address the topic or give some key-statements. It is noteworthy, that there are papers or standards that cover more than one attribute.

**Table 2: Requirements-centric Classification of Literature**

<table>
<thead>
<tr>
<th>Quality dimension/attribute</th>
<th>Papers</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Usability</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contextual-Requirements Completeness</td>
<td>Becker and McMullen (2015); Judoo (2015); Ardagna et al. (2018); Firmani et al. (2016); Ramaswamy et al. (2013); Abdullah et al. (2015); Cai and Zhu (2015); Hazen and Boone (2014); Caballero et al. (2014); Lakshen et al. (2016); Mosley (2010); ISO/IEC 8000-61 (International Organization for Standardization, 2016a); Serhani et al. (2016); Noorwali et al. (2016)</td>
<td>14</td>
</tr>
<tr>
<td>Metadata Provision</td>
<td>Ardagna et al. (2018); Immonen et al. (2015); Klüs et al. (2016); Supriya and Devendrasingh (2017); Wahyudi et al. (2018); Serhani et al. (2016); Dinter et al. (2015); Kulkarni (2016); Smith et al. (2014); Mosley (2010)</td>
<td>10</td>
</tr>
<tr>
<td>Data Fitness</td>
<td>Chen et al. (2014b); Serhani et al. (2016); Taleb et al. (2015); ISO/IEC 8000-61 (International Organization for Standardization, 2016a)</td>
<td>4</td>
</tr>
<tr>
<td>User Acceptance</td>
<td>Abbasi et al. (2016); Shin (2016); Khan and Brock (2017)</td>
<td>3</td>
</tr>
<tr>
<td><strong>2. Security</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confidentiality</td>
<td>Joshi and Kadiwal (2017); Duhan and Singh (2018); Abouelmehdi et al. (2018); Manikandakumaran and Ramanujam (2018); Samsudin (2016); Serrao and Moura (2016); NIST (2015a); Big Data Working Group (2016); Chaudhary and Kumar (2017); Mattsson (2014); Kumar et al. (2015); Tsaur and Wu (2013); Jain et al. (2016); Toshniwal et al. (2015); ISO/IEC 27000 (International Organization for Standardization, 2016b); ISO/IEC 27002 (International Organization for Standardization, 2005)</td>
<td>16</td>
</tr>
<tr>
<td>Integrity</td>
<td>Alshafi and Saife (2017); Ward and Barker (2013); Duhan and Singh (2018); Abouelmehdi et al. (2018); Big Data Working Group (2016); Chen et al. (2014a); NIST (2015a); Stojkov et al. (2017); Xu and Shi (2016); Cloud Security Alliance (2013); Kumar et al. (2015); Kanyan and Mehra (2018); ISO/IEC 27000 (International Organization for Standardization, 2016b); ISO/IEC 27002 (International Organization for Standardization, 2005)</td>
<td>14</td>
</tr>
<tr>
<td>Availability and System Immunity</td>
<td>Alshafti and Saife (2017); Duhan and Singh (2018); Joshi and Kadhivala (2017); Kune et al. (2016); Subbalakshmi and Madhavi (2018); Mishra and Singh (2016); Joglekar and Pise (2016); Akutota and Choudhury (2017); ISO/IEC 27000 (International Organization for Standardization, 2016b); ISO/IEC 27002 (International Organization for Standardization, 2005)</td>
<td>10</td>
</tr>
<tr>
<td>Accountability</td>
<td>Mosley (2010); Mattsson (2014); ENISA (2015); ISO/IEC 27002 (International Organization for Standardization, 2005)</td>
<td>4</td>
</tr>
</tbody>
</table>

### 3. Privacy

| Non-disclosure | Referring to the literature identified in Confidentiality and Integrity at Security | - |
| Pre-serving | Terzi et al. (2015); Jhaveri et al. (2015); Duhan and Singh (2018); NIST (2015a); Jain et al. (2016); Bisht and Singh (2016); Fang et al. (2017); Benjelloun and Lahcen (2015); Big Data Working Group (2016); Soria-Comas and Domingo-Ferrer (2016); Abdouelmezhdi et al. (2018); Xu and Shi (2016) | 12 |
| Provenance | Xu et al. (2014); Joshi and Kadhivala (2017); Fang et al. (2017); Big Data Working Group (2016); NIST (2015a); Cloud Security Alliance (2013) | 6 |
| Conformance | Soria-Comas and Domingo-Ferrer (2016); Hoffman (2018); Yu (2014); NIST (2015a); Perera et al. (2015) | 5 |
| Compliance | Yu (2014); Benjelloun and Lahcen (2015); Gudivada et al. (2015); Alka and Khan (2018); Hoffman (2018); Cloud Security Alliance (2013); Big Data Working Group (2016); European Commission (n.d.) | 8 |

### 4. Persistence

| Scalability | Chen et al. (2014b); Cui et al. (2014); Hashem et al. (2015); Zhang et al. (2017); Tao and Gao (2016); Che et al. (2013); Hu et al. (2014); Katal et al. (2013); Chen and Zhang (2014) | 9 |
| Automation | Immonen et al. (2015); Rathika and Arcokiam (2014); Taleb and Serhani (2017); Tao and Gao (2016) | 4 |
| Interoperability | The Open Group (2011); Cameron and McMillan (2013); Cloud Standard Customer Council (2017); Soares (2015); Kune et al. (2016); Ramaswamy et al. (2013); NIST (2015b); Orit (2013); Pääkönen and Pakkala (2015); Demchenko and Membrey (2014); Gudivada et al. (2016); Immonen et al. (2015); Moreno et al. (2018); Doug (2013); Oracle (2013) | 15 |
| Extensibility | Soley (2000); Zhang et al. (2017); Casale et al. (2015); Guerriero et al. (2016); Artac et al. (2016) | 5 |
| Agility/Flexibility | Frankova et al. (2016); Early (2014); Chen et al. (2016); Nancy et al. (2017) | 4 |

### 5. Validity

| Requirements | Fabijan et al. (2017); Deng et al. (2013); Gupta et al. (2018); Dimitriev et al. (2016); Mattos et al. (2017a); Schermann et al. (2018); Kohavi et al. (2013); Mattos et al. (2017b); Fabijan and Dimitriev (2017); Kevic et al. (2017); Tang et al. (2015); Kaufman et al. (2017); Xu et al. (2015); | 13 |

| Validity | Fabijan et al. (2017); Deng et al. (2013); Gupta et al. (2018); Dimitriev et al. (2016); Mattos et al. (2017a); Schermann et al. (2018); Kohavi et al. (2013); Mattos et al. (2017b); Fabijan and Dimitriev (2017); Kevic et al. (2017); Tang et al. (2015); Kaufman et al. (2017); Xu et al. (2015); | 13 |

**Total count (without duplicates)**: 111
According to table 2, the proposed *Quality-driven Big Data Application Taxonomy* is visualized in figure 9. The taxonomy consists of two high-level dimensions denoted as QoC and QoD. The term specifications are already provided in chapter 2. Since this work focalizes quality design attributes (i.e. QoD), QoC is only shortly presented. However, QoD should ensure that user needs and QoC-attributes are met and maintained, based on Big Data application design features and considerations.

![Figure 9: The Quality-driven Big Data Application Taxonomy](image)

### 4.1. Quality of Conformance

As already mentioned, this taxonomy sets its focus on quality design attributes of Big Data applications. On that account, QoC is only shortly presented in order to deliver a comprehensible classification on data quality in Big Data applications. As already presented in chapter 2.2.1, QoC can include different quality dimensions and corresponding attributes. As we will see within the attribute *Contextual-Requirements Completeness* presented in the next chapter, data quality is use-case dependent. Hence, providing a holistic set of quality attributes might be impossible. To this end, it sounds rationale to qualify those attributes to be incorporated in this taxonomy, which are mainly addressed in academics, because these
attributes might be of overall importance. This argument must be proven from two perspectives, namely from the viewpoint of (i) traditional data and (ii) Big Data.

Considering traditional data quality attributes, chapter 2.2.1 previously described some of the most cited data quality attributes. Scannapieco and Cartarci (2002) present those attributes that are denoted as the most discussed, including accuracy, completeness, consistency, and timeliness. The authors even identified two additional attributes, interpretability and accessibility, but as Batini et al. (2009) explain, accuracy, consistency, completeness, and timeliness were addressed by the majority of researchers.

From the perspective of Big Data, there is common consensus that traditional attributes also hold for Big Data (Becker & McMullen, 2015; Juddoo, 2015). Consequently, these attributes could provide a common set of data quality attributes for Big Data. However, it seems that Big Data has revealed an additional quality attribute, that is trustworthiness/credibility. In traditional settings, data producers and consumers are often one and the same entity. This largely does not hold for Big Data (Cai and Zhu, 2015). If data are collected from unknown producers (or data owners), the trustworthiness or credibility of data is highly questioned. As one will see in the following quality design descriptions, trustworthiness plays a major role in Big Data.

Figure 10: Quality of Conformance
In summary, it seems legit to define a basic set of QoC-attributes by Completeness, Correctness/Accuracy, Consistency, Currentness/Timeliness, and Credibility/Trustworthiness. Indeed, those are only denoted as general QoC-attributes. As we will see in the next chapter, the QoD-attribute Contextual-requirements Completeness addresses the problem of varying data quality requirements which must be considered within application development.

### 4.2. Usability

*Usability* describes design features and considerations a Big Data application might incorporate in order to be denoted as “usable”. In particular, *Contextual-requirements Completeness* and *User acceptance* should base design efforts on data and users. Additionally, *Metadata Provision* and *Data Fitness* include features that will serve understandability and suitability of Big Data in alignment with company-specific Big Data quality requirements.

![Usability Diagram](image)

*Figure 11: Usability*

#### 4.2.1. Contextual-Requirements Completeness

*Contextual-Requirements Completeness* should ensure that all data quality requirements are met within application design. Since quality is use-case dependent, meeting individual requirements in development phases is vital. Otherwise, the application might not be able to handle data appropriately, which, in turn, could affect Big Data application usability.
Within the identified set of literature, there are various papers found that directly present quality attributes of Big Data. These quality attributes mostly differ from each other, as shown in table 3. As stated in the last chapter, traditional quality attributes, like accuracy, completeness, or consistency also hold for Big Data (Becker & McMullen, 2015; Juddoo, 2015). On the contrary, Big Data is use-case dependent (Juddoo, 2015) and hence, it might be required to either add more dimensions (Ardagna et al., 2018; Firmani et al., 2016) or focus on specific dimensions that are more relevant for a specific application (Juddoo, 2015). Ramaswamy et al. (2013) for example state that even in the same use case, there might be different quality dimensions and give an example of two different sensors, namely sonic anemometers and mill anemometers, which have distinct quality needs. Due to the reliance on individual use cases, it is hard to define a holistic Big Data quality model and therefore seems to be meaningless (Firmani et al., 2016).

Table 3: Differences in Big Data Quality Dimensions

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Availability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Completeness</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Consistency</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Contextual Consistency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Credibility</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Integrity</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operational Consistency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Presentation Quality</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relevance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reliability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temporal Consistency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Timeliness</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Trust</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Uniqueness</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Validity</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Regarding the definition of context-specific data quality requirements, two standards can be mentioned. DAMA’s (Mosley, 2010) DMBOK states that data requirements must be defined in terms of meeting company-specific requirements based on business processes. ISO/IEC 8000-61 (International Organization for Standardization, 2016a) underlines that requirements and expectations of all stakeholders should be refined into data requirements. Those requirements should be investigated with respect to their feasibility, prioritized and balanced within different needs in an enterprise to achieve a holistic set of individual data requirements.

In a Big Data context, an interesting paper is provided by Caballero et al. (2014). Figure 12 shows influences and dependencies between processes, data, software and usage. Processes influence data characteristics which, in turn, influence software characteristics that finally influence usage. This strengthens the importance of defining application software components on data quality attributes.

![Figure 12: Quality Life-cycle (Caballero et al., 2014)](image)

Additionally, Noorwali et al. (2016) present interdependencies between data quality attributes and Big Data system development. In particular, they suggest a requirements specification technique in Big Data environments. The idea is to permute Big Data characteristics (V’s) with traditional data quality attributes (e.g. availability). Big Data characteristics should be satisfied by using common quality attributes. For example, Veracity of Big Data can be achieved by considering security requirements as quality attributes (e.g. by using encryption). One Big Data characteristic can be explained through more quality attributes (or also zero), while not all quality requirements can be related to each Big Data characteristic.

In short, quality attributes of Big Data often differ from each other in existing literature. I think this is natural since Big Data is use-case dependent and therefore, it seems impossible to synthesize all quality attributes in an “one-fits-all” compilation. Application-individual data quality requirements specification is crucial for application design, otherwise the application might miss to address important quality needs, which could lead to an overall low-quality.
4.2.2. Metadata Provision

*Metadata Provision* supports the understandability of data structure, quality, and provenance by delivering additional information about data. This might be especially useful in Big Data environments where data is often collected from unknown sources and transferred multiple times leading to “interpretation deficits”.

Metadata highly influences Big Data processing but is still not well addressed in Big Data projects (Kulkarni, 2016). According to a survey of The Data Warehousing Institute (TDWI), 19% of all respondents claim a “lack of metadata and schema in some big data” (Russom, 2013, p. 10). In general, metadata can be denoted as “data about data” that support the extraction and processing of data features (Immonen et al., 2015, p. 2030; Mosley, 2010, p. 259; Serhani et al., 2016, p. 4). There is a broad consensus that metadata is required to provide information about the following topics:

- **Information about data quality** (Immonen et al., 2015; Kulkarni, 2016, Ardagna et al., 2018, Kläś et al., 2016, Serhani et al., 2016, Supriya and Devendrasingh, 2017)
- **Information about data provenance or trustworthiness** (Immonen et al., 2015; Kulkarni, 2016, Smith, 2014, Ardagna et al., 2018, Serhani et al., 2016)
- **Information about technical details or schemata** (Smith, 2014, Immonen et al., 2015, Kläś, 2016)

From a data quality perspective, metadata should deliver a holistic quality overview to end-users (Ardagna et al., 2018). It serves an important means to foster understandability or interpretability of data quality by describing quality levels of individual defined data quality attributes as well as their resulting metrics (Immonen et al., 2015; Kulkarni, 2016; Serhani et al., 2016). In doing so, it should support users within their quality evaluation processes wherever there is an “interpretation deficit” (Supriya and Devendrasingh, 2017; Wahyudi et al., 2018). Furthermore, literature suggests that metadata should be stored and easily accessible for end-users (Ardagna et al., 2018), produced by the data provider itself in order to get a sense of initial quality levels (Kläś et al., 2016) and be available throughout the whole Big Data value chain (Kulkarni et al., 2016). A comprehensive metadata creation process along the Big Data value chain is presented by Immonen et al. (2015) and can be seen in figure 13. They focus their work on how data quality can be supported with metadata and provide an architecture which continuously considers data quality through metadata. Based on high accessible and
available metadata, different values can be assigned to different quality attributes dependent on metadata. This allows context-independent data quality assessment.

![Diagram of Quality Metadata Creation across the Big Data Pipeline (Immonen et al., 2015)](image)

**Figure 13: Quality Metadata Creation across the Big Data Pipeline (Immonen et al., 2015)**

But there is also a downside in Big Data metadata. The problem is that Big Data was initially developed without any support of metadata while it is largely deployed in traditional environments (Smith et al. 2014). Especially in Big Data technologies like Hadoop, in which data is commonly stored schema-less, descriptions of data (i.e. what it really is) are missing (Kulkarni, 2016). The difference to traditional environments is that while metadata is “native” within structured databases, this is not true for Big Data (Kulkarni, 2016).

In summary, metadata within Big Data applications should provide information about where the data came from, what initial quality the data has and how it is structured. This information should serve increased understandability and interpretability of data which could lead for example to decreased user trainings, costs, and time.

### 4.2.3. Data Fitness

*Data Fitness* addresses the need to increase the suitability of external collected data. It can be described as data adequacy for further processing activities, e.g. analytics. Usually, data fitness is also addressed in traditional standards like ISO/IEC 8000-61 (International Organization for Standardization, 2016a), where internal data should be processed to meet requirements. However, in a Big Data context there are additional aspects to consider.
Due to Big Data Variety, collected data is often embodied by diverse data sources (Chen et al., 2014b). These data items can have various quality levels with respect to redundancy, consistency, or noise (Chen et al., 2014b; Serhani et al., 2016). Because external collected data cannot be controlled in accordance with company-internal data specifications, consistency and validity of data is critical and must be defined for the intended purpose (Gudivada et al., 2015). Additionally, some analytical approaches might imply serious data quality requirements (Chen et al., 2014b; Serhani et al., 2016). To make heterogenous data “fit for purpose”, pre-processing activities must be in place. This will lead to data that correspond to company-individual data quality requirements and hence, will lead to lower storage expenses and higher analysis accuracy (Chen et al., 2014b). Common pre-processing activities are displayed in table 4 according to Taleb et al. (2015) (p.4).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data integration</td>
<td>Combining data from different sources and providing a unified view of data as well as coherent storage.</td>
</tr>
<tr>
<td>Data enhancements and enrichment</td>
<td>Processes for data value improvement (e.g. data integration or data fusion).</td>
</tr>
<tr>
<td>Data transformation</td>
<td>The activity of normalizing and aggregating data and is considered in data migration between systems.</td>
</tr>
<tr>
<td>Data reduction</td>
<td>Reduced data views without any impact on analytics results (activities like clustering, compression, etc.).</td>
</tr>
<tr>
<td>Data discretization</td>
<td>Separate continuous attribute range into intervals in case of applying algorithms accepting just categorial variables.</td>
</tr>
<tr>
<td>Data cleansing</td>
<td>Processes including the searching, the identification, and the correction of data errors.</td>
</tr>
</tbody>
</table>

Regarding the integration of pre-processing, the papers from Taleb et al. (2015) and Serhani et al. (2016) can be mentioned.

Taleb et al. (2017) propose a Big Data Pre-Processing data Quality (BDPQ) framework which consists of the key components profile selection, adaption, and quality control and monitoring (p. 5). In short, profile selection offers different data quality rules which are based on quality requirements, data types, data domains and quality dimensions. The quality rules are selected and combined to a quality profile that defines an individual set of quality rules to achieve quality
requirements. Based on that rules, pre-processing takes place (e.g. data integration, cleansing, etc.). If pre-processing is accomplished, quality control is responsible to validate data quality after pre-processing phases. The authors evaluate their model on a large electroencephalogram (EEG) dataset. Results indicate the importance of considering Big Data quality in early phases of the Big Data value chain.

Serhani et al. (2016) present a hybrid quality evaluation approach for Big Data across the Big Data value chain. The approach consists of four steps, namely (i) pre-Big Data quality evaluation, (ii) pre-processing quality evaluation, (iii) post-Big Data quality evaluation, and (iv) processing and analytics quality evaluation. Pre-processing must be applied to improve the quality of data. Quality assessment should take place before and after pre-processing. The authors conclude that the earlier data quality is addressed, the better the quality will be in remaining Big Data value chain phases. Furthermore, data quality must be addressed in each phase of the Big Data value chain.

Big Data pre-processing is vital for data quality. It can be seen as the activity to make data “fit for purpose”. If pre-processing is not applied, redundant or inconsistent data could be stored, which lead to higher storage costs and lower analysis accuracy. Pre-processing could be especially important if data is collected from various non-proprietary sources where quality requirements cannot be influenced during data creation.

4.2.4. User Acceptance

User acceptance is closely related to Contextual-requirements Completeness but addresses design issues from a different angle. As already explained, Contextual-requirements Completeness should be responsible for application design according to a complete set of context-specific data requirements. In contrast, User Acceptance should deal with the question of how an application must be designed in order that end-users will accept and further use it. If the Big Data application is not accepted by users, it won’t be usable.

In general, Big Data highly relies on its end-users and a combination of various technologies and techniques including high investments in software, hardware, and training (Khan and Brock, 2017). Since the usefulness of Big Data is often nebulous, enterprises could spend a lot of money for Big Data technologies without any usage from end-users if they perceive the application as not useful (Khan and Brock, 2017). In other words, deciding on software components within the application could highly rely on how useful end-users will perceive
those components. However, to measure user acceptance, research in Big Data-related literature is often based on technology adoption, like the Technology Acceptance Model (TAM) from Davis (1989):

For example, Abbasi et al. (2016) mention TAM and state that such models support impact analysis of the perceived usefulness and ease of use on the behavioral intention using Big Data tools. They stress Big Data V-characteristics for the description of this impact. For example, Variety of data sources could have a negative effect on perceived usefulness and ease of use due to increasing complexity when collecting various data types. Additionally, Veracity could imply such negative effects if trustworthiness is low. Consider Big Data V’s will help to define applications in a way that could counter negative effects on user acceptance. In the case of Veracity for example, trusted-systems are of overall importance.

Shin (2016) evaluates the implementation of Big Data technologies by modifying Unified Technology Acceptance and Usage Theory (UTAUT) (p. 2) for a better description of Big Data acceptance. Among others, he defines and tests the hypothesis of how Big Data application features like quality, security or interoperability affect the intention to implement Big Data, which, in turn, affect usage behavior (e.g. perceived quality can be fostered by ensuring high-quality data). Based on survey and interpretation methods, results show that Big Data technologies should be integrated with a user-centered focus considering the Big Data design.

Khan and Brock (2017) also examine the usage and acceptance of Big Data in enterprises and how user perceptions affect the initiation of Big Data applications. They state that Big Data is often promoted as a set of commercial products. In reality, it is not a so-called “off-the-shelf” product. Rather more it is a combination of various technologies and techniques involving high investments in software, hardware, and training. Therefore, it is highly critical to involve user acceptance in Big Data application development.

In summary, Big Data is a phenomenon which is strongly technology- and user-driven. Therefore, user acceptance might be essential and should be considered in Big Data initiations. Involving users in design phases could lead to higher usability, because acceptance will grow and thus, Big Data applications are used. High acceptance, which could also mean higher motivated users, might also affect data quality by offering quality-affecting components in an easy way to use. Enterprises should especially focus user acceptance within those application components that must meet multiple user needs. Addressing Big Data V-characteristics will help defining “acceptance-risk-areas”.

37
4.3. Security

Security ensures that (non-personal) data is protected from unauthorized access, modification and deletion, and that data is timely available to authorized entities. Additionally, data accountabilities should be integrated into the application to track inappropriate behavior and an enterprise must hold accountable for appropriate data security, which, in turn, affects trustworthiness.

Figure 14: Security

The techniques and models described in the following should address the ten key challenges defined by the Cloud Security Alliance (CSA) (2013), which are widely accepted in literature (Serrão and Moura, 2016; Gaddam, 2015; Samsudin, 2016; Terzi et al., 2015).

4.3.1. Confidentiality

According to ISO/IEC 27000 (International Organization for Standardization, 2016b), Confidentiality is the “property that information is not made available or disclosed to unauthorized individuals, entities, or processes” (p.4). Confidentiality requirements must be defined in accordance to company’s needs (International Organization for Standardization, 2005). It is important to specify authorization for information usage, classify information according to their sensitivity and value, and ensure user authentication for all users (International Organization for Standardization, 2005).
In Big Data, confidentiality is typically achieved by using a data-centric encryption (Samsudin, 2016; Serrao and Moura, 2016). Additionally, authentication is used to control the access to data (Joshi and Kadhiwala, 2017). Cryptographic techniques (i.e. encryption techniques), which are the most popular in achieving confidentiality (Duhan and Singh, 2018), are valuable means for preventing data from unauthorized access and thus, protect data ownership across the data lifecycle (Abouelmehdi et al., 2018). The different techniques for ensuring confidentiality through encryption and authentication can be seen in table 5. The highlights of those techniques and models are presented in the following:

Security concerns should be data-centric (Manikandakumar and Ramanujam, 2018; Samsudin, 2016; Serrao and Moura, 2016) using different encryption techniques like homomorphic encryption, searchable encryption, attribute-based encryption, functional encryption, systematic encryption, (Manikandakumar and Ramanujam, 2018; NIST, 2015a; Samsudin, 2016; Serrao and Moura, 2016) or relational encryption as an alternative to homomorphic encryption (Big Data Working Group, 2016) in order to ensure data protection regardless where the data resides (Samsudin, 2016).

Table 5: Classification of Confidentiality Models and Techniques

<table>
<thead>
<tr>
<th>Class</th>
<th>Sub-class</th>
<th>Description</th>
<th>Techniques</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Models</td>
<td>Abouelmehdi et al., 2018; Chaudhary and Kumar (2017); Zainudeed et al. (2014); Kert (2014); NIST (2015a)</td>
</tr>
</tbody>
</table>
Environmental trust should be achieved by ensuring continuous server and user authentication, e.g. by Transport Layer Security (TLS) and Secure Sockets Layer (SSL) (Abouelmehdi et al., 2018; NIST, 2015a), mc-TSL as an useful alternative to TSL (Chaudhary and Kumar, 2017), Kerberos (Mattsson, 2014; Big Data Working Group, 2016; Abouelmehdi et al., 2018; Kumar et al., 2015), SESAME (Tsaur and Wu, 2013) or similar. The sensitivity of data should be considered in cloud storage and computations (e.g. by using HybrEx) in order to protect high sensitive data (Jain et al., 2016; Abouelmehdi et al., 2018). Databases should be partly encrypted in order to speed up analytical processes (Toshniwal et al., 2016).

According to the literature above, confidentiality should use a data-centric encryption to ensure secure computation across distributed programming frameworks. Additionally, authentication techniques should ensure that data requests are authenticated assuring secure communication and establish an overall trusted platform.

### 4.3.2. Integrity

According to ISO/IEC 27000 (International Organization for Standardization, 2016b), *Integrity* is the “*property of accuracy and completeness*” (p.7). In other words, it is the property to protect data and system applications from unauthorized deletion (affect completeness) and modification (affect accuracy) of data (Alshafi and Saife, 2017). Therefore, integrity can be on a data-level or software-level. Ward and Barker (2013) state that integrity and quality of data are crucial in drawing valuable conclusions in the decision-making process. Most common attacks regarding integrity are man-in-the-middle attacks, salami attacks, diddling attacks or trust relationship attacks (Duhan and Singh, 2018). Integrity techniques and models are presented in table 6. Key-concepts are highlighted in the following:
ISO/IEC 27002 (International Organization for Standardization, 2005) emphasize integrity through access control. Access control should be defined and implemented for information processing of users, customers and external parties. Additionally, information exchange must be specified in terms of integrity requirements and user activities should be controlled with the help of audit logs.

In a Big Data context, a large body of knowledge also suggests using access control for ensuring integrity (Abouelmehdi et al., 2018; Stojkov et al., 2017; Chen et al., 2014a; NIST, 2015a; Big Data Working Group, 2016). Generally, after a user is authenticated, he has access to the data. The magnitude of access is controlled by different access control techniques (Abouelmehdi et al., 2018). Solutions like Role-based Access Control (RBAC) or Attribute-based Access Control (ABAC) are popular techniques to protect data from unauthorized deletion or modification (Abouelmehdi et al., 2018). RBAC grant access based on individual roles. ABAC uses characteristics of users, resources or environments in order to set up access policies (Stojkov et al., 2017). In this respect, Stojkov et al. (2017) compare different access control techniques, namely RBAC, ABAC and a new technique called “Capability-based Access Control” (CapBAC). They conclude, that RBAC and ABAC are inflexible and difficult to handle. CapBAC on the other side provides a good basis for application in an IoT-environment (Stojkov et al., 2017). Chen et al. (2014a) present a multilabel-based scalable access control model for data-sensitive applications. This model defends sensitive data in the Hadoop Distributed File System (HDFS). The granularity of access control varies with the quantity of multilabel as well as their content. Scalability is achieved with the labels. Data owners can selectively use labels and system designer and administrators can revise, add or delete labels depending on the security requirements of their application (Chen et al., 2014a). Xu and Shi

### Table 6: Classification of Integrity Models and Techniques

<table>
<thead>
<tr>
<th>Class</th>
<th>Sub-class</th>
<th>Description</th>
<th>Techniques/Models</th>
<th>Approaches</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrity</td>
<td>-</td>
<td>Property to protect data and system applications from unauthorized deletion (affect completeness) and modification (affect accuracy)</td>
<td>Techniques: Role-based access control, attribute-based access control, capability-based access control, TLS/SSL connections, End-point input validation, granular audits based on logging, Digital signature techniques, query integrity protection</td>
<td>Models: Multilabel-based scalable access control model from Chen et al. (2014a), Trusted Platform Module (TPM)</td>
<td>Abouelmehdi et al. (2018); Big Data Working Group (2016); Stojkov et al. (2017); NIST (2015a); Xu and Shi (2016); Cloud Security Alliance (2013); Cloud Security Alliance (2013)</td>
</tr>
</tbody>
</table>
(2016) present a high-class collection of integrity techniques for Big Data applications. They describe digital signature techniques, query integrity protection and storage integrity protection. For example, digital signatures are used for Big Data processing. Classical digital signatures and Mandatory Access Control (MAC) ensure integrity but come with high costs (due to the required public key infrastructure which is expensive and not scalable) and low traceability (integrity can be only verified at the last step). Another problem is sharing secret keys between parties. This problem can be addressed by using homomorphic signatures which only need public key instead of private key sharing (Xu and Shi, 2016).

NIST (2015a) emphasizes communication integrity, i.e. integrity of data in-transit. This can be achieved by the prior mentioned TLS/SSL connections in order to foster trust across the distributed network (NIST, 2015a; Abouelmehdi et al., 2018). To maintain secure data movements, end-point input validation should be used (Big Data Working Group, 2016; NIST, 2015a). Trusted Platform Module (TPM) is suggested in order to ensure trusted device data (Big Data Working Group, 2016; NIST, 2015a), although they do not ensure total protection (Cloud Security Alliance, 2013).

Granular audits are also important in order to identify attacks that are for example overlooked by real-time security monitoring (Cloud Security Alliance, 2013). To do so, logging is proposed. Logging means recording for example all map-reduce jobs as well as information about the users doing those jobs. Those records should be audited on a regular basis (Kumar et al., 2015). In Hadoop, built-in tools for monitoring of potential malicious queries or misuse are still non-existent (Kanyan and Mehra, 2018).

In summary, integrity can be achieved by using access control, communication integrity through secure communications and granular audits based on log files. In other words, access control and communication integrity should warrant data integrity, regardless where the data resides (i.e. data-in-transit and data-at-rest). Additionally, logging of data handling processes should be implemented in order to conduct granular audits on a regular basis. End-point input validation is also suggested but will be focused in the Provenance attribute.

4.3.3. Availability and System Immunity

According to ISO/IEC 27000 (International Organization for Standardization, 2016b), Availability is the “property of being accessible and usable upon demand by an authorized entity” (p.3). In other words, it should be defined how acceptable use looks like and afterwards
it must be implemented (International Organization for Standardization, 2005).Availability refers to data, applications as well as physical components (Alshafi and Saife, 2017). In the fast-paced environment of Big Data it is essential to provide required data on-time wherever and whenever it is requested (Duhan and Singh, 2018). Popular attacks against availability are Denial of Service (DoS), Distributed Denial of Service (DDoS) and SYN attacks (Duhan and Singh, 2018; Joshi and Kadhiwala, 2017). Techniques and models are classified in table 7. Once again, the most important concepts are presented below:

Table 7: Classification of Availability and System Immunity Models and Techniques

<table>
<thead>
<tr>
<th>Class</th>
<th>Sub-class</th>
<th>Description</th>
<th>Techniques/Models</th>
<th>Approaches</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability and System Immunity</td>
<td>Availability</td>
<td>Property of ensuring timely available data</td>
<td>Techniques</td>
<td>High Availability (HA) systems</td>
<td>Joshi and Kadhiwala (2017); Kune et al. (2016)</td>
</tr>
<tr>
<td>System Immunity</td>
<td></td>
<td>Apply Big Data for security purposes</td>
<td>Techniques</td>
<td>Applying advanced analysis in order to identify potential vulnerabilities, threats, etc.</td>
<td>Mishra and Singh (2016); Joglekar (2016)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Models</td>
<td>Security Intelligence</td>
<td>Mishra and Singh (2016); Joglekar (2016)</td>
</tr>
</tbody>
</table>

High Availability systems are proposed in order to satisfy data availability (Joshi and Kadhiwala, 2017). High Availability systems provide availability by offering multiple components and access points to data in case that any of the components or access points fail (Kune et al., 2016). Since the advent of cloud solutions, availability becomes a minor problem (Subbalakshmi and Madhavi, 2018).

System immunity refers to the application of Big Data for security purposes, like real-time security monitoring. The keyword in this context is “Security Intelligence”. In particular, large amounts of non-relational data are collected in order to identify potential vulnerabilities, threats and security events for the purpose of fast elimination (Astya et al., 2016). For example, Joglekar and Pise (2016) describe typical cyber threats, like DDoS attacks, phishing, Trojan attacks, SQL injection, Zero-day attacks and others. They stress the usefulness of mining information that will serve a broader view on vulnerabilities and risks. They also describe how Big Data can support security and privacy, e.g. by applying machine learning algorithms, ability to mine semi- and unstructured data, ability to store large amounts of security-related information in a Big Data environment and so on. Finally, they also present some existing Big Data related tools for cybersecurity, like Fortscale or IBM Security QRadar (Joglekar and Pise,
Akutota and Choudhury (2017) additionally suggest using advanced analytics (like machine learning) for detecting flawed behavior of the company-internal users that can lead to internal threats, vulnerabilities or others. First, an User Behavior Analysis (UBA) tool determine a baseline of normal user behavior. Second, deviations to the normal behavior are searched in almost real-time. Thus, it is a comprehensive method for identifying risks in near real-time (Akutota and Choudhury 2017).

All in all, availability is crucial for Big Data Velocity. It should be achieved by using High Availability systems, which can provide the required information in case of one access point fails. The use of cloud solutions is also valuable for the fast retrieval of data (Cloud solutions will be addressed later). Applying Big Data for security can lead to security monitoring in near real-time. This could be especially important in Big Data applications which continuous collect and analyze streaming data, like data from sensors.

**4.3.4. Accountability**

Accountability stresses two needs: First, there is a need for roles and responsibilities, especially for sensitive data. This is important for application design because integrating roles and responsibilities enable the control of data usage and allow to track inappropriate behavior. According to ISO/IEC 27002, roles must be defined and documented to protect information assets from un-authorized access, modification or destruction (International Organization for Standardization, 2005). This is closely related to Integrity described above. Second, an enterprise should foster application trustworthiness by designing the application in a way that it can provide non-repudiation to security standards. In other words, it holds accountable for appropriate data security.

Regarding the first need, DAMA DMBOK includes a chapter about data security. According to DAMA, role privileges should be defined through the identification of role groups. This should be granted by enrolling specific roles into specific role groups. They should be hierarchically defined and require reporting tools able to drill down to user privileges on a granular level. If somebody changes roles or responsibilities, this should require approval and tracking in order to trace back to the responsible person. Additionally, user roles and responsibilities should be stored centrally in heterogeneous environments in order to prevent multiple versions of “truths” (Mosley et al., 2010). It is noteworthy, that this is a traditional standard without focusing on Big Data. The applicability in Big Data is therefore not validated.
Mattsson (2014) proposes a security methodology based on data sensitivity. They explain, that security policies must be integrated based on the least-privilege principle. This principle implies that a user should receive the least possible set of sensitive data required to perform a specific job function. Such a concept can be embedded by integrating roles according to the policies. Furthermore, responsibilities are needed to prevent abuses of privileged users, which should not have those privileges (e.g. because of inappropriate distinctions between data security professionals and data management professionals). Both, roles and responsibilities, should be considered in application design in order to define the right access controls, audits, etc.

Considering the second need, ENISA (2015) elucidate that trustworthiness is one of the biggest challenges within Big Data. In order to support trustworthiness of applications, enterprises must design their security based on different standards (e.g. on ISO/IEC). This non-repudiation of an appropriate security foundation could lead to increased overall trust. Nowadays, there are also more cost-effective and flexible ways to provide security standard compliance.

Commonly, accountability is largely part of data governance by defining and assigning clear roles and responsibilities for users and hence, seems to be not that important in application design. But the implementation of roles and responsibilities within the application is valuable for access control or audits and therefore should be also addressed. Application design should especially consider accountability for sensitive data. If the application would not be able to store, control, and track roles and responsibilities, this could quickly lead to an application where everyone can view, modify, or delete sensitive enterprise data, regardless if access control is in place or not. Additionally, if the enterprise holds accountable for data security by providing non-repudiation to specific security standards, trustworthiness of third parties will be increased.
4.4. Privacy

*Privacy* describes requirements an application must satisfy with respect to personal data protection and usage. Privacy and Security are highly related, especially the security attributes *Confidentiality* and *Integrity* are equal to the privacy attribute *Non-disclosure*, at least regarding existing models and techniques. In this taxonomy, the difference between the terms is that while non-disclosure should ensure data confidentiality and integrity of Personal Identifiable Information (PII), confidentiality and integrity at security refer to non-PII. This argument can be underpinned by looking at a general definition of privacy, which also focuses on PII. Privacy can be defined as “the rights and duties of individuals or institutions about the collection, use, disclosure, retention and disposal of personal information” (Fang et al., 2017, p. 547).

![Figure 15: Privacy](image)

4.4.1. Non-disclosure

*Non-disclosure* is highly allied to *Confidentiality* and *Integrity* at security. According to Jain et al. (2016), confidentiality and integrity are directly related to the privacy of data. This means, breaching confidentiality and/or integrity directly affects a user’s privacy. Therefore, confidentiality and integrity are prerequisites for ensuring privacy, especially non-disclosure. Due to this fact, techniques and models presented in the confidentiality and integrity parts are also valid in the sense of non-disclosure.
At this point, one might think about the meaningfulness of separating non-disclosure from confidentiality and integrity, because of their high level of similarity. In my opinion, the difference can be defined by its focus and resulting terminology. In general, security is about defending organizational information assets by using different technologies, processes, etc. Defending means protecting information assets from un-authorized access, disruption, inspection, modification, destruction or recording (Jain et al., 2016). Data can be denoted as secure if confidentiality, integrity and availability (CIA) are ensured (Duhan and Singh, 2018; Joshi and Kadhiwala, 2017). Contrary to security, privacy is not about company-internal data or information. Rather more, it sets its efforts on protecting PII. The term PII is often used in NIST (2015a). This means, how personal information is collected, used and protected in order that only those people can identify personal information that have received those information (Jain et al., 2016). Because of the different focus area, it seems meaningful to adjust the terminology. The resulting key-term in this taxonomy is therefore Non-disclosure and should describe efforts of protecting PII through confidentiality and integrity techniques and models. Indeed, if PII access an enterprise, it might be also denoted as organizational information assets, but they must receive special treatment. This treatment is expressed through the other attributes in the privacy dimension.

Taken all together, non-disclosure should be achieved by using confidentiality and integrity models and techniques. The difference is that those models and techniques should be used to protect PII from non-disclosure.

4.4.2. Pre-serving

Generally, collecting PII for any purpose can imply big privacy challenges (Terzi et al., 2015). The problem is, if sensitive information about individuals would go into wrong hands, this can have severe consequences for an enterprise (Jhaveri et al., 2015). To prevent personal information to be identified, privacy pre-serving must be considered. Privacy pre-serving means hiding PII (Duhan and Singh, 2018) that it cannot be identified by users or third-parties. This is especially important in data analytic activities where PII is analyzed for organizational purposes (Jhaveri et al., 2015). To ensure privacy preserving, data anonymization/de-identification must be applied (NIST, 2015a). In particular, PII must be sanitized based on generalization (replacing quasi-identifiers) and suppression (some values are not released at all) before data mining takes place (Jain et al., 2016). Data anonymization can be defined as “the process of modifying
personal data in such a way that individuals cannot be reidentified and no information about
them can be learned” (Bisht and Singh, 2016, p. 146).

According to Fang et al. (2017), privacy pre-serving largely considers two aspects: (i) How to ensure that privacy is not leaked within the application process and (ii) How to improve the application usage (Benjelloun and Lahcen, 2015). Privacy pre-serving challenges mainly affect system performance and data quality. For example, traditional anonymization/de-identification techniques are time-consuming and based on a number of iterations, which could negatively affect the consistency of data as well as system performance, especially when considering Big Data Volume (Benjelloun and Lahcen, 2015). Fully anonymization without affecting data set utility is difficult. However, there are a number of techniques for privacy-preserving in literature. For example, the Big Data Working Group (2016) provides 100 best practices for security and privacy in Big Data. Those 100 best practices are structured in ten chapters. Chapter 6 is about “scalable and composable privacy-preserving analytics” (p. 32 ff.). They present approaches like differential privacy, homomorphic encryption, authorization, encryption for data-at-rest, etc. Other papers explain that anonymization/de-identification can largely be achieved by various techniques like k-anonymity, l-diversity, or t-closeness (Jhaveri et al., 2015; Soria-Comas and Domingo-Ferrer, 2016; Abdouelmehdi et al., 2018; Jain et al., 2016; Xu and Shi, 2016). The three techniques are shortly defined in table 8.

Table 8: Anonymization/De-Identification Techniques

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-anonymity</td>
<td>“k-Anonymity seeks to limit the disclosure risk of data set by limiting the capability of intruders to re-identify a record, that is, k-anonymity seeks to prevent identity disclosure” (Soria-Comas and Domingo-Ferrer, 2016, p. 24).</td>
<td>Not always effective to infer sensitive values in attributes of the data records (Jhaveri et al., 2015).</td>
</tr>
<tr>
<td>L-diversity</td>
<td>L-diversity seeks to address the limitations from k-anonymity by providing diversity of values in sensitive attributes (Xu and Shi, 2016).</td>
<td>L-diversity might be difficult to achieve and is not able to prevent disclosure (Jain et al., 2016).</td>
</tr>
</tbody>
</table>
T-closeness is an improvement of 1-diversity by treating attribute values distinctly, taking into account data value distribution for the attribute values (Abdouelmehdi et al., 2018).

T-closeness requires close distribution between sensitive attribute of equivalent class and the overall table (Jain et al., 2016).

In summary, privacy pre-serving is an essential requirement to ensure that privacy is always achieved, and that no personal information can be identified by users or third parties. Data de-identification should be especially applied during analytical activities in order to counter PII identification.

### 4.4.3. Provenance

Data provenance highly relies on the concept of trustworthiness or credibility. In traditional settings, data producers and consumers are often the same person. This fact mostly does not hold in Big Data (Cai and Zhu, 2015). In other words, if data can be collected from unknown producers, the trustworthiness of data is highly questioned. This, in turn, will affect decision-making processes. Decisions should be based on highly trusted data. If the decision-maker does not know how data values are delivered and if or how they were modified or transformed, results are not trustworthy (Xu et al., 2014; Joshi and Kadhiwala, 2017). In general, provenance can be defined as “the chronology of the ownership, custody or location of a historical object” (Xu et al., 2014, p. 1166). Provenance is listed as one of three future key technologies form the US Department of Homeland Security (Fang et al., 2017).

In literature, provenance must be ensured by balancing consistency between data and their origins (Cloud Security Alliance, 2013; Big Data Working Group, 2016) and by balancing data source and data privacy protection (Fang et al., 2017).

The Big Data Working Group (2016) addresses provenance as one of ten key-concerns. They describe provenance information as metadata and denote data that is based on consistent provenance metadata as highly trustworthy. If provenance and data are inconsistent, data cannot be used with total confidentiality. Furthermore, they state that provenance should be managed by storage systems and should be constructed in a chain. Consistency between data and provenance can be achieved by using hash functions and tables in order to locate provenance information if needed. Other best practices for provenance metadata protection include prior mentioned techniques like authentication, integrity, encryption, or access control.
NIST (2015a) presents their conceptual taxonomy for Big Data security and privacy and treat provenance as one of four main topics. They propose to use end-point input validation (which was already shortly mentioned within Integrity of this taxonomy) in order to verify, if input data come from authenticated sources. End-point input validation can be on a syntactic and semantic level. Additionally, they describe provenance through prior mentioned techniques like communication integrity, authentication, and granular audits.

Within their top ten Big Data security and privacy challenges, the Cloud Security Alliance (2016) covers provenance as the tenth challenge. They also describe provenance through metadata and how to handle those metadata in an application. To do so, sources should be authenticated in a first step. This should be maintained with the help of periodic updates of the provenance status. Provenance metadata accuracy should be ensured by applying integrity checks. Additionally, Volume, Velocity, and Variety of provenance metadata must be ensured. Within the enterprise, it should be secured with access control.

All in all, data provenance is vital for Big Data applications. It must provide information about where the data came from and if or how the data is modified or transformed. This is highly critical if data sources are not known. Immonen et al. (2015) for example state that if data is freely available (which could imply that sources are not known), trustworthiness of data is an essential aspect that must be well addressed. Therefore, provenance is placed within privacy where PII is collected from various different sources and thus, requires validation of data origins.

4.4.4. Conformance

Conformance is a smaller attribute but important to establish trust across data owners. To convey this idea, we have to look at a real-world example. As already stated in prior chapters, Marr (2016a) presents 45 examples of successful Big Data applications. One example is about Amazon. He states “by far the biggest challenge for Amazon […] was getting the public to put their faith in taking part in online commercial activity” (p. 291). Such a challenge could be countered by ensuring security and privacy in general. But what might be also important is to seek consent from the data owners and handle PII in a way that it conforms to data owners’ expectations. This consent must be designed that it is simple, informed, explicit, and specific (Soria-Comas and Domingo-Ferrer, 2016). Following papers address this issue:
Hoffman (2018) presents a Big Data privacy framework to support the understanding of the interaction between Big Data and privacy. Based on a literature review, he identifies four main topics. Beside the other topics, i.e. privacy by design, privacy preservation, and rights and ethics, the fourth topic focuses on consent of the data owner. He states that consent is needed for data collection and usage. This is normally achieved by noticing the data owner or let the owner decide on his own if data can be collected or not. But such a consent will get irrelevant if people do not know what personal information is collected and how it will be analyzed (which could also imply to add additional information to a data owner´s information). Consent should be ensured by informing data owners on what data is collected and how this data will be used.

Yu (2014) analyzes Big Data privacy compliance issues and considerations and defines written and implied rules. While written rules are considered in the next attribute (i.e. Compliance), implied rules can be seen as implicit expectations between data owners and data handlers. For example, if a person downloads a mobile game in an app store, this person will expect that personal information, like an address book, will not be collected and further stored by a data handler. Because laws and regulations are often not updated or generally slow, it seems valuable to consider implied rules for privacy protection. The author suggests a framework that ensures the consideration of implied privacy expectations, including consent, collection limitation, purpose specification and others.

From the viewpoint of application design, NIST (2015a) state that proper consent might be challenged if data receive multiple treatment from various custodians. Especially within Big Data, consent must be augmented through technical controls to ensure accountability for data usage. Perera et al. (2015) state that one of the biggest privacy challenges within IoT is to design and develop technologies which can request user consent in an effective and efficient fashion. Taken all together, Big Data application design should consider data owner expectations which could lead to higher trust of the data owners. This is an important but often neglected requirement within privacy assurance.
4.4.5. Compliance

*Compliance* is highly related to *Conformance*. While conformance should consider data owner expectations, compliance must ensure adherence to existing laws and regulations. Yu (2014) denotes such laws and regulations as “written rules” and describes them as rules that refer to privacy-related laws and regulations an enterprise must comply with. Although, such written rules exist before the advent of Big Data, they must be refined due to Big Data’s special characteristics (Yu, 2014). Providing compliance includes two main challenges. First, companies might require the consideration of multiple, country-specific laws and regulations if an application affects different entities in many countries (Benjelloun and Lahcen, 2015). Second, those written rules might have different requirements which must be considered within Big Data application design (Gudivada et al., 2015). Therefore, it is still a problem providing an individual’s privacy in Big Data (Alka and Khan, 2018). For example, the recent privacy breach on Facebook where around 87 million user accounts were hacked, demonstrates that privacy is far from perfect (Badshah, 2018). Due to its current presence, we will shortly look at the GDPR which was released in May 2018. In general, GDPR has been developed in response to the ongoing growth of data and due the inappropriateness of its predecessor, the Data Protection Directive from 1995 (Yu, 2014).

The official homepage of the European Commission (European Commission, n.d.) defines GDPR from a data owner and a data handler side. Regarding the data handler side, new data handling rules include:

- Enterprises must process personal data in a transparent and lawful fashion
- Data processing is only allowed for a specific purpose
- Only those data should be collected that fulfill this specific purpose
- Personal data reuse is generally not allowed
- Personal data can only be stored as long as the purpose for the collection is in place
- Technical safeguards must ensure confidentiality and integrity of personal data (in this taxonomy denoted by *Non-disclosure*)
- Data owner must be informed if his data is processed, who the company is, why personal data is collected, how long data is stored, etc.

In addition, GDPR protects personal data being transferred outside the European Union (EU), if data does not fulfill specific requirements and provide compliance with EU privacy standards (Yu, 2014).
However, remembering the last attribute *Conformance*, regulations like GDPR are well established and seem to make *Conformance* redundant. But in my opinion, two arguments contradict. First, there might be other countries in the world where regulations about personal data lag. Second, it will be always valuable to grapple with possible data owner expectations that could lead to higher trust compared to other companies. Furthermore, since judicature is generally slow, it might be possible that future expectations are not included.

In order to cope with those regulatory compliance requirements, Hoffman (2018) proposes “Privacy by Design”. Privacy by design claims that privacy assurance is an issue of system development. In other words, privacy should be considered straight from the design phase of a system or an application within a “privacy first”-culture. Considering only security will not be enough for privacy compliance. Some applications might have different privacy requirements, e.g. healthcare. Privacy by design must address topics like how to protect personal data, how it can be linked and analyzed in order that de-identified data will not be re-identified, etc.

The Cloud Security Alliance (2013) and The Big Data Working Group (2016) address the problem of compliance with the prior explained concepts of real-time security monitoring and logging. As already explained, real-time security monitoring uses advanced analytics for security and/or privacy compliance control. It should be able to alert at the time of an attack. Since this cannot be always the case, logging for granular audits is also suggested. Audits are well-established in enterprises, but not on such a granularity and scope like in real-time security monitoring contexts. Audit information require completeness, availability, integrity, and access control.

To sum it up, compliance is a highly essential topic that must be considered straight from the design phase. It is important that companies define how strong the data and the application rely on different country-specific laws and regulations and afterwards, design the application based on a “privacy-first”-rationale. Techniques like real-time security monitoring and logging for granular audits seem to be valuable approaches to counter privacy challenges as well as to ensure high system health and integrity of data.
4.5. Persistence

*Persistence* refers to Big Data application requirements that will continuously expand. This means, requirements that might grow over time and thus, should be also ensured prospectively. (i) *Scalability* refers to the ever-growing amount of data, (ii) *Automation* considers the increasing need to automate processes, (iii) *Interoperability* focuses on the ability to cope with growing interactions between different software, tools, devices, etc., (iv) *Extensibility* should maintain a high-quality Big Data application by offering easy component extensibility in case of changes, and (v) *Agility/Flexibility* must deal with the ongoing fast-paced change of Big Data environments.

![Figure 16: Persistence](image)

4.5.1. Scalability

In this taxonomy, *Scalability* is the ability to provide scalable storage and processing power. Additionally, due to its high relationship to the presented technologies, scalability also refers to the ability to store large amounts of heterogenous data sources. Hence, it addresses the needs of Big Data Volume, Velocity, and Variety.

As an explanation, let’s look at the problem of traditional DBMSs. Traditional DBMSs were intentionally built for structured data and require expensive hardware (Chen et al., 2014b). Within classical architectures, traditional DBMSs can become the bottleneck if they must deal with scalability, because traditional DBMSs cannot easily scale up with additional hardware or machines (Cui et al., 2014). Furthermore, classical data warehouses and analytics can no longer
handle the scale and real-time analysis requirements of Big Data (Cui et al., 2014). For example, Zhang et al. (2017) describe scalability as one out of five key requirements for high quality Big Data applications. Also, Tao and Gao (2016) define quality dimensions for prediction systems and recommendation systems and identify scalability as important requirement for both.

Looking at existing literature, scalability is also required within the context of data quality evaluation. Some papers describe that traditional quality assessment techniques hamper in Big Data environments due to the Volume, Variety, and Velocity of data (Ardagna et al., 2018; Kläs et al., 2016). Considering Velocity for example, data quality cannot be assessed periodically (like in traditional settings) but must often be evaluated in almost real-time (Kläs et al., 2016). This requires high processing power for quality assessment (Cai and Zhu, 2015). Those issues must be considered when applying traditional standards on data quality management, like ISO/IEC 8000-61 (International Organization for Standardization, 2016a), because they miss such aspects.

In a Big Data context, scalability is mostly addressed by (i) cloud computing and corresponding services like Platform as a Service (PaaS), Software as a Service (SaaS) or Infrastructure as a Service (IaaS) (Che et al., 2013; Hashem et al., 2015), (ii) Hadoop environment, especially MapReduce and HDFS (Che et al., 2013; Cui et al., 2014; Hashem et al., 2015; Hu et al., 2014; Katal et al., 2013), and (iii) NoSQL DBMSs with new data models like key-value, document-oriented, column-oriented or graph models (Hashem et al., 2015; Hu et al., 2014; Chen and Zhang, 2014):

Cloud computing can be seen as powerful architecture in order to deploy complex parallel and scalable computing (Che et al., 2013). Hashem (2015) analyzes Big Data in cloud computing and explains that cloud computing removes requirements to maintain the often-expensive hardware and software by delivering convenient, on-demand and ubiquitous access to configurable resources like storage, server, etc. Due to its service provision, companies can focus on other core-businesses without fearing issues including flexibility or availability.

HDFS is designed to store large files with fast data access based on clusters and running on commodity hardware (Katal et al., 2013). HDFS can easily scale-up from one server to numerous machines where each of those machines offer local storage and computation (Hashem et al., 2015). MapReduce, as the name implies, has two different interfaces, namely map and reduce (Cui et al., 2014). Large input data is divided into key-value pairs in a first step. Afterwards, the mapping function is applied and divided into many different instances that concurrently process the key-value pairs. The reduce function finally merges the processed
values according to common keys (Che et al., 2013). There is also the ability to implement individual MapReduce functions (Cui et al., 2014). HDFS and MapReduce are closely related to each other, which means, that the storage and processing system are not separated from each other (Hashem et al., 2015). Other Apache projects were aligned with Hadoop, for example Hive, Pig, Hbase or Spark (Hashem et al., 2015). Hu et al. (2014) state that Hadoop is highly suitable for the management and analysis of Big Data, because it is scalable, cost-effective, flexible and fault tolerant.

NoSQL is an approach to the storage of non-structured (Chen & Zhang, 2014) and distributed data (Hashem et al., 2015). It is schema-free, scalable, fast and reliable (Hu et al., 2014). There are different types of NoSQL databases, namely key-value, document-oriented and column-oriented (Hashem et al., 2015). NoSQL divides storage and management, which is different to relational databases. Storage is generally done by storing different data structures in key-value pairs. Management is provided at a lower-level where management tasks are implemented and thus separated from database-specific procedure languages (Chen & Zhang, 2014).

In summary, scalability can be denoted as a critical requirement, because it addresses Big Data Volume, Velocity and Variety. Scalability should ensure scalable data storage and processing power, e.g. for real-time data quality assessment as well as the ability to store heterogeneous data structures. The presented solutions fulfill these requirements in a flexible and cost-effective way.

### 4.5.2. Automation

In general, the term automation is very widespread. Within this taxonomy, Automation focuses on data quality processes. Automation could be valuable when considering time and costs. Manual processes might take too long and could imply higher failure rates. In literature, automation of data quality processes hardly lacks. Only a few papers identify the need for automation and still less papers address this issue. Looking at the identified literature, automation is needed for example in quality evaluation (Immonen et al., 2015; Rathika and Arcokiam, 2014), policy adaption (Immonen et al., 2015) and data quality rule generation (Taleb and Serhani, 2017).

Taleb and Serhani (2017) address the need for automation and propose an automated data quality rules discovery model. This paper is related to Taleb et al. (2017) presented in Data Fitness. Data quality rule discovery should be placed between quality evaluation and pre-
The aim of the model is to define well-established quality rules that should afterwards be used in pre-processing. Pre-processing should be personalized for every dataset and based on quality requirements and quality evaluations. This can be achieved by including data attributes and target data quality dimensions in quality specifications. Furthermore, evaluation results should be used to discover, build, test, validate and finally optimize quality specifications. The model can be seen in figure 17. Experiments were conducted that show increased data quality scores after the application of the model.

Figure 17: Data Quality Rule Discovery Model (Taleb and Serhani, 2017)

Rathika and Arcokiam (2014) propose an automated model for data quality assessment in Big Data migrations, i.e. if vast amounts of data are extracted, transformed and loaded from legacy datastores into newer structures. The proposed process considers the quality dimensions completeness, validity, conformity, consistency, integrity and accuracy. It examines source as well as destination stores. After the migration process is completed, it focusses solely on the target store. The process consists of (i) data assessment, (ii) mapping document, (iii) data extraction, validation and loading, (iv) migration validation, and (v) post migration. Benefits of the process are that it saves money, time and ensures quality by automating the process.

Automation of different data quality evaluation processes could lead to higher overall quality as well as to cost and time savings. Such automations could be especially valuable when considering Big Data Volume, Velocity and Veracity where manual processes cannot cope with these characteristics. Automating processes will become more and more important and hence, should be considered with a focus on potential future needs. It is noteworthy that beside
automation in data quality evaluation processes, automated application testing is stated as one out of three main issues for Big Data application quality assurance in Tao and Gao (2016). This might be crucial for maintaining the quality of the Big Data application.

### 4.5.3. Interoperability

This taxonomy describes *Interoperability* as the interaction of Big Data application components across the Big Data value chain in order to provide consistent flows of high-quality data. Interoperability means defining and developing a well-established Big Data value chain considering the right components (like collection, ingestion, storage, analytics and visualization) and technologies that best fit quality requirements. There exist both, traditional standards on Enterprise Architecture (EA) as well as Big Data Reference Architectures (BDRAs). While traditional EA standards are generally more about strategic processes in building an architecture, BDRAs describe different Big Data value chain phases and resulting technologies which could be applied.

Regarding traditional EA standards, TOGAF (The Open Group, 2011) should be mentioned. The heart of TOGAF is their Architecture Development Method (ADM) that describes how to develop and manage an EA lifecycle. The framework includes four domains, namely (i) Business Architecture, (ii) Data Architecture, (iii) Application Architecture and (iv) Technology Architecture. The architecture consists of eight components representing a roadmap how to develop an EA. In particular, the phases describe the definition, planning, implementation and governance of the baseline architecture as well as the development of a migration plan for target architectures (Cameron and McMillan, 2013).

According to TOGAF (The Open Group, 2011), interoperability is “*the ability of two or more systems or components to exchange information*” (p. 27). Technical interoperability according to TOGAF can be seen as common methods “*for the communication, storage, processing, and access to data primarily in the application platform and communications infrastructure domains*” (p. 298) and is commonly based on the IT infrastructure. Architecture is key in defining standards to achieve interoperability. The goal of interoperability is an architecture that promotes a communicating infrastructure based on systems transparency including database management, user interface, data interchange and others.

To transfer the explanations above into our context, interoperability is highly related to a set of well-defined, high-interacting components that can be compromised in the Big Data value
chain. To do so, BDRAs might be essential to define value chain phases, technologies and resulting connections.

For example, NIST (2015b) presents a vendor-neutral, infrastructure- and technology-independent BDRA which consists of five roles, namely (i) data provider, (ii) system orchestrator, (iii) Big Data application provider, (iv) Big Data framework provider and (v) data consumer. The components are connected with arrows that present data flows, service usage or software tools. Those arrows define the interoperability interfaces. Similar can be seen in Pääkönen and Pakkala (2015) and Orvit (2013) who present BDRAs along the Big Data value chain and resulting data flows. Therefore, the individual solutions used in each of the value chain phases must be able to exchange data and hence, provide interoperability from data sources to data usage.

Regarding BDRAs, the reviewed literature present both, domain-specific (Cloud Standard Customer Council, 2017; Soares, 2015; Kune et al., 2016; Ramaswamy et al., 2013) as well as domain-independent high-level architectures (NIST, 2015b; Orit, 2013; Pääkönen and Pakkala, 2015). Other papers do not provide BDRAs but focus on general architectural requirements in Big Data (e.g. Demchenko and Membrey, 2014), data quality-driven architectural concerns (e.g. Gudivada et al., 2016) or present valuable enhancements for existing reference architectures (e.g. Immonen et al., 2016; Moreno and Serrano, 2018).

In table 9, key-BDRAs are presented and classified in alignment with the Big Data value chain. Such a classification provides a good overview over different phases and resulting technologies, which can be applied. Especially in business literature, company-specific reference architectures like that from Soares (2015) or Oracle (2013) often promote their products. Sticking to technological solutions from only one provider could have advantages and disadvantages. One advantage might be that the interaction of the components will be better, which could imply higher interoperability of components. One disadvantage could be that there will be a high dependence on only one vendor.
<table>
<thead>
<tr>
<th>Reference</th>
<th>Data creation/source</th>
<th>Data integration (Extract-Transform-Load)</th>
<th>Data storage</th>
<th>Data analytics (Processing and analysis)</th>
<th>Data usage (visualization)</th>
<th>Supplementary components</th>
<th>Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pääkönen and Pakkala</td>
<td>Structured, Semi-structured, Unstructured, In situ, Streaming</td>
<td>Extraction, Loading and Pre-processing</td>
<td>Eleven different storage types based on the value chain</td>
<td>Various technologies based on Big Data use cases from Facebook, LinkedIn, Twitter, etc.</td>
<td>Various technologies based on Big Data use cases from Facebook, LinkedIn, Twitter, etc.</td>
<td>Various technologies based on Big Data use cases from Facebook, LinkedIn, Twitter, etc.</td>
<td>Job and model specification</td>
</tr>
<tr>
<td>Orit (2013)</td>
<td>Defined based on Big Data Volume (massive data), Variety (structured, semi- and unstructured data), and Velocity (streaming data)</td>
<td>Transformation (Collection and Aggregation)</td>
<td>Type of storage technology is dependent on the transformation (sequel vs. parallel)</td>
<td>NoSQL and SQL</td>
<td>Data mining (Descriptive and Predictive)</td>
<td>-</td>
<td>Usage based on different formats, granularity and security consideration</td>
</tr>
<tr>
<td>NIST (2015b)</td>
<td>Data provider (e.g. sensor, human, etc.) provides data like public record, images, videos, tapes, audio, sensor data, web-logs, etc.</td>
<td>Application provider handles interface to data provider based on Extract-Transform-Load</td>
<td>Storage solutions that considers capacity and transfer</td>
<td>Analytics based on batch or stream processing</td>
<td>-</td>
<td>Visualization that communicate data in meaningful knowledge</td>
<td>-</td>
</tr>
<tr>
<td>CSCC (2017)</td>
<td>Structured: enterprise data (e.g. master data, historical data, etc.); Unstructured: Machine and sensor, image and video, social, internet data sets, weather data, third party.</td>
<td>Batch ingestion (Extract-Transform-Load), Change data capture, Document interpretation and classification, Data quality analysis</td>
<td>Six data repositories, e.g. Data warehouses, sand boxes, etc.</td>
<td>Data science practices, Search and survey for data</td>
<td>-</td>
<td>Visualization and storyboarding, Reporting and content analytics, Decision management, Predictive analytics and modelling, Cognitive analytics, Insight as a Service</td>
<td>-</td>
</tr>
</tbody>
</table>
Choosing the right technologies and build them along the Big Data value chain with a focus on the needed functional flows is highly important for the interoperability of technological components. A high-level collection of different application components will serve high interaction and hence, could lead to more consistent data quality.

It is noteworthy that some papers on data quality address the need to consider data quality across all phases of the Big Data value chain, like Immonen et al. (2015) or Taleb and Dssouli (2015). This might require special allowance when defining application component interoperability.

### 4.5.4. Extensibility

*Extensibility* refers to the utilization of model-driven approaches for application design. Model-driven approaches can lead to ultimate flexibility, system quality and maintainability in the long run by making software component changes (i.e. extensions) easier and more cost-effective. Approaches like the Model Driven Architecture (MDA) from the Object Management Group (OMG) already received attention in Big Data community, e.g. Zhang et al. (2017).

**MDA** (Soley, 2000) was developed by the OMG in 2000. In general, the MDA is an architectural framework which is vendor-, middleware- and language-neutral and based on system
modelling. It aims to integrate OMG standards, like Unified Modeling Language (UML) or Common Object Request Broker Architecture (CORBA), into the system development process. In particular, the first step in MDA is building a platform-independent application model. This model is also called “core model” and is built in UML. In a second step, the platform-independent core model can then be transformed into platform-specific models. Such a transformation is done by using so-called “mappings”. In other words, the MDA has platform-independent core models as a baseline and platform-specific models on top. If the core model is platform-independent, it is a lot easier to migrate for example future needs of different middleware. Hence, MDA maintains ultimate flexibility in changing infrastructures and modelling will also increase system quality.

Within a Big Data context, Zhang et al. (2017) present general MDA as a method for assuring the quality of Big Data applications and conclude that MDA is an important technique for maintaining the Big Data environment. Thus, MDA seems to be highly important for maintaining Big Data applications in the long run. Extensibility should be the key-term. There are already a few valuable quality-aware, model-driven approaches in Big Data related scientific literature. Casale et al. (2015) for example state that the understanding of Big Data technologies like MapReduce or NoSQL is poor from a modelling point of view and present DICE as a quality-aware, model-driven engineering approach applicable in data-intensive cloud applications. DICE consists of three layers which are compromised into a so-called DICE profile and are mainly based on UML. It has commonalities to the MDA by using independent models, technology-specific models (which still stay independent in the context of deployment) and additional information about technologies used and deployment characteristics of the application. DICE covers architectural optimization, verification and simulation. Guerriero et al. (2016) base their research on the DICE model and its profile layers and present a novel model-driven Big Data architecture. Furthermore, they introduce meta-models for design support. Artac et al. (2016) combine model-based quality assurance and agility by presenting DICER, a model-driven continuous deployment framework based on Model Driven Engineering (MDE) and DevOps.

In these days, Big Data still fails to comprehensively address model-driven approaches. Therefore, it is hard to make conclusions about its value in Big Data environments. On the contrary, model-driven approaches are well-established and crucial in traditional environments. Additionally, some authors, like Zhang et al. (2017), already define model driven approaches as key-requirements for Big Data application quality assurance.
4.5.5. Agility/Flexibility

In this classification, Agility/Flexibility refers to the fast adaption to changes, which seems to be vital in the fast-paced environment of Big Data. Agility/Flexibility can be achieved with prior mentioned cloud computing and model-driven approaches. As already stated, cloud computing offers configurable and ubiquitous resources (Hashem et al., 2015) and hence ensures flexible adjustments in storage space, processing power, or similar. MDA from OMG maintains ultimate flexibility when existing components need changes or enhancements (Soley, 2000). Beside the already explained approaches, there are also other topics addressed in literature, namely agile approaches for application development and data analytics.

In general, agile approaches are required in Big Data. For example, Frankova et al. (2016) explore if and how agile methodologies can be applied in Big Data projects. Based on a survey, results show that the most important aspects in Big Data applications include functional software, the right tools and processes (processes are less important in this regard) and reaction to change. Additionally, Earley (2014), with the help of a Chief Information Officer from the insurance industry, delivers a practical view on agility in Big Data analytics. He stresses the need of agility/flexibility by describing an example of the inappropriateness of a holistic organizational data model. Due to inconsistencies regarding the meaning of data between different departments, he suggests to better avoid developing a holistic data model than to let employees try to do so, because they will quickly give up. Contrary to a holistic solution, he qualifies the need to allow evolution and change. In this respect, the papers from Chen et al. (2016) and Nancy et al. (2017) analyze agile approaches in Big Data architecture and analytics. Chen et al. (2016) present an evolved architecture-centric approach to the development of agile Big Data analytics, called “Architecture-Centric Agile Big Data Analytics” (AABA) (p. 5378). AABA considers system design of Big Data and architecture-supported DevOps for agile analytics, which is denoted as critical for the continuous delivery of value. The developed AABA-architecture is different from traditional agile analytics as it considers software architecture as a central role of agility. AABA was developed based on collaborative practice research with a selected company (called “SSV”) to define practical issues and validate results with the help of practitioners. In particular, AABA is based on Big Data Design (BDD) that focuses on the application of reference architectures for an attribute-driven Big Data system design. Design fragments should be afterwards implemented and tested based on DevOps
technologies. AABA should support the identification of trade-offs, the discovery of value, experimentation and DevOps processes for the continuous and fast delivery of value.

Nancy et al. (2017) elaborate the implications of agile processes in Big Data analytics, in particular the steps cleansing, transformation, as well as analytics. They present a process model for Big Data analytics, which is an enhancement to Cross Industry Standard Process for Data Mining (CRISP-DM). It aligns the process with software development, adopts a flexible architecture based on the reference architecture from NIST (2015b) and incorporates agile practices. This model is able to adapt architectural differences considering Big Data V’s.

What this all amounts to is that fast environmental changes in Big Data are ubiquitous and require some form of organizational reaction. Within Big Data, there is a need to deliver value fast. Maybe this must be done to the cost of data quality. If decisions must be made quickly, data quality evaluation, pre-processing, or similar might be sometimes neglected. Therefore, it is all the better to focus on agile approaches within Big Data application development in order to be able to deliver fast value while on the other side have the abilities to provide quality data. Once again, agility/flexibility can be also achieved with the prior mentioned cloud solutions and model-driven approaches.
4.6. Validity

Validity is largely based on incorporating experimentation at run-time for requirements evaluation. Experiments are valuable means to validate non-functional (as well as functional) requirements and further decide on their value. It addresses the importance to validate various quality decisions made.

![Diagram of Validity](image)

Figure 18: Validity

4.6.1. Requirements Validity

Within the identified literature, there is a broad consensus that experimentation is critical for data-driven companies (Fabijan et al., 2017, Deng et al., 2013, Gupta et al., 2018) like Facebook, Amazon, Google, or Microsoft (Deng et al., 2013). In general, experiments are used to evaluate and test new features (Dimitriev et al., 2016), e.g. improved security (Mattos et al., 2017a), by analyzing collected data from a subset of users that received novel experimental versions of a product (Schermann et al., 2018). Controlled experiments are used by data-driven organizations in order to optimize the system and to support decision-making as well as design processes (Mattos et al., 2017a). Experiments are especially needed in environments that apply agile development methods (Kohavi et al., 2013), continuous development methods (e.g. based on DevOps) (Schermann et al., 2018) or autonomous systems (Mattos et al., 2017b). This is in line with the statements made at Agility/Flexibility by Chen et al. (2016) who conclude that
their developed AABA model supports experimentation. However, experimentation can lead to various benefits. A holistic description of benefits that go beyond product improvement can be found in Fabijan and Dimitriev (2017). Reports on performance measures of experiments (e.g. mean duration of an experiment, average time between first code change and experiment deployment, etc.) can be found in Kevic et al. (2017) who analyze experiment performance data from 21,220 experiments collected at Bing between 2014-2017.

Many of the identified literature is based on experience reports on large-scale experiments, which can be found for example at Microsoft (Kohavi et al., 2013; Gupta et al., 2018) or Facebook (Tang et al., 2015). Those reports set the focus on the evaluation of experiments. What is lacking in literature, is a clear consideration on the application of experiments in order to perform run-time validation of data-intensive, intelligent applications. It could be highly valuable to consider continuous experimentation in Big Data application design in order to continuously validate quality design requirements, which, in turn, highly improves data and application quality. Some papers are identified that contribute to this gap by describing continuous experimentation and corresponding integration of an experimental environment.

Scherman et al. (2018) analyze the current state in continuous experimentation. 31 interviews and a survey that result in 187 responses were conducted. Outcomes are classified in three different perspectives, namely (i) software perspective, (ii) developer perspective and (iii) process perspective. For example, (i) highly requires the right architectures like micro-service-based architectures. (ii) need comprehensive analytics and monitoring activities in order to enable problem discovery at run-time. (iii) requires a structured approach for conducting experiments since they are mostly based on intuition rather than on guidelines.

Fabijan et al. (2017) present a high-level model called “Experimentation Evolution Model” that describes the evolution from ad-hoc data analysis of customers to large-scale, continuous experimentation. The model consists of three dimensions (i.e. technical, organizational and business). From a technical point of view, the model focuses on technical aspects of the experimentation platform, like complexity of the platform, experimentation pervasiveness within product teams as well as general development activities. The model is divided into the four steps (i) crawl, (ii) walk, (iii) run and (iv) fly and should describe the evolution from non-data driven to data-driven companies where continuous controlled experiments are the norm for every change of any product within the company.

Kaufman et al. (2017) present the democratization of experiments at Booking.com. At Booking.com, there are around 1000 concurrent experiments conducted daily. Hence,
experimentation is so well-established that they wrap every organizational change, from infrastructure changes to various bug fixes, in an experiment. The paper presents the infrastructure of Booking.com as well as additional key features that support experimental democratization. In particular, the key features are (i) central repository including information about both, success and failure, that enables knowledge sharing, (ii) generic and extensible platforms for conducting experiments with minimized ad-hoc effort, (iii) development of trusted experimentation infrastructures by constantly monitoring data quality, (iv) loose coupled experimentation infrastructure and business logic, and (v) establishing safeguards for ensuring experimental ownership in an end-to-end fashion.

Xu et al. (2015) present the evolution from experimental infrastructure to organizational culture within continuous deployment at LinkedIn. They present their experimentation platform called “XLNT” with corresponding properties like scalability, usability or flexibility as well as the implemented experimental design, continuous deployment, and automated off-line analysis. They also describe the support processes for building experimental cultures. For example, they use unified metrics for A/B test reports and business reports in order to make experimental and business reports comparable. This enables experimentation for business changes. Other support processes are simplification of multiple testing and experiment-tracking of the experiments having the most impact.

Both papers from Mattos et al. (2017a; 2017b) focus on architectural aspects of continuous experimentation. Mattos et al. (2017a) discuss the need for an architectural framework for automated experiment development. They describe functional requirements of the proposed system architecture as well as potential problems and corresponding solutions. Mattos et al. (2017b) demonstrate the architecture presented above in a case study. They focus on clear architectural aspects which can support the automated experimentation environment.

All in all, experimentation provides the ability to validate quality requirements at-runtime. For example, if there is a need to integrate a new security feature or to fix a bug, experiments should be conducted prior to its initiation in order to evaluate how these changes could affect overall quality. Therefore, decisions that would lead to a lower quality can be abandoned before they are integrated into existing environments. Continuous experimentation should be especially used in fast environments, for example real-time data analytics applications.
4.7. Discussion

Summarizing this chapter, quality requirements of Big Data applications seem to be largely similar compared to traditional environments. For instance, scientific literature on Big Data security emphasize the assurance of quality requirements like Confidentiality, Integrity or Availability. This is in line with existing (non-Big Data-related) security standards, like ISO 27xxx, that express security through the same attributes. On the contrary, traditional standards often miss to consider Big Data peculiarities (i.e. Big Data V-characteristics) that makes their full applicability sometimes questionable. This argument can be underpinned by a few statements made in Big Data-related academic literature. Cheng et al. (2017) for example state that due to Big Data the data characteristics have changed (e.g. diverse data sources or remote data storage) and hence, standards like DAMA DMBOK cannot deal with Big Data in a comprehensive way. Furthermore, Tekiner and Keane (2013) explain that TOGAF or the Zachman Framework do not have a data-focus, which makes them less applicable in Big Data environments. As a result, quality attributes between Big Data and traditional environments have similarities but might be treated differently according to Big Data V’s. Furthermore, there are quality attributes that were already existent in traditional settings but receive growing attention in Big Data. For example, Scalability becomes one of the most important factors in Big Data to deal with the vast amounts of data while it might be more neglected in traditional environments. In a nutshell, high-quality Big Data applications strongly rely on its V-characteristics, especially on Volume, Variety, Velocity, and Veracity. Those characteristics represent the differences between Big Data and traditional environments. Consequently, present research largely follows those characteristics throughout their solutions.

The presented taxonomy provides aspects to achieve Big Data applications that can be denoted “high-quality” nowadays and in the future. To ensure data quality in Big Data applications, it is thus important to (i) choose solutions that enable the development of a quality Big Data application and to (ii) choose solutions that maintain the quality Big Data application. (i) is affected by attributes like Contextual-Requirements Completeness, Confidentiality, or Provenance. (ii) is influenced for example by Extensibility, Agility/Flexibility, or Requirements Validity.

Depending on the application at-hand, some attributes might be more important than others. Hence, there is a need to quantify those quality attributes. The next chapter will describe the approaches used for quantification, which is needed for the final artifact evaluation.
5. Criteria Catalogue

Chapter 5 outlines the criteria catalogue based on a GQM-approach. More precisely, the structure of the criteria catalogue is described in the first part. Afterwards, the metrics and their corresponding calculations are specified in section two. Finally, the visualization used to display metric calculation results is presented in the last part.

5.1. Basic Structure

To make the identified quality attributes within the classification scheme measurable, the GQM-approach is applied. This approach is already defined in chapter 1. The goals, questions and metrics are integrated in a criteria catalogue which was done in Microsoft Excel. In particular, the Excel-sheet consists of a goal for each of the quality dimensions (Security, Usability, etc.), and questions and metrics for each quality attribute (e.g. Confidentiality or Integrity at Security). The questions are based on the goals and the information from the main part. As metrics, I solely chose rating-scales. The reason is that rating-scales are highly useful for the visualization since they are rather generic and can be easier answered by my own. Mostly, metrics require specific information about a system or its performance (e.g. complexity of a system, number of privacy disclosures, etc.) that would be impossible to answer for an outside individual. However, all metrics consist of a name, a measuring object, a unit and scale, as well as a measuring method.

5.2. Calculation

The rating-scale has five levels and each level has a distinct value between 0 and 1, as shown below:

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>
Answering all questions according to this scheme, each question will have an answer with a value between 0 and 1. After the completion of the questionnaire, the next step is to calculate mean quality scores of the quality attributes.

The mean quality score of a quality attribute can be calculated as follows:

\[
X_i = \frac{\sum_{l=1}^{n} m_l}{n}
\]

where \( n \neq 0; n = \mathbb{N}; 0 \leq x_i \leq 1 \)

\( x_i \) = Attribute-values (like non-disclosure for Privacy) which must be a number between 0 and 1
\( i \) = The first question or metric in an attribute and should always start with 1
\( n \) = Number of questions or metrics in an attribute (without cost and time-related questions), e.g. if the attribute “Non-disclosure” at Privacy has three questions, then \( n = 3 \)
\( m_i \) = The value of the respective metric (0, 0.25, 0.50, 0.75, or 1)

After the calculation of the quality attributes, the mean quality scores of the quality dimensions should be calculated the following way:

\[
X_i = \frac{\sum_{l=1}^{n} x_i}{n} \times 100
\]

where \( n \neq 0; n = \mathbb{N}; 0 \leq X_i \leq 100 \)

\( X_i \) = Dimension-values (like Privacy or Persistence) which must be a number between 0 and 100
\( i \) = The first attribute in a dimension and should always start with 1
\( n \) = Number of attributes in a dimension, e.g. if the dimension “Privacy” has five attributes, then \( n = 5 \)
After applying formula (2), a required quality score with values between 0 and 100 should be the result for each quality dimension.

### 5.3. Visualization Tool

The quality scores are further visualized to make potential requirement differences visible. To do so, a radar-chart was built. The chart was developed in “PyCharm” and is written in Python. The code-base stems from the radar-chart example provided by the “Python Graph Gallery” (accessible at: [https://python-graph-gallery.com/390-basic-radar-chart/](https://python-graph-gallery.com/390-basic-radar-chart/)). First, the values from the questionnaire are imported and structured in a data frame. Afterwards, the quality scores are calculated and restored in another data frame. Column-names correspond to the identified quality dimensions. Then, the quality scores are extracted, and the angles are calculated. Both are stored in a separate list. Finally, the radar chart is initiated, and values are added. A code-snipped of the initiation of the radar-chart and the added values, names and angles can be seen in the following. The whole code can be found in appendix B.

```python
# Radar chart without data
radar_chart = plt.subplot(111, polar=True)

# Assigning variables (attributes) and angles
plt.xticks(angles[:-1], attributes, color="grey", size=12, fontweight="bold")
radar_chart.set_rlabel_position(-45)
plt.yticks([20, 40, 60, 80], ['20°', '40°', '60°', '80°'], color="grey", size=8, fontweight="bold")
plt.ylim(0, 100)

# Plot data
radar_chart.plot(angles, values, "b", linewidth=1, linestyle="dashed")
radar_chart.fill(angles, values, "b", alpha=0.1)
plt.show()
```
The developed radar-chart with example values of 50 for each quality dimension is presented in figure 19.

![Radar-Chart Example](image)

The colored area represents the plotted quality scores of the quality dimensions. “S” is Security, “U” is Usability, “V” is Validity, “Pe” is Persistence, and “Pr” is Privacy. Plotting quality scores by answering the questions based on different Big Data applications might reveal different quality profiles. To prove this assumption, the next chapter evaluates the criteria catalogue by answering the questions for two distinct Big Data applications.
6. Evaluation

This part evaluates the developed artifact. Based on the two case examples explained in chapter 1.2.4, the questionnaire is answered and resulting quality requirements are visualized. Answering the questions requires some information about the companies and their Big Data applications. This is done by using web articles as well as the book from Marr (2016a). I have to note that some quality attributes identified could not be answered fully based on available information. This can be seen as major validity threat. Testing the artifact under real conditions would be an interesting step in the future. Additionally, Big Data applications from global players like Amazon or Rolls Royce might have high requirements in most of the proposed attributes due to their firm size. Hence, the requirements are sometimes generalized by referring also to smaller applications. However, in the following, each question in every attribute is successively answered for both Big Data applications and afterwards, the results are visualized. The answered questions can be found in appendix A.

6.1. Usability

G 1.1: Contextual-requirements Completeness

Predictive Maintenance

At Rolls Royce, sensors track everything from pressure, temperature to altitude and many more (Choudhury and Mortleman, 2018). As stated in the main part, quality attributes can even differ between two sensors (Ramaswamy et al., 2013). Application design might become less useful if specific quality requirements of sensor data are not met. For example, sensor data quality could imply higher granularity, which, in turn, would require higher efforts in quality evaluation (e.g. increased accuracy or completeness checks). If design components do not satisfy such requirements, usability is questioned.

Recommendation Systems

Recommendation Systems also rely on specific quality requirements and must define either additional attributes or focus more on specific ones. For example, credibility (or trustworthiness) might be a quality attribute that should be highly addressed in application design, because data is mostly collected from non-proprietary sources.
G 1.2: Metadata Provision

Predictive Maintenance

The most emphasis at Rolls Royce’s Predictive Maintenance application is on internal data that originate from aircraft engines equipped with various sensors (Marr, 2016a). Hence, a large amount of data that is collected can be denoted as proprietary data. As stated in the main part, metadata should provide additional information about data quality levels, provenance and schemata. Metadata about data quality might be required to support credible analysis results. In general, quality efforts could be directly influenced at their proprietary sensors. But trustworthy results might only be obtained if additional information about the current quality levels of data are provided. Metadata about data provenance could be also required to validate which aircraft produces the data. Furthermore, if data is transformed one or more times, metadata about transformation history might be useful. Metadata of the data owner might not be required, because Rolls Royce is data owner and data handler at the same time. Metadata about data schemata is also not needed since IoT-sensors are proprietary and schemata could therefore be defined prior to its incorporation. Additionally, Rolls Royce also collects weather data as a supplementary factor of the analysis which are collected from third parties (RTInsights, 2016). In this case, metadata about data quality, provenance and schemata could be useful to support experts making conclusions about the current level of the data status. All in all, metadata about data quality and data provenance could support data credibility but might be not as important as if data is collected from non-proprietary sources. Metadata about data schemata seem to be not required.

Recommendation Systems

The Recommendation System at Amazon collects data of their customers in order to build “360-degree views” of each customer (Marr, 2016a). They collect what you buy, what you look for, your shipping address, etc. (Marr, 2016a). All those data represent external data where provenance, quality and schemata cannot be fully controlled. Thus, metadata is highly important for the collected data to provide some additional information about its quality, structure and provenance. This will support users to identify quality lacks, transformation changes, data owners and possible inappropriate structures in the data by directly providing the needed information to judge current data fitness. As a result, metadata about data quality, data provenance and data schemata are highly required.
G 1.3: Data Fitness

Predictive Maintenance

As already stated, Rolls Royce mainly collects data from diverse proprietary sensors. Those sensors can be directly influenced by Rolls Royce that could lead to similar data quality levels per sensor type. On the other side, vast amounts of signals could produce many data trash. As Nick Farrant, Rolls Royce’s senior vice president, states, “we and our customers are drowning in data, and many existing systems struggle to filter the signal from the noise and offer means to analyze things in a consistent way” (RTInsights, 2016). On that account, pre-processing of data might be required (e.g. activities like data cleansing, or similar). This should help to make data analysis more accurate and reduce storage costs by storing only essential data. Within the real-time on-board analytical efforts, pre-processing could be challenging and maybe leads to outdated results. Rather more, pre-processing is required for the in-depth analysis after a flight. Compared to data collected from various external, non-proprietary sources and consisting of heterogeneous quality levels, pre-processing might not be as important as in such environments but still needed.

Recommendation Systems

Amazon collects external data from numerous customers. The collected data could imply heterogeneous quality levels (e.g. consistency or redundancy) that must be unified through pre-processing based on data quality rules. Consequently, Data Fitness might be critical if data are collected from non-proprietary sources since data with highly diverse quality levels will be useless to analyze and will lead to low accurate results.

G 1.4: User Acceptance

Predictive Maintenance

Achieving high User Acceptance implies that the utilization of the application will increase. This could also affect data quality if corresponding software components are rated high. Nevertheless, since Big Data analytics efforts seem to be accomplished by real expert engineers, like at Rolls Royce’s operation centers (Marr, 2016a), User Acceptance might be a little less (but still) required in Predictive Maintenance. The reason is that experts will have a higher education in this field, which could lead to a general higher acceptance.

Recommendation Systems
User Acceptance seems to be highly demanded within Recommendation Systems. No information was found who analyzes data at Amazon, but it might be possible that also less-educated users (e.g. business users) will be involved in making small analytics and recommendation-based decisions in such an application. Therefore, User Acceptance might be achieved for a larger and probably less-educated crowd. Since the intention of Big Data is often nebulous and complex, application design must convey its usefulness to the users.

6.2. Security

Before I continue with security evaluations, we should remember the explanations regarding security and privacy relationships in the main part. The perception was that Confidentiality and Integrity are prerequisites for Privacy and equal to Non-disclosure in their models and techniques. The difference is their data focus, which means, if collected data represents personal data or non-personal data. In other words, Confidentiality and Integrity at Security are high if we look at non-personal data and Non-disclosure is high if we look at PII. This must be done, because those attributes are closely related but should still be seen distinct. Otherwise, there would not be any meaningful differences between these attributes, regardless on what Big Data application is looked at.

G 2.1: Confidentiality

Predictive Maintenance

Proprietary sensor data collected by Rolls Royce does not include any private information about specific individuals, like PII. Hence, efforts must be addressed in this dimension. In general, sensor data at Rolls Royce are highly critical within their businesses embodied by new services like “Total Care” (Marr, 2016a). Data theft could have severe consequences for those services as well as for the company’s reputation. On that score, it is highly required that data is protected from non-authorized entities. Ensuring Confidentiality in Big Data applications is especially needed in distributed environments. For example, if data is stored in a cloud, like at Rolls Royce (Marr, 2016a), it is not enough to ensure company-internal Confidentiality. Rather more, protection from non-authorized entities must be also achieved and maintained in the cloud, at the sensors, or through data transmissions. Protecting data from non-authorized access might also require classifying data to give special consideration to sensitive data. Confidentiality can therefore be seen as high requirement.
Recommendation Systems

Looking at Recommendation Systems, Amazon largely collects PII. Thus, data have high privacy requirements and must be consequently addressed at Non-disclosure.

G 2.2: Integrity

Predictive Maintenance

*Integrity* might be ensured at the sensors itself, during the transmission of data between sensors and operation centers, and within and across the operation centers. Hence, integrity seems to be highly required for the proprietary data across distributed environments in order to prevent data theft. To maintain integrity, data handling processes must be monitored and reviewed in appropriate intervals.

Recommendation Systems

Looking at the Recommendation System at Amazon, same conclusions can be drawn than at Confidentiality. Integrity is also a pre-requisite for Non-disclosure and therefore, should be ensured at the Privacy dimension.

G 2.3: Availability and System Immunity

Predictive Maintenance

Rolls Royce collects streaming data and analyzes them on-board as well as instantly after a flight at the operation centers (Marr, 2016a). Consequently, *Availability* might become a prime factor. Data must be available straightaway, otherwise anomalies within the datasets might be recognized too late. It is thus important to build other security requirements in a way that data is still fast available. As stated in Marr (2016a), Rolls Royce uses a secure private cloud which allows fast processing as well as fast retrieval. Regarding *System Immunity*, no information was found for Rolls Royce and Predictive Maintenance. However, *System Immunity* is highly precious if the application can be characterized through Volume and Velocity of data, which is the case at Rolls Royce. For example, hacking sensors and block data availability could lead to a general downtime of the application, which is fatal for such an application.

Recommendation Systems

At Amazon, data are often analyzed offline, i.e. products, which are likely to be bought together, are clustered offline due to time constraints and afterwards uploaded as input for online recommendations (Grajales, 2017). Consequently, product data clusters must have high *Availability* to compare these clusters with the user data in order to make instant item
suggestions. Considering *System Immunity*, also no information was found for Amazon and Recommendation Systems. As prior stated, *System Immunity* is crucial for Big Data Volume and Velocity. It is noteworthy that Amazon as a global player undoubtedly requires high *System Immunity*. But if we would look at a smaller version of this application, it might not be as required as for example in continuous real-time or autonomous environments, because Volume and Velocity might be lower.

**G 2.4: **Accountability

**Predictive Maintenance**

*Accountability* is needed for data that is accessible and used by many people. This requires roles and responsibilities which should be embedded in the application. *Accountability* is important to define access control and authorization of data or to conduct granular audits. At Rolls Royce, data are analyzed by real experts at the operation centers (Marr, 2016a). Such expert teams could represent a smaller fraction of people which are better educated and hence, might require a little less effort in *Accountability*. Considering *Accountability* through non-repudiation might be essential to foster overall trust.

**Recommendation Systems**

No information was found about who analyzes data but collected data at Amazon are highly sensitive and probably used by a lot of employees. Thus, accountabilities embedded in the Big Data application might be strongly required within Recommendation Systems. *Accountability* through non-repudiation should be provided by ensuring *Compliance* to privacy laws and regulation, which must be addressed at the *Privacy* dimension.

**6.3. Privacy**

**G 3.1: **Non-disclosure

**Predictive Maintenance**

As stated in Marr (2016a) “At Rolls-Royce, the emphasis is most definitely on internal data, particularly sensors fitted to the company’s products.” (p. 28). Therefore, Rolls Royce does not collect PII for their Predictive Maintenance application and hence, *Non-disclosure* of PII is not demanded.

**Recommendation Systems**
At Amazon, *Non-disclosure* is one of the prime factors. As stated by Marr (2016a), “*privacy [...] is an absolute priority*” (p. 291). One data breach can lead to destroy overall customer confidence (Marr, 2016a). Therefore, *Non-disclosure*, which is embodied by *Confidentiality* and *Integrity* of PII, is highly required. Amazon uses SSL-connections and an encrypted database for ensuring *Confidentiality* and *Integrity* of sensitive information (Marr, 2016a).

**G 3.2: Pre-serving**

**Predictive Maintenance**

Similar to *Non-disclosure*, there is no need to pre-serve the privacy of PII, because no personal data is collected.

**Recommendation Systems**

At Amazon, data *Pre-serving* must be ensured if PII are collected for analysis purposes. This should be done to use the data without knowing something about who owns them. At Amazon for example, data is also sold to advertisers. They sell anonymized access to the customer data (Marr, 2016a). Privacy *Pre-serving* is therefore highly demanded.

**G 3.3: Provenance**

**Predictive Maintenance**

Regarding Predictive Maintenance at Rolls Royce, data provenance could be required to validate individual aircrafts as data source and to track potential data transformations. However, from a privacy-perspective, it might be less needed. Since it is defined as generally required but probably not in this dimension, this attribute is denoted as neutral.

**Recommendation Systems**

*Provenance* of data is highly important in Recommendation Systems where data are largely collected from different sources containing PII. It serves information about who owns the data, leading to general trustworthiness.

**G 3.4: Conformance**

**Predictive Maintenance**

Rolls Royce collects proprietary data, which means, the enterprise is data owner and data handler simultaneously. On these grounds, there is no need for ensuring *Conformance* to data owners.
Recommendation Systems

Assuring Conformance seems to be required. As Marr (2016a) state, “[...] by far the biggest challenge for Amazon and all e-tailers was getting the public to put their faith in taking part in online commercial activity” (p. 291). Implied rules between data owners and data handler will raise overall data owner trust. Thus, ensuring Conformance is critical for Recommendation Systems.

G 3.5: Compliance

Predictive Maintenance

Privacy compliance should be neglected in Predictive Maintenance. New laws and regulations, like GDPR, focus on the protection of personal data, which is not gathered at Rolls Royce. Anyway, due to the importance of such new laws and regulations, totally ignore them might be the wrong way.

Recommendation Systems

Compliance within Recommendation Systems like that from Amazon must strongly base their efforts to privacy laws and regulations, like GDPR. There is also a need to focus on differences in country-specific laws and regulations. If privacy compliance is not achieved, expensive penalties and vanishing trust could be the result.

6.4. Persistence

G 4.1: Scalability

Predictive Maintenance

Scalable storage capabilities and processing power is highly required. Marr (2016a) for example states that “Data volumes are increasing fast, both with the growth in fleet and the increasing introduction of more data-equipped aircraft” (p. 28). Also, RTInsights (2016) emphasize the “rapidly increasing volume of data coming from different types of aircraft equipment”. Rolls Royce has around 13,000 engines (RTInsights, 2016) that produce 1,000 times more data through sensors than back in the 1990s (Marr, 2016a). In general, an Airbus A350 aircraft is equipped with around 6,000 sensors and produces 2.5Tb of data each day (Rolls-Royce, 2019). Especially Marr (2016a) explains that the high data growth “creates a demand for low-cost, scalable storage as well as rapid processing and retrieval” (p. 28 f.).
**Recommendation Systems**

Amazon and its Recommendation System also require *Scalability* in data storage and processing. Amazon analyzes data from around 310 million customer accounts (Hufford, 2018). Especially global players like Amazon strongly need scalable storage and processing power due to their firm size. Although, *Scalability* is not highlighted that much in literature related to Recommendation Systems, it seems to be of general high importance in applications that deal with large amounts of data (i.e. Big Data applications).

**G 4.2: Automation**

**Predictive Maintenance**

At Rolls Royce, sensor data of flight engines must be analyzed in real-time. Furthermore, the data must have certain levels of quality that must be available in a timely manner, otherwise all efforts are wasted (Marr, 2016a). Manual data quality checks might take too long to provide results in appropriate time. Automated data quality evaluation could be highly required to prevent failures and provide quality results in a timely manner within their live performance evaluation. As Marr (2016a) explains, “The emphasis [...] is to detect as early as possible with a confident diagnosis and prognosis, while minimizing the rate of false-positives” (p. 29).

**Recommendation Systems**

Looking at the Recommendation System, *Automation* is also required to prevent manual flaws in quality evaluation activities. It will make quality checks more reliable and faster. Especially if data is collected from non-proprietary sources, automated quality checks could improve credibility of data-driven decisions while save money and time.

**G 4.3: Interoperability**

**Predictive Maintenance**

At Rolls Royce, data are created by sensors, transmitted via SATCOM and VHF radio during the flight and through 3G/Wi-Fi when the aircraft lands (Marr, 2016a). Real-time analysis implies that the data transmitted is instantly analyzed and visualized in order to make fast decisions. This requires high *Interoperability* of sensors, on-board analytical frameworks, analytical and visualization frameworks at the operation centers as well as appropriate storage. *Interoperability* is highly required to keep data quality consistent from creation to visualization.

**Recommendation System**
Looking at Amazon, data is created as user browse the website, collected, stored in a central data warehouse, and analyzed for building customer profiles (Marr, 2016a). This also requires high *Interoperability* of different components like interoperability between the data warehouse and the analytics framework.

**G 4.4: Extensibility**

**Predictive Maintenance**

As explained in the main part, using model-driven approaches will serve optimization for architectural application components. No information was found neither Rolls Royce uses modelling within their architecture nor how important it is for specific components. Nevertheless, using such approaches could lead to maintain application quality in the long-run. On the down side, it might be not an essential requirement and due to the novelty of model-driven approaches in Big Data, it might also lead to unsatisfied results, at least in these days. Therefore, *Extensibility* can be seen as neutral requirement, but might become highly required in the future.

**Recommendation Systems**

Regarding Amazon and its Recommendation System, also no information about the use of model-driven approaches and its importance was found. However, similar conclusions could be drawn than at Predictive Maintenance.

**G 4.5: Agility/Flexibility**

**Predictive Maintenance**

As stated in the main part, *Agility/Flexibility* can be achieved by cloud solutions, model-driven approaches, or agile analytics and/or architectures. Fast respondence to change seems to be especially required in real-time environments like that from Predictive Maintenance. This can be underpinned by examples of how Rolls Royce applies agile approaches. In general, Rolls Royce uses a cloud facility which enables the processing of throughput (i.e. real-time analysis) while on the other side maintains a so-called “data lake” for their offline analytics (Marr, 2016a). The cloud enables the combination of growing data sources that has the ability to identify and develop new service innovations for their customers. Additionally, Rolls Royce uses agile principles in their “R2 Data Labs”, a data-centric initiative, that enables the development of data applications for operational, manufacturing and design efficiencies (Choudhury and Mortleman, 2018). So, *Agility/Flexibility* of analysis is ensured by the cloud
facility and is also addressed in service efficiency improvement innovations. This might affect data and application quality by continuously providing appropriate services in this fast-paced environment.

**Recommendation Systems**

No information was found if agile approaches are used within their Big Data application. Once again, agile approaches might be required for fast decision making and performance efficiency. However, Agility/Flexibility could be a little less important as within applications where Velocity is the dominating factor.

### 6.5. Validity

**G 5.1: Requirements Validity**

**Predictive Maintenance**

*Requirements Validity* should be conducted whenever there is a feature change that might affect quality. For example, if a new encryption method is implemented that would highly reduce availability, prior validation will support the fast identification of that flaw and prevent the enterprise to integrate this new feature in a wrong way. Generally, requirements validation can be achieved using experimentation at run-time. At Rolls Royce, where data is (among others) analyzed autonomously (Choudhury and Mortleman, 2018) on-board during flights to transfer highlights to the operation centers (Marr, 2016a), run-time experimentation of potential new features might be strongly demanded. The reason is that within this real-time analysis environment, faulty features should be identified prior to its initiation. New needs should be automatically identified, and appropriate solutions must be instantly integrated. Additionally, run-time experimentation is highly suitable for agile environments as stated in the main part.

**Recommendation Systems**

*Requirements Validity* could be also valuable to validate feature changes of the application. But since Velocity is not the prime factor in Recommendation Systems, real-time validation needs might be slightly lower.
6.6. Result Visualization

Above, the evaluation results are displayed. The results show example quality profiles of individual Big Data applications that can be achieved by using the proposed taxonomy. Below, a short summary of both quality profiles is presented:

Predictive Maintenance mainly collects proprietary data coming from sensors. Those proprietary data do not contain any PII. Confidentiality and Integrity are therefore prime factors to protect data from un-authorized access and from modification or deletion. Additionally, fast-paced environments dare data availability and overall security and hence, require Availability and System Immunity. Furthermore, as Volume and Velocity dominates Predictive Maintenance, attributes like Scalability, Agility/Flexibility as well as Requirements Validity are highly needed in order to cope with the high data growth as well as to serve a fast respondence to change. Especially Requirements Validity fits well with Agility/Flexibility. Privacy can be neglected since no personal information is collected and hence, security requirements are expressed through Security. Usability is also slightly less important beside Contextual-Requirements Completeness. Since proprietary sensors can be directly influenced, Metadata Provision and Data Fitness are not highly (but still) demanded. Furthermore, User Acceptance can be easier achieved, because data is mainly used by experts, which might already have a certain level of acceptance due to their higher knowledge in this field.

Looking at Recommendation Systems, the main goal is to reveal customer preferences by collecting personal data, analyzing them and define customer profiles for each of the customer. Due to the collection of personal data, Privacy comes to the fore. The focus should be on Non-
disclosure of PII embodied by confidentiality and integrity requirements. During analytic activities, privacy needs to be continuously Pre-served to prevent data identification. Furthermore, Provenance should increase data trust and ownership traceability. Finally, Conformance between data owners and data handler as well as privacy Compliance to external laws and regulations are highly demanded. Since Recommendation Systems rely more on Variety than on Velocity, attributes like Agility/Flexibility and Requirements Validity are a little less (but still) important. Instead, this application highly demands Usability-attributes like Metadata Provision or Data Fitness. High Variety in data could imply diverse quality levels. These quality variances must be unified through data pre-processing according to defined quality rules. Additionally, because data quality levels, data provenance and data structures cannot be directly influenced (since the collected data is non-proprietary), metadata must be delivered to support quality judgements.

All in all, the evaluation based on the taxonomy proves its usefulness by illustrating contextual quality profiles. Within the taxonomy, I tried to present a comprehensive set of quality requirements for Big Data applications where the importance of specific quality attributes can vary according to the specific Big Data application at-hand.

6.7. Costs and Time

What is still missing within the evaluation above is the consideration of costs and time. Resources will never be infinite in real-world applications and hence, require a certain trade-off. This trade-off between achieving certain levels of quality and available resources must be balanced to get an appropriate level of application quality dependent on available budget and time. Due to this importance, the questionnaire also includes questions about costs and time needed to ensure each quality attribute. Those questions can be answered by real enterprises but not on my own. Since I haven’t any insights about needed costs and time for integrating specific quality levels at individual enterprises, including costs and time in the evaluation above would go outside the scope of this work. However, cost- and time-related questions are included in the questionnaire to view how much costs and time are needed to achieve the desired level of application quality, which can afterwards be compared to available resources. If needed costs and time exceed available resources, one must find a meaningful trade-off. For example, if needed costs and time to ensure overall quality of a Predictive Maintenance application are too high, it might be required to analyze which quality attributes could be reduced to meet resource
constraints while on the other side still deliver an acceptable level of quality. Furthermore, this taxonomy includes quality attributes which could be highly valuable for quality assurance but on the other side might be not essential in conducting a specific Big Data application. For example, Requirements Validity seems to be highly important in Predictive Maintenance (and therefore is also denoted as highly required in this evaluation), but depending on the Big Data application size, it might not be a key requirement like scalability, confidentiality, integrity, availability, or similar. This should be also considered if monetary or time-related decisions are made.
7. Limitations

This design-science research includes some limitations which should be acknowledged. The biggest limitation of this work might be the evaluation part. Needed information is mainly gathered from various web articles and the book from Marr (2016a). All these materials play a part in contributing to the evaluation. On the downside, several questions in the developed questionnaire could hardly be answered due to missing information. Example quality attributes are Extensibility or Requirements Validity. If rare information was provided, the questions are largely answered based on own related conclusions drawn from the main part and from the material used in the evaluation. For example, Requirements Validity can be achieved using experimentation at run-time to continuously validate quality characteristics. No information was found if and how strong real-time quality validation is demanded in both, Predictive Maintenance and Recommendation Systems. As described in Mattos et al. (2017b), continuous experimentation is especially valuable for autonomous systems, which are also used by Rolls Royce within their aircraft engines for on-board data analysis. Based on such insights, I could ponder the importance of Requirements validity for the application at-hand. In my opinion, the answers sound reasonable, but indeed, they are not proven.

Another limitation which inevitably also stems from the fact that the evaluation has to be done by myself is the selection of metrics in the questionnaire. In general, many valuable metrics can be found in literature, standards or over the internet. Mostly, those metrics refer to operational information within an enterprise. For example, reasonably answering common questions like “number of privacy breaches”, “complexity of a system” or “percentage of data that is denoted as sensitive”, is impossible to do by my own. Building questions like those above enable a wide range of metrics to be applied. Metrics like “McCabe metric for system complexity” or “percentage of sensitive data” could serve valuable answers to measure. Since this is not possible for me, the only metrics that could remain include “Yes/No” or “Rating-scales”. Yes/No-questions might be inappropriate because values can either be “1” in case of “Yes” or “0” in case of “No”. This ignores values between 0 and 1 that makes it especially hard if individual Big Data quality profiles should be developed. As a result, the questions are solely based on rating-scales, which were found as most appropriate to answer the questions by myself but also reveal potential differences. Additionally, costs and time are included but not answered in the evaluation, also due to missing information. The reason for their inclusion was that if this
questionnaire is used in real-life, costs and time can be specified and in case that available resources exceed required, trade-offs could be defined based on the quality scores prior defined.
8. Conclusion and Future Research

This master thesis investigated data quality-specific criteria of Big Data applications. In particular, it proposes a non-functional requirements specification framework, the so-called "Quality-aware Big Data Application Taxonomy", that should support future Big Data initiations by tailoring Big Data quality according to application-individual quality requirements. Based on design-science research, the developed artifact has its roots on an in-depth review of standards, scientific as well as business literature. It is embodied by a multi-dimensional, hierarchical classification of quality dimensions and corresponding attributes. Additionally, to make those quality criteria tangible, a GQM-approach was used for the purpose of quantification. The GQM-approach is embedded in a criteria catalogue in the form of a questionnaire, including questions and resulting measuring metrics. Metrics are finally visualized by developing a radar-chart. The taxonomy was evaluated with the help of the questionnaire to prove its usefulness by plotting Big Data quality profiles of two distinct applications. It is based on the case examples “Predictive Maintenance at Rolls Royce” and “Recommendation Systems at Amazon”.

Reviewing this master thesis, I can conclude that by far the biggest challenges for quality-aware Big Data applications originate from Volume, Variety, Velocity, and Veracity. Though, Value is also sometimes used to characterize Big Data, I personally think that Value is the outcome if quality requirements from Volume, Variety, Velocity, and Veracity are met throughout the Big Data application. In other words, if quality attributes consider the assurance of Volume, Variety, Velocity, and Veracity, data to be collected for the Big Data application at-hand will have Value. Moreover, most of the identified literature base their proposed solutions on challenges that grow out from those four V’s. If an enterprise will start a Big Data project, the most important aspect might be to continuously base its efforts on Big Data V’s and analyze how to satisfy them in order to achieve Big Data Value. Consequently, Big Data should be characterized by Volume, Variety, Velocity, and Veracity. This perception serves a valuable contribution for the standardization of Big Data characterization.

The research question can be answered by looking at the developed artifact. Based on this artifact, I can further conclude that Big Data and its V-characteristics often have similar quality requirements compared to traditional environments, but those requirements must be satisfied differently. Like a traditional IS requires information security through confidentiality, integrity, or availability, so does a Big Data application. But the differences are embodied by the ways to
ensure them. Another perception that can be drawn from this taxonomy is that some quality requirements, which were already existent before the rise of Big Data, experience increased importance, like scalability or automation. Basically, the taxonomy is classified in the two high-level dimensions QoC and QoD. Indeed, focus relied on QoD. QoC describes inherent quality dimensions of Big Data, including Completeness, Correctness, Consistency, Currentness and Credibility. QoD refers to the surrounding quality design dimensions of Big Data applications, namely Usability, Security, Privacy, Persistence and Validity. The quality design dimensions are further divided into quality attributes. Usability consists of Contextual-Requirements Completeness, Metadata-Provision, Data Fitness and User Acceptance. Security includes Confidentiality, Integrity, Availability and System Immunity and Accountability. Privacy can be explained through Non-disclosure, Pre-serving, Provenance, Conformance and Compliance. Persistence is embodied by Scalability, Automation, Interoperability, Extensibility and Agility/Flexibility. Finally, Validity includes Requirements Validity. Such a classification serves a valuable contribution to existing research, which, to the best of my knowledge, has not been provided so far.

To prove its value, the taxonomy is evaluated with the help of the questionnaire and the corresponding visualization tool. Results show that the classification serves contextual design quality profiles depending on the Big Data application at-hand. From my personal perspective, the proposed classification and questionnaire deliver a valuable contribution to both, research and industry. First, it supports a general understanding of data quality in Big Data applications from the viewpoint of research that could foster other academic efforts in the future. Second, the taxonomy is meaningful for future definitions of quality-aware Big Data applications in industry by reviewing the classification, elaborating the questionnaire and making decisions about the right initiation.

Key-insights from the main part are, (i) that Usability of the Big Data application is highly challenged by its contextual-nature coupled with the inability to influence data quality at the creation phase. This requires defining quality needs for the task at-hand and pre-process data according to those quality needs. Additionally, users must be focused by considering a user-centric application design and support general understandability by providing metadata. This might be even more important as in traditional settings, because the value of Big Data is often nebulous leading to general user dissatisfaction. Another key-insight (ii) is described within the often-confusing relationship between Security and Privacy. In literature, security and privacy efforts are often mixed-up. Although their tight relationship is frequently cited, they
must be viewed as two distinct topics. Security, especially Confidentiality and Integrity, are prerequisites for privacy, but privacy has a different focus and therefore requires a distinct terminology. While security aims to protect company-internal information assets, privacy focuses on the protection of PII. This leads to the terminology of Non-disclosure, which refers to Confidentiality and Integrity at security. Furthermore, new aspects like data de-identification, provenance or privacy compliance must be considered in Big Data privacy. The third key-insight (iii) from the main part is that some quality attributes seem to be time-sensitive, which means, that the requirements in those attributes will grow over time and therefore must be ensured today and in the future. These attributes are embedded in the quality dimension Persistence. I think that especially due to the fast-paced environment of Big Data, addressing quality attributes that maintain a high-quality Big Data application now and prospectively, is decisive. A last key-insight (iv) stems from the quality dimension Validity. It teaches us that quality requirements are not stable and thus, must be controlled and tested to apply only those changes that really lead to desired outcomes.

Finally, some research challenges emerge from the main part. First, there is no widely accepted standard on Big Data quality, neither a data-specific nor an application-specific. Second, research addressing the applicability of traditional standards (e.g. ISO/IEC 8000-61, ISO/IEC 2700x, DAMA DMBOK, TOGAF, etc.) is hardly lacking. There are only some papers giving some general hints about their applicability. Third, Big Data security and privacy are often mixed-up in academic literature. There should be a clearer distinction in focus and terminology. Fourth, literature on run-time experimentation misses the consideration of Big Data requirements validation.

The aim of this work was to deliver a comprehensive classification of Big Data application quality characteristics and to validate its usefulness. The limitations presented in the last chapter create some possible incentives for future research for both, other researchers and myself. In particular, real-world evaluations of the proposed artifact could be highly valuable to prove its applicability. Additionally, further classifications from other researchers could lead to better understand the completeness and value of this taxonomy. Increasing reviewed literature, including other standards, applying different search processes for study identification or focusing more on real-world applications in the definition of such a classification can also be future directions to new research.
Appendix A: Completed Questionnaire for the Evaluation (Chapter 6)

In the following, the answers to the questions from the artifact evaluation in chapter 6 are illustrated.

**Dimension 1: Usability**

**Attribute 1.1: Contextual Requirements Completeness**

*Predictive Maintenance:*

Q 1.1.1: According to the data that must be collected for the Big Data application at-hand (e.g. sensor data, personal data, etc.), do the data items warrant context-specific quality requirements which could affect application development?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
</tbody>
</table>

*Recommendation Systems:*

Q 1.1.1: According to the data that must be collected for the Big Data application at-hand (e.g. sensor data, personal data, etc.), do the data items warrant context-specific quality requirements which could affect application development?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
</tbody>
</table>

**Attribute 1.2: Metadata Provision**

*Predictive Maintenance:*

Q 1.2.1: Are data collected from external (proprietary or non-proprietary) sources and require provision of additional information about data quality levels to the user?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
</tr>
</tbody>
</table>

Q 1.2.2: Are data collected from external (proprietary or non-proprietary) sources and require provision of additional information about the data owner/data source to the user?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highly not required</td>
<td>Not required</td>
<td>Neutral</td>
<td>Required</td>
<td>Highly required</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------</td>
<td>---------</td>
<td>----------</td>
<td>----------------</td>
</tr>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
</tr>
</tbody>
</table>

**Q 1.2.3:** Are data collected from external (proprietary or non-proprietary) sources and require provision of additional information about the data itself (i.e. technical details or schemata) to the user?

**Recommendation Systems:**

**Q 1.2.1:** Are data collected from external (proprietary or non-proprietary) sources and require provision of additional information about data quality levels to the user?

**Q 1.2.2:** Are data collected from external (proprietary or non-proprietary) sources and require provision of additional information about the data owner/data source to the user?

**Q 1.2.3:** Are data collected from external (proprietary or non-proprietary) sources and require provision of additional information about the data itself (i.e. technical details or schemata) to the user?

**Attribute 1.3:** Data Fitness

**Predictive Maintenance:**

**Q 1.3.1:** Regarding the diversity of data quality levels (within the collected data), is pre-processing of data needed in order to unify data quality levels based on the pre-defined set of contextual quality-requirements related to G.1.1?
**Recommendation Systems:**

**Q 1.3.1:** Regarding the diversity of data quality levels (within the collected data), is pre-processing of data needed in order to unify data quality levels based on the pre-defined set of contextual quality-requirements related to G.1.1?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
</tbody>
</table>

**Attribute 1.4:** User Acceptance

**Predictive Maintenance:**

**Q 1.4.1:** How strong do the software components of the application rely on a user-centric design to maximize user acceptance?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
</tr>
</tbody>
</table>

**Recommendation Systems:**

**Q 1.4.1:** How strong do the software components of the application rely on a user-centric design to maximize user acceptance?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
</tbody>
</table>

**Dimension 2:** Security

**Attribute 2.1:** Confidentiality

**Predictive Maintenance:**

**Q 2.1.1:** Must data be classified according to their sensitivity (e.g. if using distributed frameworks like cloud)?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
</tbody>
</table>

**Q 2.1.2:** Do proprietary data demand prevention of non-authorized access across distributed environments?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
</tbody>
</table>
**Q 2.1.3:** Should proprietary data requests be authenticated across distributed environments in order to ensure environmental trust (e.g. TLS/SSL connections)?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
</tbody>
</table>

**Recommendation Systems:**

**Q 2.1.1:** Must data be classified according to their sensitivity (e.g. if using distributed frameworks like cloud)?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
</tbody>
</table>

**Q 2.1.2:** Do proprietary data demand prevention of non-authorized access across distributed environments?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
</tbody>
</table>

**Q 2.1.3:** Should proprietary data requests be authenticated across distributed environments in order to ensure environmental trust (e.g. TLS/SSL connections)?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
</tbody>
</table>

**Attribute 2.2:** Integrity

**Predictive Maintenance:**

**Q 2.2.1:** Is integrity of proprietary data required, regardless where the data resides (e.g. through access control, communication integrity, or end-point input validation)?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
</tbody>
</table>

**Q 2.2.2:** Must data handling processes be controlled (e.g. for granular audits)?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
</tbody>
</table>
**Recommendation Systems:**

*Q 2.2.1: Is integrity of proprietary data required, regardless where the data resides (e.g. through access control, communication integrity, or end-point input validation)?*

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
</tbody>
</table>

*Q 2.2.2: Must data handling processes be controlled (e.g. for granular audits)?*

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
</tbody>
</table>

**Attribute 2.3: Availability and System Immunity**

**Predictive Maintenance:**

*Q 2.3.1: Do data items must be highly available, or easily or quickly retrievable (e.g. with High Available Systems)?*

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
</tbody>
</table>

*Q 2.3.2: According to data Volume and Velocity, how strong do data warrant real-time security monitoring through advanced analytics (i.e. Security Intelligence)?*

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
</tbody>
</table>

**Recommendation Systems:**

*Q 2.3.1: Do data items must be highly available, or easily or quickly retrievable (e.g. with High Available Systems)?*

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
</tbody>
</table>

*Q 2.3.2: According to data Volume and Velocity, how strong do data warrant real-time security monitoring through advanced analytics (i.e. Security Intelligence)?*
**Attribute 2.4: Accountability**

*Predictive Maintenance:*

**Q 2.4.1:** Do sensitive data items have high accountability requirements (integrating roles and responsibilities), e.g. to track inappropriate user behavior for granular audits, or to define roles for access control?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
</tr>
</tbody>
</table>

**Q 2.4.2:** Does the security effort require to be based on standards (e.g. ISO/IEC 2700x) in order to provide security non-repudiation and hence, overall trust?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
</tbody>
</table>

**Recommendation Systems:**

**Q 2.4.1:** Do sensitive data items have high accountability requirements (integrating roles and responsibilities), e.g. to track inappropriate user behavior for granular audits, or to define roles for access control?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
</tbody>
</table>

**Q 2.4.2:** Does the security effort require to be based on standards (e.g. ISO/IEC 2700x) in order to provide security non-repudiation and hence, overall trust?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>●</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

**Dimension 3: Privacy**

**Attribute 3.1: Non-disclosure**

*Predictive Maintenance:*
**Q 3.1.1:** Does the Big Data application largely rely on Personal Identifiable Information (PII) and therefore demand mechanisms to ensure non-disclosure (i.e. confidentiality and integrity) of PII?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>●</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

*Recommendation Systems:*

**Q 3.1.1:** Does the Big Data application largely rely on Personal Identifiable Information (PII) and therefore demand mechanisms to ensure non-disclosure (i.e. confidentiality and integrity) of PII?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>●</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
</tbody>
</table>

**Attribute 3.2:** Pre-serving

**Predictive Maintenance:**

**Q 3.2.1:** Should data items be de-identified during analytic activities in order to counter the identification of PII?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>●</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

*Recommendation Systems:*

**Q 3.2.1:** Should data items be de-identified during analytic activities in order to counter the identification of PII?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
</tbody>
</table>

**Attribute 3.3:** Provenance

**Predictive Maintenance:**

**Q 3.3.1:** Have data items been mainly collected from external (proprietary or non-proprietary) sources and therefore require data owner/data source validation (e.g. endpoint-input validation)?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Recommendation Systems:

Q 3.3.1: Have data items been mainly collected from external (proprietary or non-proprietary) sources and therefore require data owner/data source validation (e.g. endpoint-input validation)?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

Attribute 3.4: Conformance

Predictive Maintenance:

Q 3.4.1: Is data collected largely from external data owners which requires to seek consent (or similar expectations) of the owner regarding the data to be collected?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>●</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

Recommendation Systems:

Q 3.4.1: Is data collected largely from external data owners which requires to seek consent (or similar expectations) of the owner regarding the data to be collected?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
</tbody>
</table>

Attribute 3.5: Compliance

Predictive Maintenance:

Q 3.5.1: How strong do data within the application requires compliance to country-specific privacy laws and regulations (e.g. GDPR) which comprises a privacy-centric application development and control through granular audits or real-time security monitoring?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
</tbody>
</table>
Recommendation Systems:

Q 3.5.1: How strong do data within the application requires compliance to country-specific privacy laws and regulations (e.g. GDPR) which comprises a privacy-centric application development and control through granular audits or real-time security monitoring?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

Dimension 4: Persistence

Attribute 4.1: Scalability

Predictive Maintenance:

Q 4.1.1: According to Big Data Volume and Variety, how high are scalability requirements regarding data storage of heterogenous data over time?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
</tbody>
</table>

Q 4.1.2: According to Big Data Volume and Velocity, do data highly warrants scalability of processing power (e.g. for real-time data quality evaluation over time)?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
</tbody>
</table>

Recommendation Systems:

Q 4.1.1: According to Big Data Volume and Variety, how high are scalability requirements regarding data storage of heterogenous data over time?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
</tbody>
</table>
Q 4.1.2: According to Big Data Volume and Velocity, do data highly warrant scalability of processing power (e.g. for real-time data quality evaluation over time)?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
</tbody>
</table>

**Attribute 4.2: Automation**

**Predictive Maintenance:**

Q 4.2.1: According to Big Data Volume, Velocity and Veracity, do data require increasing automation of data quality evaluation processes in order to prevent manual failures?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
</tbody>
</table>

**Recommendation Systems:**

Q 4.2.1: According to Big Data Volume, Velocity and Veracity, do data require increasing automation of data quality evaluation processes in order to prevent manual failures?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
</tbody>
</table>

**Attribute 4.3: Interoperability**

**Predictive Maintenance:**

Q 4.3.1: Should the software components of the application be highly interoperable to ensure data portability throughout the whole Big Data value chain (from creation/collection to visualization)?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
</tbody>
</table>

**Recommendation Systems:**
Q 4.3.1: Should the software components of the application be highly interoperable to ensure data portability throughout the whole Big Data value chain (from creation/collection to visualization)?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
</tbody>
</table>

**Attribute 4.4: Extensibility**

**Predictive Maintenance:**

Q 4.4.1: Do the software components of the application demand Model-Driven Engineering (MDE-) approaches to deliver long-term stability and flexible extensibility of component changes?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

**Recommendation Systems:**

Q 4.4.1: Do the software components of the application demand Model-Driven Engineering (MDE-) approaches to deliver long-term stability and flexible extensibility of component changes?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>●</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

**Attribute 4.5: Agility/Flexibility**

**Predictive Maintenance:**

Q 4.5.1: Are agile methodologies required within different application-specific activities (e.g. data analytics)?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>●</td>
</tr>
</tbody>
</table>

**Recommendation Systems:**

Q 4.5.1: Are agile methodologies required within different application-specific activities (e.g. data analytics)?
### Dimension 5: Validity

**Attribute 5.1: Requirements Validity**

**Predictive Maintenance:**

Q 5.1.1: How strong do application feature changes require experimentation at run-time?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>•</td>
<td>○</td>
</tr>
</tbody>
</table>

**Recommendation Systems:**

Q 5.1.1: How strong do application feature changes require experimentation at run-time?

<table>
<thead>
<tr>
<th>Highly not required</th>
<th>Not required</th>
<th>Neutral</th>
<th>Required</th>
<th>Highly required</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0%)</td>
<td>(25%)</td>
<td>(50%)</td>
<td>(75%)</td>
<td>(100%)</td>
</tr>
<tr>
<td>○</td>
<td>○</td>
<td>○</td>
<td>•</td>
<td>○</td>
</tr>
</tbody>
</table>
Appendix B: Visualization Code

```python
import matplotlib.pyplot as plt
import pandas as pd
from math import pi
import os

class InvalidValueError(Exception):
    pass

file_path = os.path.join(os.path.dirname(__file__), "Questionnaire.xls")
sheet = pd.read_excel(file_path)

# Extracting metric values from the questionnaire and build a data frame
df = pd.DataFrame({
    "Groups":       ["Row1", "Row2", "Row3", "Row4", "Row5", "Row6", "Row7", "Row8", "Row9"],
    "Usability":    [sheet.loc[7][0], sheet.loc[15][0], sheet.loc[17][0], sheet.loc[19][0], sheet.loc[27][0], sheet.loc[35][0], None, None, None],
    "Security":     [sheet.loc[46][0], sheet.loc[48][0], sheet.loc[50][0], sheet.loc[58][0], sheet.loc[60][0], sheet.loc[68][0], sheet.loc[70][0], sheet.loc[78][0], sheet.loc[80][0]],
    "Privacy":      [sheet.loc[91][0], sheet.loc[99][0], sheet.loc[107][0], sheet.loc[115][0], sheet.loc[123][0], None, None, None, None],
    "Persistence":  [sheet.loc[134][0], sheet.loc[136][0], sheet.loc[144][0], sheet.loc[152][0], sheet.loc[160][0], sheet.loc[168][0], None, None, None],
    "Validity":     [sheet.loc[179][0], None, None, None, None, None, None, None, None]
})

# Calculating the quality scores
# Usability
usability = ((float(df.iloc[0, 1]) + float((sum(df.iloc[1:4, 1]) / 3)) + float(df.iloc[4, 1]) + float(df.iloc[5, 1])) / 4) * 100
# Security
security = ((float(sum(df.iloc[0:3, 2]) / 3) + float(sum(df.iloc[3:5, 2]) / 2) + float(sum(df.iloc[5:7, 2]) / 2) + float(sum(df.iloc[7:9, 2]) / 2)) / 4) * 100
# Privacy
privacy = ((df.iloc[0, 3] + df.iloc[1, 3] + df.iloc[2, 3]) / 3) * 100
# Persistence
persistence = (float(sum(df.iloc[0:2, 4]) / 2 + df.iloc[2, 4] + df.iloc[3, 4] + df.iloc[4, 4] + df.iloc[5, 4]) / 5) * 100
# Validity
validity = float(df.iloc[0, 5]) * 100

# Raise an error if a value is not correctly typed in the questionnaire
for index, row in df.iterrows():
    if row["Usability"] < 0 or row["Usability"] > 1:
        raise InvalidValueError("Oops, something goes wrong! Please check the excel-file if all answers only contain values between 0 and 1!")
    if row["Security"] < 0 or row["Security"] > 1:
        raise InvalidValueError("Oops, something goes wrong! Please check the excel-file if all answers only contain values between 0 and 1!")
    if row["Privacy"] < 0 or row["Privacy"] > 1:
        raise InvalidValueError("Oops, something goes wrong! Please check the excel-file if all answers only contain values between 0 and 1!")
    if row["Persistence"] < 0 or row["Persistence"] > 1:
        raise InvalidValueError("Oops, something goes wrong! Please check the excel-file if all answers only contain values between 0 and 1!")
    if row["Validity"] < 0 or row["Validity"] > 1:
        raise InvalidValueError("Oops, something goes wrong! Please check the excel-file if all answers only contain values between 0 and 1!")

# Creating the second data frame with the final quality scores
df2 = pd.DataFrame({
    "Groups": ["Results"],
    "U": [usability],
    "S": [security],
    "Pr": [privacy],
    "Pe": [persistence],
    "V": [validity]
})

# Names of attributes and number of attributes
attributes = df2.columns.values[1:]
length = len(attributes)

# Extract result values
values = df2.loc[0].drop("Groups").tolist()
values.append(df2.loc[0][1])

# Calculating angles of the radar chart
angles = []
```
for i in range(length):
    angles.append(i / float(length) * 2 * pi)
    angles.append(angles[0])

# Radar chart without data
radar_chart = plt.subplot(111, polar=True)

# Assigning variables (attributes) and angles
plt.xticks(angles[:-1], attributes, color='grey', size=12, fontweight="bold")
radar_chart.set_rlabel_position(-45)
plt.yticks([20, 40, 60, 80], ['20°', '40°', '60°', '80°'], color='grey', size=8, fontweight="bold")
plt.ylim(0, 100)

# Plot data
radar_chart.plot(angles, values, "b", linewidth=1, linestyle='dashed')
radar_chart.fill(angles, values, "b", alpha=0.1)
plt.show()


Cheng, Guangming; Li, Yao; Gao, Zhiwei; Liu, Xiaoyin (2017): Cloud data governance maturity model. In: 8th IEEE International Conference on Software Engineering and Service Science (ICSEES). Guangzhou, China, pp. 517-520.


Earley, Seth (2014): Agile Analytics in the Age of Big Data. In: IT Prof. 16 (4), pp. 18–20. DOI: 10.1109/MITP.2014.44.


Hevner, Alan; March, Salvatore; Park, Jinsoo; Ram, Sudha (2004): Design science research in information systems. In: Design research in information systems. Springer, Boston, pp. 9-22.


Kohavi, Ron; Deng, Alex; Frasce, Brian; Walker, Toby; Xu, Ya; Pohlmann, Nils (2013): Online Controlled Experiments at Large Scale. In: Proceedings of the 19th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Chicago, pp. 1168-1176.


Manyika, James; Chui, Michael; Brown, Brad; Bughin, Jacques; Dobbs, Richard; Roxburgh, Charles; Byers, Angela H. (2011): Big data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute.


Mosley, Mark; Brackett, Michael; Earley, Susan (Hg.) (2010): The DAMA guide to the data management body of knowledge. (DAMA-DMBOK guide). First edition. Bradley Beach, NJ: Technics Publications LLC.
Nancy, Grady W.; Payne, Jason A.; Parker, Huntley (2017): Agile Big Data Analytics: 
AnalyticsOps for Data Science. In: Proceedings in IEEE International Conference on Big 
Data (BIGDATA). Boston, pp. 2331-2339.

NIST (2015a): NIST Big Data Interoperability Framework: Volume 4, Security and Privacy: 
National Institute of Standards and Technology.

Architecture: National Institute of Standards and Technology.

Noorwali, Ibtehal; Arruda, Darlan; Madhavji, Nazim H. (2016): Understanding quality 
requirements in the context of big data systems. In: Proceedings of the 2nd International 
Workshop on Big Data Software Engineering. ACM, pp. 76-79.

Ohlhorst, Frank (2012): Big data analytics: turning big data into big money (Vol. 65). John 
Wiley & Sons, New Jersey.


Pääkkönen, Pekka; Pakkala, Daniel (2015): Reference Architecture and Classification of 
166–186. DOI: 10.1016/j.bdr.2015.01.001.

Perera, Charith; Wang, Lizhe; Zomaya, Albert Y. (2015): Privacy of Big Data in the Internet 

Ramaswamy, Lakshmish; Lawson, Victor; Gogineni, Siva Venkat (2013): Towards a Quality-

Rathika, V.; Arcokiam, L. (2014): Automated Data Validation Framework for Data Quality in 
Big Data Migration Projects. In: SSRG International Journal of Computer Science and 
Engineering 1(10), pp. 24-27.


19(4), pp. 1-34.

pp. 1-40.

Samsudin, Azman (2016): Big data: Related technologies, security challenges, and research 
opportunities. University of Sains Malaysia, pp. 1-70.

Scannapieco, Monica; Catarci, Tiziana (2002): Data quality under a computer science 


Smith, Ken; Seligman, Len; Rosenthal, Arnon; Kurcz, Chris; Greer, Mary; Macheret, Catherine et al. (2014): "Big Metadata". In: Proceedings of Workshop on Data analytics in the Cloud (DanaC). UT, pp. 1-4.


Stojkov, Milan; Milosavljevic, Branko; Sladic, Goran (2017): On the Usability of Access Control Models in IoT. In: 8th PSU-UNS International Conference on Engineering and Technology. Novi Sad, pp. 1-5


Xu, Ya; Chen, Nanyu; Fernandez, Adrian; Sinno, Omar; Bhasin, Anmol (2015): From Infrastructure to Culture: A/B Testing Challenges in Large Scale Social Networks. In: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 2227-2236.


Online References


The Economist (2017): The world´s most valuable resource is no longer oil, but data. 

Van Rijmenam, Mark (n.d.): How Amazon Is Leveraging Big Data. 
Plagiarism Disclaimer

I hereby declare that this diploma thesis is my own and autonomous work. All sources and aids used have been indicated as such. All texts either quoted directly or paraphrased have been indicated by in-text citations. Full bibliographic details are given in the list of works cited, which also contains internet sources including URL and access date. This work has not been submitted to any other examination authority.

........................................................................................................................................

(Date) ......................................................................................................................... (Signature)