Adopting NoSQL Databases Using a Quality Attribute Framework and Risks Analysis

Hilda Mackin, Gonzalo Perez and Charles C. Tappert
School of Computer Science and Information Systems, Pace University, 861 Bedford Road, Pleasantville, NY, USA
hd79424w@pace.edu, hdiaz19@hotmail.com,76gonzalo@gmail.com, ctappert@pace.edu

Keywords: Distributed Databases, NoSQL Databases, Relational Databases, Quality Attributes.

Abstract: NoSQL has emerged in recent years to provide increased scalability and performance, and organizations have a problem in choosing between traditional SQL and NoSQL databases. This research gives software engineers and architects a way to select a NoSQL database for a particular big data environment and domain. It proposes a Quality Attribute Framework and Risk Analysis of NoSQL databases that can measure quality metrics associated with availability and security, which are critical to choosing the right NoSQL database for a given domain and to making better software development and design decisions. The framework will help IT departments align perceived risks of NoSQL database adoption with actual risks, helping IT managers in their database adoption and in the identification of risk factors that affect the new database technologies. The framework developed here will be finalized through a qualitative analysis of risk vectors via surveys of top IT leaders and IT companies.

1 INTRODUCTION

In the past, relational databases were used for all tasks to be processed by a database because of their query capabilities, and transaction management features.

Traditional Relational databases were designed for different hardware and software times and are facing challenges in meeting the performance and scale requirements of Big Data (Grolinger et al., 2013).

The vital factor for a change in data storage was the need to support large volumes of data in distributed environments. Two companies in particular Google and Amazon have been influential with their Big Table from Google and Dynamo from Amazon papers.

One of the first solutions to increase the amount of data that could be stored by a DBMs system was to spread the data among several database servers or clusters instead of just one server.

A second important factor for a change was that distributed systems were not very efficient performing transactions and join operations using a Relational Database System (RDBMS).

Since Relational databases need to maintain strict consistency using the transactional ACID model and must be highly available a new family of NoSQL clustered databases emerged.

As more companies' considered adopting Big Data Solutions, discussions about the most appropriate NoSQL database for their use case, application or environment originated.

NoSQL databases are highly scalable, have good performance, are designed to store and process a significant amount of unstructured data faster than Relational Databases. So why to be cautious about adopting a NoSQL database? Availability and Security are major concerns for IT Enterprise Infrastructures. There are perceived Security Risks associated with the new database technologies (Obijaju, 2015).

2 RELATED WORK AND RESEARCH CONTRIBUTIONS

NoSQL data stores are seen as data processing alternatives that can handle considerable volume of data and provide better scalability, but attributes and risks associated with these new technologies are not well understood. Because of the large number and diversity of NoSQL solutions, it is challenging to

Mackin H., Perez G. and Tappert C.
Adopting NoSQL Databases Using a Quality Attribute Framework and Risks Analysis.
DOI: 10.5220/0006227600970104
In *Proceedings of the Fifth International Conference on Telecommunications and Remote Sensing (ICTRS 2016)*, pages 97-104
ISBN: 978-989-758-200-4

choose an appropriate NoSQL solution for a specific task or Use Case. Some studies have identified NoSQL challenges including the immense diversity and inconsistency of terminologies, limited documentation, sparse comparison and benchmarking criteria, and lack of standardized query languages (Grolinger et al., 2013).

Many researchers have focused on Performance Evaluation but have not included other Software Quality Attribute requirements (Lourenço et al., 2015).

A previous study created a comparison of NoSQL databases, identifying the software attributes that would aid a software engineer's decision process (Lourenço et al., 2015). It identified several desirable quality attributes to evaluate NoSQL databases: Availability, Consistency, Durability, Maintainability, Read Performance, Recovery Time, Reliability, Robustness, Scalability, Stabilization Time and Write Performance. It also selected some popular NoSQL databases: Aerospike, Cassandra, Couchbase, CouchDB, HBase, Mongo DB, and Voldemort. Their study resulted in a summary table (Table 1) to aid software engineers and architects in their decision process when selecting a given NoSQL database according to certain quality attributes.

Our research investigates, in a qualitative manner, the attributes of an effective NoSQL Quality Attribute Framework. This work will make two new contributions to the state-of-the-art.

1) Provide a straightforward and coherent way for IT Managers to understand NoSQL database quality attributes and gain some insight on mitigation strategies for current NoSQL databases risks.

2) Provide guidance to rationalize risks associated with Security and Business Availability.

This paper improves and complements the study (Lourenço et al., 2015).

1) Quality attributes are clearly defined (Section 4.2) and classified and evaluated by four types of NoSQL database types: Key-Value, Document, Columnar and Graph Database (Section 4.4).

2) This research adds Security attributes, enhances Availability attributes and adds Additional Quality Attributes to provide a complete set of quality attributes. (Table 4).

The framework will be completed shortly through a quality-attribute-focused survey based on NoSQL database type (Key Value, Document, Columnar, and Graph), where databases are compared with regards to their suitability for quality attributes.

2.1 Research Approach

The goal of this research is to gain a thorough understanding of NoSQL databases to clarify the risks associated with these new technologies and to provide a framework and recommendations to mitigate risks. Two main research problems and a sub problem are identified:

First Research Problem

What NoSQL databases Quality Attributes have the most positive and negative impacts on Security, and Business Availability risks?

What Software Quality Attributes are required to evaluate and adopt NoSQL databases?

What is the perceived Business Availability and Security Risks associated with these new database technologies in the context of Big Data?

Second Research Problem

Are there any misalignments between actual and perceived NoSQL database risks uncovered through surveys and the risks perceived by experts and IT professionals?

What are the technical and non-technical impacts on the organization?

First Research Sub-problem

Are Relational and NoSQL databases totally mutually exclusive?

How do applications intersect between Relational and NoSQL databases?

If the technologies do not overlap, but intersect choose between the databases?

3 NOSQL DATABASES

The NoSQL movement is a contemporary approach to data persistence using novel storage methods (Franks, 2012). NoSQL databases were built to deal with the increasing amount of complex data, required in some real-time applications, to address availability over consistency, and to allow horizontal scalability, using a distributed architecture and open source (Bazar & Cosmin, 2014).

The common characteristics of NoSQL databases are (Sadalage & Fowler, 2013):

- Not using the relational model
- Running well on clusters
- Open source projects
- Built for the 21st-century web estates
- Schema-less.

3.1 Data Model

There are four accepted types of NoSQL databases:

Key Value Stores, Consist of keys and their corresponding values, data is stored in a schema-less way. This allows search millions of values in a fraction of the time needed by conventional storage (Franks, 2012).

Document Stores, Consist of a set of documents possibly nested. Data can be structured in a schema-less way or in the form of collections. Popular examples are Couch Base, Mongo DB (Franks, 2012).

Column Stores or *Extensible Record Stores*, Consist of tables which may have different schema for each row having one huge extensible column containing the data. Column stores do not declare null fields as relational databases do (Franks, 2012).

Graph Stores, Consist of a set of graph nodes linked together by edges (providing index-free adjacency of nodes) (Barmpis & Kolovos, 2012).

Each type of NoSQL database is tailored for storing a different and specific type of data (Barmpis & Kolovos, 2012).

3.2 Advantages of NoSQL Databases

Most NoSQL data stores run on clusters. Relational databases use ACID transactions to handle consistency across the entire database. In contrast in a cluster environment, NoSQL databases offer a range of options for uniformity and distribution.

NoSQL databases operate without a schema, allowing one to freely add fields to database records without having to define any changes in structure first. This is particularly useful when dealing with non-uniform data and custom fields.

As a result, Relational databases are seen as just one option for data storage. The result is that many organizations will have a mix of data storage technologies or Polyglot Persistence.

Besides handling data access with sizes and performance that demand a cluster, the other important reason for NoSQL technology is the impedance mismatch problem. Big Data concerns have created an opportunity for people to think in a different way about their data storage needs. Some development teams see that using a NoSQL database can help their productivity by simplifying their database access. This can be achieved even if they have no reason to scale beyond a single machine, improving the productivity of application development by using a more suitable data interaction style (Sadalage & Fowler, 2013).

4 WORK PLAN

The work plan will be composed of the following sections:

1. Presenting a previous study's comparison of NoSQL engines and their quality attributes.
2. Defining a framework by enlarging the quality attributes to include Security attributes, enhancing Availability attributes and adding Additional Quality Attributes.
3. Two additional NoSQL engines are added to the study, illustrating two examples each of the four types of NoSQL databases.
4. NoSQL databases proposed are categorized into the four NoSQL database types.
5. The quality attributes of the two examples of each type are consolidated to arrive at a direct comparison of the four NoSQL database types.
6. Finally, the set of attributes are enlarged to include Security, Availability and Additional Attributes, obtaining a framework to form the basis for a subsequent qualitative study.

4.1 Previous NoSQL Database Study

A previous study created a comparison of NoSQL databases, identifying the software attributes that would aid a software engineer's decision process (Lourenço et al., 2015).

Their study resulted in Table 1 to aid software engineers and architects in their decision process when selecting a given NoSQL database according to certain quality attributes.

The table uses a 5-point scale ranging from "Great" to "Bad" to allow a direct comparison among databases. For example, Cassandra is written based on a performance-oriented approach, more than Couch base. Worse grades were assigned when a database was not an ideal pick, according to their authors' literature revision. This does not mean that the database lacked the attribute entirely, but it was not the best compared to the others databases (Lourenço et al., 2015).

The quality attributes have the following meanings:

Availability: Downtime was used as a primary measure.

Consistency: Graded according to ACID semantics consistency and how much consistency can be fine-tuned.

Durability: Measured according to the use of single or multi-version concurrency control schemes, the way the data are persisted to disk.

Maintainability: Measured for ease of setup and use, accessibility of tools to interact with the database.

Read Performance: Considered studies about the fine tuning of each database.

Write Performance: Considered studies about the fine tuning of each database.

Recovery Time: Related to availability, took results from a previous study.

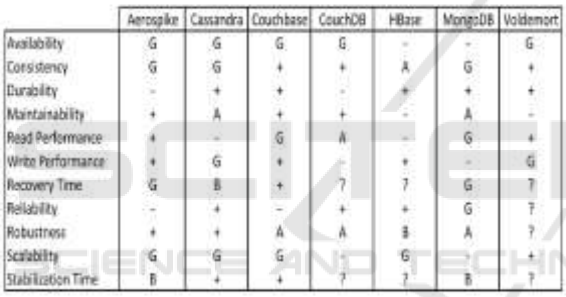
Reliability: Looked at synchronous propagation modes

Robustness: Considered the tendency of databases to have problems dealing with crashes or attacks.

Scalability: Looked at each database elasticity, horizontal scaling, and ease of online scalability.

Stabilization Time: Related to availability.

Table 1: Quality attributes for popular NoSQL engines (redrawn from (Lourenço et al., 2015)). Legend: "G" = Great, "+" = Good, "A"= Average, "_" = Mediocre, "B" = Bad, and "?" = Unkown/NA.



	Aerospike	Cassandra	Couchbase	CouchDB	HBase	MongoDB	Voldemort
Availability	G	G	G	G	-	-	G
Consistency	G	G	+	+	A	G	+
Durability	-	+	+	-	+	+	+
Maintainability	+	A	+	+	-	A	-
Read Performance	+	-	G	A	-	G	+
Write Performance	+	G	+	-	+	-	G
Recovery Time	G	B	+	?	?	G	?
Reliability	-	+	-	+	+	G	?
Robustness	+	+	A	A	B	A	?
Scalability	G	G	G	-	G	-	+
Stabilization Time	B	+	+	?	?	B	?

The study concluded that even though there was a variety of other research and evaluations of NoSQL technology, there was still not enough information to verify how fit a NoSQL database is in a specific scenario or system, making the following recommendation for future work: "The development of a framework for assessing most of these quality attributes would greatly benefit the database adoption of software engineers and architects" (Lourenço et al., 2015).

4.2 Proposing a Framework

The framework proposed covers Availability and Security attributes highly valued by NoSQL databases users. Definitions of the quality attributes included in the study are given below:

Availability: The percentage of time a system is operating correctly. Is the data always accessible? Is the data permanently available? In the context of the CAP Theorem Availability was evaluated versus Consistency and Availability versus Partition Tolerance.

Consistency: The valid and reliable data that is saved in every cluster node. Is the data the same in every replication on every cluster node? In the context of CAP Theorem Consistency is assessed when all nodes see the same data at the same time.

Partitioning: Defined as the data divided in smaller segments to be allocated in different data store tables. Is Horizontal Partitioning or Vertical Partitioning allowed?

Replication: Keeping a copy of the data in different databases and servers. Is Replication transparency allowed? Is Replication considered in different layers? Does Master slave and master replication have one instance or have different instances?

Scalability: Related to horizontal scaling. Is Scaling achieved by replicating the data synchronously or asynchronously?

Shared Nothing: Are all replica nodes allowed to continue working even if they are disconnected?

Recovery Time: Related to the time it takes for several NoSQL systems to recover from a node failure.

Stabilization Time: Related to the time it takes for the system to stabilize when that node re-joins the cluster.

Reliability: System's probability of operating without failures for a given period.

Robustness: Defined as the ability of the database to cope with error during execution

Durability: Property that guarantees that a transaction that has been saved/commit in the database will be committed even in the event that the system crashes

Maintainability: Does the NoSQL database provide features for easy maintainability, administration, management and operation?

Read Write Performance: Is the NoSQL database more robust on Reading operations than writing operations?

Security: Defined as the software and the set of management tools that protect the database against attacks, hackers, and viruses. Some properties that need to be analysed in the study are:

Authentication: Related to password and user's login. What types of authentication does the NoSQL database provides?

Authorization: Defined as a set of read and write permissions request on tables, creation of users. What administrative functions the NoSQL database provides?

Encryption: Does the NoSQL database provide mechanisms that allow encryption techniques that preserve data confidentiality? If not what mechanisms are used to enforce data confidentiality?

Three levels will be explored in the study: Data at rest, Client to server communication, Server to Server connection.

Auditing: Does the NoSQL database provide mechanisms that allow writing to the database or Audit Logs?

Data integrity: Does the NoSQL database provide mechanisms that allow data integrity such as ACID or eventually consistent BASE? Could the NoSQL database achieve different levels of Data Integrity?

Confidentiality: Data Confidentiality. Does the NoSQL database provide different mechanisms to preserve data confidentiality?

Documentation: Does the NoSQL database provide End User Documentation? What levels of documentation does it provide?

Additional Quality Attributes –Additional attributes will be addressed in the study as part of the contribution envisioned. These include attributes related to the ease of developing when using NoSQL databases:

Popularity Depending on the Type on NoSQL database (Key Value, Document, Columnar or Graph) some databases are more popular than others. Different aspects need to be analysed to determine what database is applicable to the business case and from that infer the popularity.

Maturity: Considering Maturity of the API, time in the market, and Enterprise adoption.

Query Possibilities: Does the NoSQL database provide SQL like query possibilities?

Concurrency Control: Does the NoSQL database provide features to manage Concurrency Control? Does it provide Optimistic or Multi Version concurrency control?

Conflict Resolution: Does the NoSQL database provide mechanisms to manage Conflict Resolution?

4.3 Selecting NoSQL Databases

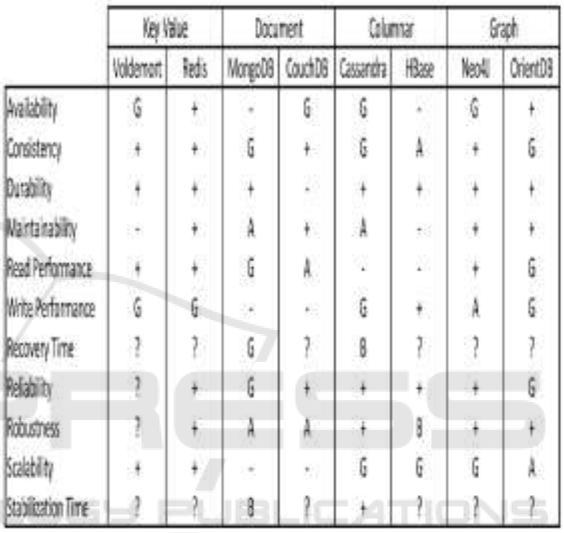
Considering four NoSQL database types (Key Value, Document, Columnar, and Graph) the study selects eight popular NoSQL databases used by enterprises, two databases to represent each type. These database types are now compared with regards to their suitability for quality attributes.

The database selection was based on literature research and data collected from preliminary interviews: Voldemort, Redis, Mongo DB, Couch DB, Cassandra, HBase, Neo4J, OrientDB.

4.4 NoSQL Database Types

We now categorize these eight popular NoSQL engines into the four NoSQL database types, taking five of the engines from Table 1 (Voldemort, MongoDB, CouchDB, Cassandra, HBase) and adding three new engines (Redis, Neo4J, and OrientDB).

Table 2: Quality attributes for eight popular databases, two of each type. Legend: “G” = Great, “+” = Good, “A” = Average, “-” = Mediocre, “B” = Bad, and “?” = Unknown/NA.



	Key Value		Document		Columnar		Graph	
	Voldemort	Redis	MongoDB	CouchDB	Cassandra	HBase	Neo4J	OrientDB
Availability	G	+	-	G	G	-	G	+
Consistency	+	+	G	+	G	A	+	G
Durability	+	+	+	-	+	+	+	+
Maintainability	-	+	A	+	A	-	+	+
Read Performance	+	+	G	A	-	-	+	G
Write Performance	G	G	-	-	G	+	A	G
Recovery Time	?	?	G	?	B	?	?	?
Reliability	?	+	G	+	+	+	+	G
Robustness	?	+	A	A	+	B	+	+
Scalability	+	+	-	-	G	G	G	A
Stabilization Time	?	?	B	?	+	?	?	?

Two examples illustrate each of the four types, as shown in Table 2.

The attributes for the engines from Table 1 come from (Lourenço et al., 2015), the attributes for Redis and Neo4J from (Redmond, 2012), and the attributes for Orient Database from (Orient Database 2016)

The quality attributes of the two examples of each type are consolidated to arrive at a direct comparison of the four NoSQL database types (Table 3). Averaging the attribute values for the two example engines of each type was done conservatively by leaning toward the lower rating on the 5-point scale.

4.5 Expert Opinions and Positions

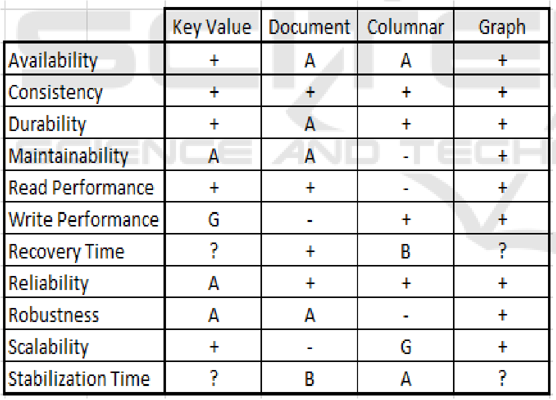
Due to the diversity of NoSQL solutions and four different types of NoSQL databases, making the choice of the most appropriate data store for a given use case scenario will be easier with the framework proposed. Table 3 is now extended by adding the security and additional quality attributes discussed above to obtain Table 4.

With the framework illustrated in Table 4, we will reach out for expert opinions and positions through interviews and surveys. A quality-attribute-focused survey will be created, based on NoSQL database type (Key Value, Document, Columnar, and Graph), where databases are compared with regards to their suitability for quality attributes. The anticipated survey results should allow completion of blank spaces on Table 4.

This research selected a methodology based on finding qualitative measures and understanding quality attributes of NoSQL databases by leveraging the knowledge of NoSQL database specialists and other early adopters users. It will involve:

- Survey of enterprises and experts
- Anonymity of Participants
- Iterations
- Controlled Feedback
- Statistical Aggregation of Group Responses
- Research Tasks Inputs and Outputs.

Table 3: Averaged quality attributes for the four NoSQL database types. Legend: "G" = Great, "+" = Good, "A" = Average, "-" = Mediocre, "B" = Bad, and "?" = Unknown/NA.



	Key Value	Document	Columnar	Graph
Availability	+	A	A	+
Consistency	+	+	+	+
Durability	+	A	+	+
Maintainability	A	A	-	+
Read Performance	+	+	-	+
Write Performance	G	-	+	+
Recovery Time	?	+	B	?
Reliability	A	+	+	+
Robustness	A	A	-	+
Scalability	+	-	G	+
Stabilization Time	?	B	A	?

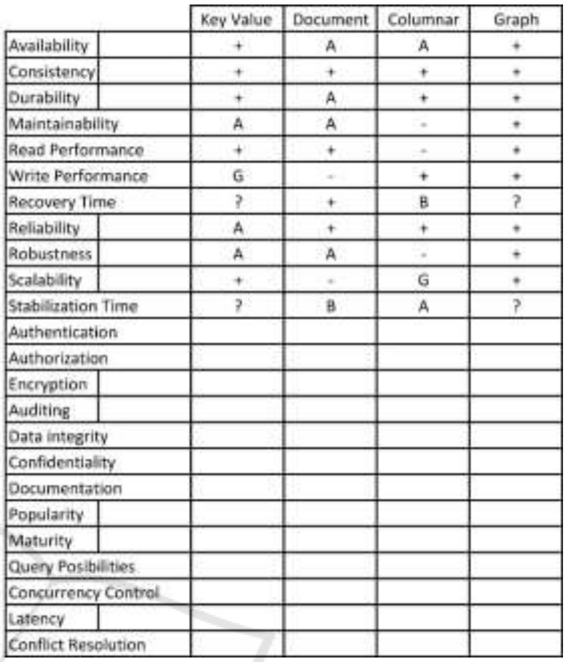
4.6 Surveys

Example of the Survey with three brief sections is listed below.

Please select the database used:

- Voldemort(Key Value)
- Redis (Key Value)
- Mongo DB (Document)
- Couch DB (Document)
- Cassandra(Columnar)
- HBase (Columnar)
- Neo4J(Graph)
- OrientDB (Graph)

Table 4: Averaged quality attributes for the four NoSQL database types with the framework proposed. Legend: "G" = Great, "+" = Good, "A" = Average, "-" = Mediocre, "B" = Bad, and "?" = Unknown/NA.



	Key Value	Document	Columnar	Graph
Availability	+	A	A	+
Consistency	+	+	+	+
Durability	+	A	+	+
Maintainability	A	A	-	+
Read Performance	+	+	-	+
Write Performance	G	-	+	+
Recovery Time	?	+	B	?
Reliability	A	+	+	+
Robustness	A	A	-	+
Scalability	+	-	G	+
Stabilization Time	?	B	A	?
Authentication				
Authorization				
Encryption				
Auditing				
Data integrity				
Confidentiality				
Documentation				
Popularity				
Maturity				
Query Posibilities				
Concurrency Control				
Latency				
Conflict Resolution				

4.6.1 Availability Survey

1. Is Horizontal Partitioning allowed? Yes, No
2. Is Vertical Partitioning allowed? Yes, No
3. Is Scaling achieved by replicating the data? (Please Select)
   Synchronously
   Asynchronously
   Both
4. Are all replica nodes allowed to continue working even if they are disconnected? Yes, No
5. Is the NoSQL database more robust on Reading operations? Yes, No
6. Is the NoSQL database more robust on writing operations? Yes, No
7. Is the data permanently available? Yes, No

8. Is the data the same in every replication on every cluster node? Yes, No

9. Does the NoSQL database provide mechanisms that allow ACID data integrity? Yes, No

10. Does the NoSQL database provide mechanisms that allow eventually consistent BASE? Yes, No

4.6.2 Security Survey

1. Does the NoSQL database provide mechanisms that allow encryption techniques? Yes, No
   If "Yes" Please name the encryption technique
   If "No" what mechanisms are used to enforce data confidentiality?
2. What encryption level does the NoSQL database provide? (Please Select)
   - Data at rest
   - Client to server communication
   - Server to Server connection
3. Does the NoSQL database provide mechanisms that allow Auditing or Audit Logs? Yes, No
4. Does the NoSQL database provide mechanisms that allow Authentication? Yes, No
   If "Yes" please mention the Authentication method
5. Does the NoSQL database provide mechanisms that allow Authorization? Yes, No
   If "Yes" please mention the Authorization method
6. Is Replication on the database allowed? Yes, No
7. How often are backups tested? Daily, Weekly
8. How often is Disaster Recovery Infrastructure tested? Daily, Weekly

4.6.3 Additional Quality Attributes Survey

1. What is the maturity of the Database - API in the market according to your experience)? (Please Select)
   Mature (5- 10 years)
   Growing (2 - 5 years)
   Start Up (0 to 2 years)
2. In what Business Area the NoSQL Database is used in your environment? (Please Select)
   Medical
   Financial
   Retail
   Social Media
   Other (Please Describe)
3. How long have the database been used in the Business Area selected previously? (Please Select)
   Popular (5- 10 years)
   Growing (2 - 5 years)
   Start Up (0 to 2 years)
   Other (Please Describe)
4. Does the NoSQL database provide SQL like query possibilities? Yes, No
   If "Yes" please mention the SQL Query name
5. Does the NoSQL database provide features to manage Concurrency Control? (Please Select)
   Optimistic concurrency
   Multi Version concurrency
   Other
   If "Other" please mention the Concurrency Control
6. Is the NoSQL database coder friendly? Yes, No
7. Is the NoSQL database used as a caching Layer? Yes, No
8. Is the NoSQL database used on Real Time Analysis? Yes, No
9. Is the NoSQL database used on Analytics? Yes, No
10. Does the NoSQL database provide End User Documentation? Yes, No

The result of this methodology will be a quality attribute framework and risks analysis of adopting NoSQL databases, which will aid software engineers and architects in their decision process when selecting a NoSQL database according to their software quality, attributes requirements.

5 CONCLUSION

There are a number of NoSQL data stores that can be classified into four different types. However, there is no Quality Software Framework that can help managers decide which NoSQL databases are the most appropriate for their Business Use Case.

The diversity of NoSQL data stores present challenges to differentiate and get a perspective of which databases is the most suited, establishing paths and opportunities for future research.

Sophisticated Security and Privacy provisions need to be explored. At the corporate level, companies and institutions need to develop software technology that offers Security features at the minimum similar if not better than the ones used by Relational Databases.

Considering a previous study's comparison of NoSQL databases and their quality attributes, the contribution of this research includes Security attributes, enhances Availability attributes and adds Additional Quality Attributes to define a Quality Attribute Evaluation and Risk Analysis of NoSQL databases framework that will benefit the NoSQL adoption in the long term.

The framework proposed will help IT departments align perceived risks of NoSQL database adoption with actual risks measuring quality metrics associated with Availability and Security, which are critical to choosing the right NoSQL database for a given domain and to making better software development and design decisions, giving software engineers and architects a better way to select a NoSQL database for a particular big data environment and domain.

REFERENCES

Barmpis Konstantinos, Kolovos Dimitrios S. 2012, 'Comparative Analysis of Data Persistence Technologies', Proceedings of the 2012 Extreme Modeling Workshop, New York.

Bazar Christina, Cosmin Sebastian 2014, 'The Transition from RDBMS to No SQL. A comparative Analysis of three popular No Relational solutions: Cassandra, Mongo DB and CouchBase', Database Systems Journal, University of Economic Studies, Bucharest, Romania, Romania.

Catell Rick 2010, 'Scalable SQL and No SQL Data Stores', ACM SIGMOD Record, ACM Digital Library, New York.

Doshi Kshitij A.,Zhong Tao,Lu Zhongyan,Tang Xi 2013, 'Blending SQL and NewSQL Approaches, Reference Architectures for Enterprise Big Data Challenges', International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC).

Fowler Adam 2014, 'Why use a NoSQL Database, and why not?', Adam's Blog.

Franks Bill 2012, Taming the Big Data Tidal Wave, finding opportunities in Huge Data Streams with Advanced Analytics, 1st edn, Wiley.

Grolinger Katarina, Higashino Wilson A,Tiwari Abhinav,Capretz Miriam 2013, 'Data management in cloud environments: No SQL and New SQL data stores', Journal of Cloud Computing, Springer Open, London.

Gualtieri Mike 2010, 'No SQL and Elastic Caching Platforms Are Kissing Cousins', Forrester, http://blogs.forrester.com/application_development/2010/02/nosql.html.

Himmel Azua Maria 2012, Qualitative Analysis of Cloud Computing Risks and Framework for the Rationalization and Mitigation of Cloud Risks, Pace, New York.

Intersimone David 2010, 'The end of SQL and relational databases?', Computer World, February 2010.

Kroenke David M., ADJ, Database Processing, 13th edn.

Lakshman Avinash 2008, 'Cassandra – A structured storage system on a P2P Network', https://www.facebook.com/notes/facebook-engineering/cassandra-a-structured-storage-system-on-a-p2p-network/24413138919/.

Lourenço João Ricardo, Cabral Bruno, Carreiro Paulo, Vieira Marco, Bernardino Jorge 2015, 'Choosing the right NoSQL database for the job: a quality attribute evaluation', Springer Open Journal, no. Journal of Big Data.

Nepa Suryal, Ranjan Rajiv, Kwang Kim, Choo Raymond 2015, 'Trustworthy Processing of Healthcare Big Data in Hybrid Clouds', IEEE Cloud Computing.

Obijaju Mike 2015, 'No SQL NoSecurity – Security issues with No SQL Database', June 2015, https://blogs.perficient.com/enterpriseinformation/2015/06/22/nosql-nosecuity-security-issues-with-nosql-database/.

Orend Kai 2012, Analysis and Classification of No SQL Databases and Evaluation of their ability to replace an Object Relational Persistence Layer, University of Munchen.

Orient Database 2016, 'Orient DB', http:// orientdb.com/orientdb/.

Penchikala Srini 2011, 'Distributed Cache as a No SQL Data Store?', InfoQ, http://www.infoq.com/news/2011/11/ distributed-cache-nosql-data-sto.

Redmond Eric, WJR 2012, Seven Databases in Seven Weeks, The Pragmatic Programmers.

Robin Hecht, SJ 2011, 'No SQL Evaluation', International Conference on Cloud and Service Computing.

Robin Hecht 2011, 'No SQL Evaluation', International Conference on Cloud and Service Computing, IEEE Computer Society, Hong Kong.

Sadalage Pradmod J.,Fowler Martin 2013, NO SQL Distilled. A brief guide to the emerging world of Polyglot Persistence, 1st edn, Addison Wesley.

Stonebracker Michael, 'New SQL: An Alternative to NoSQL and Old SQL for New OLTP Apps'.

Tudorica Bogdan George., Bucur Christian 2011, 'A comparison between several NoSQL databases with comments and notes', 2011 10th International Conference RoEduNet, IEEE.

Wikipedia, 'Cassandra', Wikipedia, https://en.wikipedia.org/wiki/Cassandra.