

# **Formative Evaluation of Data Management Tools on Data Quality and Usability**

Hian Zing Voon



**TUM**



Bachelor's thesis

# Formative Evaluation of Data Management Tools on Data Quality and Usability

Hian Zing Voon

October 31, 2024



Chair of Data Processing  
Technische Universität München



Hian Zing Voon. *Formative Evaluation of Data Management Tools on Data Quality and Usability*. Bachelor's thesis, Technische Universität München, Munich, Germany, 2024.

Supervised by Prof. Dr.-Ing. Klaus Diepold, Dr.-Ing. Stefan Röhrl and M.Sc. Bastian Busch; submitted on October 31, 2024 to the Department of Electrical and Computer Engineering of the Technische Universität München.

© 2024 Hian Zing Voon

Chair of Data Processing, Technische Universität München, 80290 München, Germany, <http://www.ldv.ei.tum.de/>.

This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

# Abstract

Big data and machine learning applications significantly impact business performance across companies of all sizes. High-quality data are crucial for enabling engineers to automate routine tasks, allowing them to carry out their daily tasks and responsibilities more efficiently. However, achieving high data quality continues to be a challenge because inaccurate, incomplete and inconsistent data can compromise the effectiveness of automation processes. More often than not, these data are managed by software or tools familiar to a wide range of users, such as *Microsoft Excel*. These tools are often intuitive and thus user friendly, but may not provide high-quality data due to their extensive degree of freedom. In this thesis, the data quality and usability of an existing *Microsoft Excel* tool is compared to a prototype for data management. To achieve this, we explore specific dimensions of data quality that fit our needs and examine qualities that define good usability for a system. With this information, we can identify the strengths and weaknesses of the prototype and iterate the user interface development process to improve its design based on valuable feedback. The findings are also applicable in other relevant fields that require high data quality and system usability in our daily lives.



# Contents

- Abstract** **3**
  
- 1. Introduction** **7**
  
- 2. Theoretical Background** **9**
  - 2.1. User Interface Development . . . . . 9
  - 2.2. Data Quality and Weighted Metric . . . . . 10
    - 2.2.1. Accuracy . . . . . 12
    - 2.2.2. Completeness . . . . . 14
    - 2.2.3. Consistency . . . . . 15
    - 2.2.4. Weighted Metric . . . . . 17
  - 2.3. Usability Evaluation . . . . . 19
    - 2.3.1. Evaluation Setup . . . . . 19
    - 2.3.2. Data Collection . . . . . 21
  
- 3. Implementation** **25**
  - 3.1. Prototype . . . . . 25
  - 3.2. Preparation for Data Quality Evaluation . . . . . 29
  
- 4. Evaluation** **31**
  - 4.1. Study Setting . . . . . 31
  - 4.2. Data Quality . . . . . 34
    - 4.2.1. Objective Assessment . . . . . 35
    - 4.2.2. Subjective Assessment . . . . . 39
  - 4.3. System Usability . . . . . 40
  
- 5. Discussion** **43**
  
- 6. Conclusion** **45**
  
- Bibliography** **49**
  
- A. Appendix** **53**





# 1. Introduction

In the current era of the digital economy, digital transformation signifies an organizational shift that integrates digital technologies and business processes [25]. Successful digital transformation of organizations ensures their survival in a fast-paced world. Many companies fall behind the competition or fail entirely due to changes brought upon by digital transformation strategies. Digital transformation is no longer an opportunity for technological growth but rather essential to handle the needs and expectations of the world population [21].

Corporations of all sizes increasingly count on big data and machine learning applications for business performance [13]. According to a survey conducted by Deloitte based on 800 C-level and business unit leaders in 2019 and in 2023 [33], 68% of companies have met or exceeded their goals during digitalization to improve their productivity, product quality and innovation. The availability and quality of data are prerequisites for automating repetitive and error-prone tasks using automated decision-making applications [28]. Therefore, the development of digital solutions for managing company data is a crucial driver for business performance.

Infineon is a semiconductor manufacturer that produces chips for various industries. Prior to the release of products to customers like original equipment manufacturers and Infineon customers, new technologies are thoroughly tested to prevent failure of components. Quality-related failures require quick actions to prevent damage to company reputation [34], which can cost the company a lot of money. To prevent this issue, samples are tested for their electrical performance early in the development process. In the context of this work, a test sample is a real, physical piece of semiconducting material that can be tested for its reliability.

These samples are not always tested at the site of their production, so their logistical tracking is a challenge. Many data sources for samples exist throughout the company. They serve as inputs for the data management tool used within the department to handle samples. Reliability engineers today face the challenge of effectively tracking these test samples and obtaining relevant data about their samples from the “source of truth systems”.

The *Single Source of Truth* refers to data management by storing crucial enterprise data and information in one central location, which can be accessed by all members of an organization [45]. Relevant data from a reliability standpoint might

## 1. Introduction

be the recipe used to produce the sample, as well as any known defects and process aberrations. Without a uniform tool connected to other company databases, significant manual work is needed for the reliability engineers to do their jobs.

Establishing a *Single Source of Truth* can improve the workflow of the reliability engineers. An example is the "Test Sample" workflow as shown in Figure 1.1. This process begins with engineers registering the samples and storing them in designated locations. Next, they plan and determine where to send these samples for testing. Lastly, they conduct the reliability tests. This system must link to existing source systems while reflecting the workflow inside the department related to sample testing. This thesis will mainly focus on the first part, which is the registration of samples.



**Figure 1.1:** An example of the workflow of reliability engineers, known as the "Test Sample" workflow

The current issue is that test samples are stored in different locations and regularly checked for scrapping due to limited office and lab space. There is no systematic and standardized procedure for archiving such data in place. Engineers in charge of the tests must manually add this information to their own *Microsoft Excel* sheets. This poses a significant amount of manual work for the engineers, which can be automated. Furthermore, there is currently no link between samples and an existing test plan tool, causing an extra step of manual labor for transferring sample data.

This thesis aims to improve the data quality and usability of the current data management tool. The practical part of this thesis focuses on replacing the existing *Microsoft Excel* based solution with a new standardized tool. It is divided into three main sections: theoretical background, implementation and evaluation. First, we do a literature review on user interface development, data quality and usability. We will also explore the methods to measure data quality and usability in both objective and subjective ways. Next, we create a prototype tool for managing test samples while taking the aspects of data quality and usability into consideration during the implementation process. Lastly, we evaluate and compare the data quality and usability between the current tool and the prototype to determine which one performs better.

In the near future, the department aims to produce a unified software to digitalize the "Test Sample" workflow, so that engineers will be able to efficiently conduct data analysis for their work with high-quality data.

## 2. Theoretical Background

The theoretical part of this thesis investigates the iterative process in user interface development and data quality for an in-house software application aimed at improving the workflow of reliability engineers. The tracking of samples tested by the reliability engineers generates a significant amount of data. As around 200 engineers in the company use this data throughout the development cycle worldwide, a scalable and efficient approach is required to ensure users have access to it.

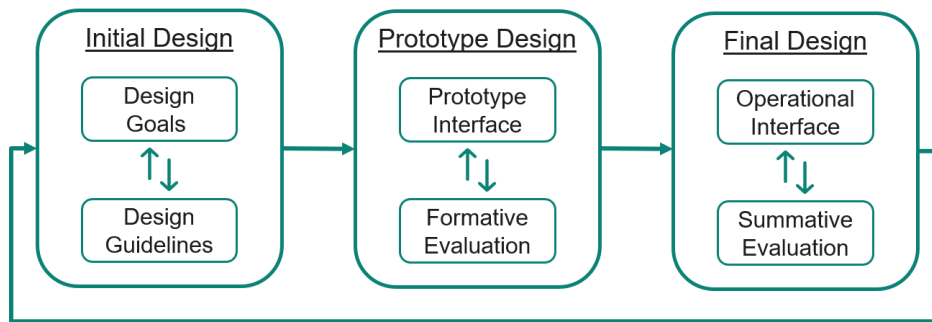
### 2.1. User Interface Development

User interface development is an iterative process that involves multiple aspects. Gould et al. [14] emphasize that iterative development is necessary for system improvement. Hartson and Boehm-Davis [18] mention two key points as to why this process must be an iteration. The life cycle of user interface development is self-correcting, which depends on trial and error as well as feedback from evaluations. Additionally, this iteration process is crucial to predict human behavior, as developers do not get it right the first time. An example of its usage is predicting patterns of interaction between humans and the software interface.

Kies et al. [20] mention that the process of user interface development cannot be like the conventional top-down, waterfall model used in software engineering. Below is Figure 2.1, which represents a combination of process designs for user interface development [14, 20], categorized into three phases.

The first phase is the initial design. In this phase, developers conduct systems analysis such as task and user analysis to identify design goals while adhering to design specifications to filter out bad designs. The second phase is the prototype design. Developers produce prototypes, especially the ones with appearance and behavior that are as close as possible to the actual system and conduct formative evaluations to improve designs. In the final design phase, developers ensure prototypes are functional with interactive components that users can engage with and conduct summative evaluations to refine designs. If needed, developers will iterate this process starting in the first phase. This thesis mainly focuses on the prototype design phase, especially the formative evaluation part. To conduct an effective formative evaluation for feedback with the goal of understanding which aspects of

## 2. Theoretical Background



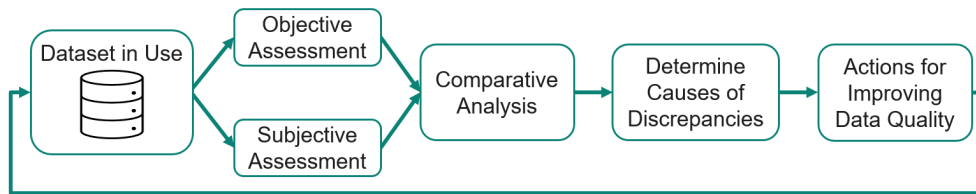
**Figure 2.1:** Process designs for user interface development, adapted from [20]

the interface are good and bad in order to improve the prototype design, a comprehensive understanding of data quality and usability evaluation is essential. This is further elaborated in the following sections.

### 2.2. Data Quality and Weighted Metric

Multiple researchers have proposed several definitions of data quality, leading to ambiguity. One frequently used definition of data quality is "fitness for use", which means the capability of suitable data collection to fulfill user requirements [29, 43]. Brodie [7] defines data quality as "a measure of the extent to which a database accurately represents the essential properties of the intended application". Brodie also states that data reliability, logical integrity, and physical integrity are unique components of data quality. The International Organization for Standardization (ISO) defines data quality as the "degree to which the characteristics of data satisfy stated and implied needs when used under specified conditions" in ISO/IEC 25012 [36]. As the concept of quality, which depends on user requirements, varies from one organization to another, these definitions of data quality are subjective. Thus, we can conclude that there is no universal definition that we can apply in all cases to define data quality. Alizamini also shares this opinion [1].

Data quality is not just a single, straightforward concept but a multi-dimensional one [9]. There are many data quality dimensions and the 10 most cited ones as summarized by Wand and Wang [42] are accuracy, reliability, timeliness, relevance, completeness, currency, consistency, flexibility, precision and format. Each dimension focuses on a specific data characteristic, which makes data useful and reliable. Examples of data characteristics include data values, data views, and data representation, i.e., the actual content of data, how data is presented to the user, and how data is formatted and structured.



**Figure 2.2:** Process for framework "Data Quality Assessment", adapted from [30]

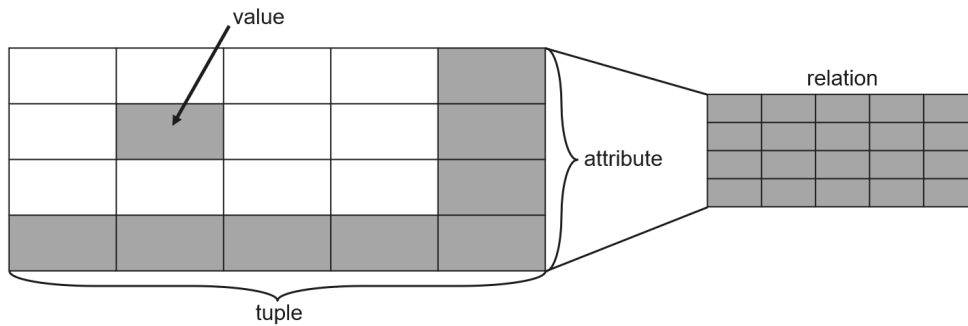
To improve data quality, we must first understand what it means and how it is measured, as proposed by Wand and Wang [42]. Researchers have proposed several frameworks to assess, manage, and improve data quality in a structured and systematic way across various dimensions. ISO has proposed a framework in ISO/IEC 25024 [38] to measure each data dimension. Another framework proposed by Pipino et al. [30] called "Data Quality Assessment" utilizes an iterative process as seen in Figure 2.2 to improve data quality. It compares objective measurements and subjective assessments to identify the root causes of discrepancies in the data, which can be used to determine necessary actions for improvement. This framework utilizes simple ratio, min or max operation and weighted average as the objective measurement, which as indicated by the authors, could be refined per use case. Furthermore, the authors also suggest that there is no "one size fits all" set of metrics to measure data quality.

Hence, this thesis utilizes the data quality definition of "fitness for use" to improve the data quality of structured data. A frequently encountered example of structured data are spreadsheets consisting of data organized into rows and columns. The standard ISO/IEC 25024 [38] will not be utilized as it is easier to implement other more practical frameworks in our use case. In order to fit the needs and meet users' expectations, we use the objective or quantitative measurement part of the framework, "Data Quality Assessment", as a guideline and focus on the data quality dimensions of accuracy, completeness and consistency.

### Data Unit

In order to assess the data quality of structured data, we can measure it using different data units. Figure 2.3 demonstrates the four data units used in this thesis: value, tuple, attribute and relation [5, 40]. A *value* is a single cell. Generally, a *tuple* is an ordered list of values, though it is a horizontal row in this context. An *attribute* is a vertical column and a *relation* is the entire structured table. To ease understanding depending on the context in this thesis when referring to data units, *tuple* and *row* will be used interchangeably, as well as *attribute* and *column*.

## 2. Theoretical Background



**Figure 2.3:** Data units for structured data, adapted from [5]

### 2.2.1. Accuracy

Accuracy is generally defined as the number of data attributes that correctly represent the true value of a concept or event in specific use cases [36]. In Batini and Scannapieca's book [5], there are two main aspects of accuracy, namely **syntactic accuracy** and **semantic accuracy**. Syntactic accuracy is defined as the closeness between a value  $v$  and the corresponding elements of domain  $D$ . Domain  $D$  is a definition domain that includes all the correct representations the user defines. On the other hand, semantic accuracy is defined as the closeness between a value  $v$  and the true value  $v'$  in real-life phenomena, which  $v$  tries to portray.

Let us take a *Delivery Schedule* relation as an example to illustrate syntactic accuracy as displayed in Table 2.1. We define domain  $D$  for the column "City" to be {"München", "Garching bei München", "Neubiberg"}. For the row containing "Alice",  $v$  is "München", then  $v$  is syntactically inaccurate as  $v$  is not an element of  $D$ .

Name	Address	Postal Code	City	Delivery Date
Alice	Arcisstraße 21	80333	München	01.09.2024
Bob	Boltzmannstraße 15	10623	Garching bei München	15.09.2024
Charlie	Am Campeon 1-15	85579	Neubiberg	30.09.2024

**Table 2.1.:** Example of a *Delivery Schedule* relation with syntactic and semantic inaccuracy

For semantic accuracy, let us first define domain  $D$  for the column "Postal Code" to be {"80333", "10623", "85579"}. For the row containing "Bob",  $v$  is "10623", then  $v$  is syntactically accurate because it is an element of  $D$ . How-

ever,  $v'$  is "85748", which is the true postal code of the city "Garching bei München" in Bavaria, Germany. Then,  $v$  is semantically inaccurate because  $v$  differs from  $v'$ . It can be seen that the accuracy dimension matters because both syntactic and semantic inaccuracies can lead to wrong parcel deliveries.

For this thesis, only syntactic accuracy is considered. This is due to the potential inaccuracies that may arise during the process of gathering metrics to assess semantic accuracy. Verifying data against the actual values in real life requires physical verification, which involves manual labor and may lead to important details being overlooked.

Batini et al. [4] define that we can determine the syntactic accuracy of tuple  $t$

$$Acc(t) = \frac{\sum_{i=1}^{|t|} acc(r_i, D(r_i))}{|t|}, \quad (2.1)$$

where  $r_i$  is the  $i$ th value of tuple  $t$ ,  $|t|$  is the number of attributes in the tuple and  $acc(r_i, D(r_i))$  is defined as

$$acc(r_i, D(r_i)) = \begin{cases} 1, & \text{if } r_i \in D(r_i), \\ 1 - NED(r_i, D(r_i)) & \text{otherwise.} \end{cases} \quad (2.2)$$

The  $acc(r_i, D(r_i))$  returns a value equal to 1 if the value  $r_i$  exactly matches its closest value in  $D(r_i)$ , else it returns a value between 0 and 1. The Normalized Edit Distance (NED) [11] considers the minimum number of character insertions, deletions and replacements needed to convert a value  $r_i$  to a value in  $D(r_i)$ . This edit distance is also referred to as the Levenshtein distance [35]. Equation 2.2 is suitable for measuring the accuracy of data type "character", such as name, address, and city.

In addition, Vaziri et al. [40] propose another equation for the  $acc(r_i, D(r_i))$  based on mathematical difference distance

$$acc(r_i, D(r_i)) = \begin{cases} 1, & \text{if } r_i \in D(r_i), \\ 1 - \frac{|r_i - D(r_i)|}{Max(r_i, D(r_i))} & \text{otherwise.} \end{cases} \quad (2.3)$$

This equation is suitable to measure the accuracy of numerical values data type. To address our needs, we modify Equation 2.1 and utilize Equations 2.2, 2.3 to define column accuracy

## 2. Theoretical Background

$$\begin{aligned}
 CA_i &= Acc(a) \\
 &= \frac{\sum_{j=1}^a acc(r_j, D(r_j))}{|a|},
 \end{aligned}
 \tag{2.4}$$

whereby  $CA_i$  is the overall syntatic accuracy of the  $i$ th column in a relation,  $r_j$  is the  $j$ th value of column  $a$  and  $|a|$  is the number of rows in the column.

### 2.2.2. Completeness

Completeness can be defined as the degree to which data are of sufficient breadth, depth and scope for the task at hand [44]. There are many perspectives on the dimension of completeness, leading to different metrics [30]. **Schema completeness** is defined as the degree to which objects and attributes are not missing from the schema at the most abstract level. An example is a delivery schedule database schema should contain fields like address, postal code, delivery date, etc. **Column completeness** is defined as a function of missing values for a column in a table. For instance, there are missing entries of last names in a name list. **Population completeness** measures missing values with respect to a reference population. For example, population completeness in a customer database means that all customers are represented and no entities are missing.

Batini and Scannapieca [5] mention that a more precise characterization of completeness is needed if we focus on a specific data model. This particular model is called completeness of relational data and it is based on column completeness. Completeness in this model can be determined by the presence or absence of null values. Null values in this model have three different definitions.

Let us take a *Person* relation as shown in Table 2.2 as an example to show these distinct types of null values.

Person ID	Name	Gender	Birth Date	Email
1	Danny	Male	01.01.1990	danny90@gmail.com
2	Emily	Female	10.02.1993	NULL
3	Frank	Male	15.03.1995	NULL
4	Gabrielle	Female	20.05.1992	NULL

**Table 2.2.:** Example of a *Person* relation with different definitions of null value for the attribute *Email*



## 2.2. Data Quality and Weighted Metric

For rows with "Person ID" equal to 2, 3 and 4, the "Email" value is "NULL". The first type of null value is a real empty value. "Emily" does not have an email, so no incompleteness occurs. The second type of null value is a missing value. "Frank" has an email, but its value is unknown. This contributes to an incompleteness. The third type of null value is an ambiguous empty value. "Gabrielle" may or may not have an email. In this case, incompleteness may not be the case.

This thesis only focuses on the second type of null value for simplification. A closed-world assumption according to Batini and Scannapieca [5] will be made, which states that only the values that are actually present in a relation will be considered and no other values represent facts of the real world.

Lee et al. [22] define that completeness can be measured using simple ratio

$$\text{Completeness rating} = 1 - \left( \frac{\text{Number of incomplete items}}{\text{Total number of items}} \right). \quad (2.5)$$

To fit our needs, we utilize Equation 2.5 to define column completeness

$$CC_i = 1 - \left( \frac{\text{Number of incomplete items in the column}}{\text{Total number of items in the column}} \right), \quad (2.6)$$

whereby  $CC_i$  is the completeness rating of the  $i$ th column in a relation.

### 2.2.3. Consistency

Consistency is the degree to which data has attributes that are non-conflicting and are coherent with other data in specific use cases [36]. This dimension can also be viewed from several perspectives [22]. The first type is the **consistency of integrity constraints**, which tracks the violation of logical rules applied to the data. For example, an employee's hire date must not be earlier than their birth date. The second type is the **consistency between two related data elements**. For instance, the postal code and the city's name must be consistent, meaning both should correspond correctly to each other in reality. The third type is the **consistency of format** for the same data element used in different tables. If a mobile phone number includes the country code in one table, it should follow the same format in all other tables where the phone number is used.

According to Batini and Scannapieca [5], the consistency of integrity constraints can be further divided into two categories, namely intrarelation constraints and

## 2. Theoretical Background

interrelation constraints. Let us take an *Employee* relation in Table 2.3 and a *Promotion* relation in Table 2.4 as an example. Intrarelation integrity constraints, also known as domain constraints, consider single or multiple attributes within a relation. An intrarelation integrity constraint defined in Table 2.3 states that an employee must have a minimum age of 18. For the row represented by "Jack", the "Age" value is less than 18. This results in an inconsistency that violates an intrarelation integrity constraint.

Interrelation integrity constraints take into account the attributes of more than one relation. An example of such constraint in Table 2.3 and Table 2.4 is that the promotion year of an employee must not be earlier than the start year. Employee "Harry" has a "Promotion Year" of "2022" and a "Start Year" of "2024". This is invalid and it is a violation of interrelation integrity constraint.

In addition to the inconsistency of integrity constraints, there is also an inconsistency of format in Table 2.4. The tuple with "Promotion ID" of "323" has the value "in 5 years" for attribute "Promotion Year". This is a violation of the third type of consistency as only numerical values are expected.

In our use case, we apply the first and the third types of consistency with slight modifications. Specifically, we focus on intrarelation integrity constraints and format consistency for data within the same relation to define consistency. The second type of consistency will not be considered to simplify the process of evaluating data consistency.

Employee ID	Name	Age	Start Year	Position
100	Harry	22	2024	Reliability Engineer
101	Ivy	25	2023	Sales Representative
102	Jack	15	2022	IT Specialist

**Table 2.3.:** Example of an *Employee* relation

Promotion ID	Employee ID	Promotion Year	New Position
321	100	2022	Senior Reliability Engineer
322	101	2026	Sales Manager
323	102	in 5 years	Senior IT Specialist

**Table 2.4.:** Example of a *Promotion* relation for employees in Table 2.3

To measure consistency, Lee et al. [22] propose that

$$\text{Consistency rating} = 1 - \left( \frac{\text{Number of inconsistent units}}{\text{Total number of consistency checks performed}} \right). \quad (2.7)$$

In addition, to ensure our needs are met, we utilize Equation 2.7 to define column consistency

$$CCS_i = 1 - \left( \frac{\text{Number of inconsistent units in the column}}{\text{Total number of consistency checks performed in the column}} \right), \quad (2.8)$$

whereby  $CCS_i$  is the consistency score of the  $i$ th column in a relation.

#### 2.2.4. Weighted Metric

Simple ratios are easy to implement for measuring data quality, but they do not account for the varying weights of different data [40]. In a company, some data may be more important than others in helping the company achieve its business goals. Elouataoui et al. [12] also share this opinion, stating that this is true in most organizations. Therefore, these relevant data must receive more attention in order to be more significant.

#### Completeness

Some attributes may be more important than others for a tuple in structured data. Let us take a *Student* relation as shown in Table 2.5 along with column completeness as an example. At first glance, we can assess the overall completeness of this relation by averaging five completeness values, resulting in 78%. However, if each column has a specific weight, it can result in a more practical completeness value for this relation.

Column	Student ID	Name	Telephone	Email	Course
Completeness Score	90%	80%	70%	60%	90%

Table 2.5.: Example of a *Student* relation with their completeness score

## 2. Theoretical Background

Let us now consider the *Student* relation with weight values as shown in Table 2.6. "Student ID" carries the most weight, as a tuple representing a student is useless without this value. "Email" carries more weight than "Telephone" because students are generally contacted via email instead of phone calls for organizational purposes. The total weights should also add up to one to keep the final completeness value normalized.

Column	Student ID	Name	Telephone	Email	Course
Weight	0.30	0.25	0.10	0.20	0.15

**Table 2.6.:** Example of a *Student* relation with their weight values

Vaziri et al. [40] proposes to calculate weighted column completeness

$$\sum_{i=1}^n (CW_i \times CC_i), \quad (2.9)$$

whereby  $CW_i$  is the column weight of the  $i$ th column and  $CC_i$  is the column completeness of the  $i$ th column by simple ratio. The column completeness,  $CC_i$ , can be calculated using Equation 2.6. This leads to a weighted completeness value of 79.5%, which is more practical and realistic to use as compared to the average completeness value of 78%. This calculation method can also be utilized for other data units and dimensions [40]. The example above illustrates column completeness and we can calculate tuple completeness in a similar manner. In the *Student* relation, some students are more active than others, making it more important to have complete tuples for active students than for inactive ones in certain contexts. However, this thesis focuses solely on column completeness.

### Accuracy

Applying the same strategy from weighted column completeness, we can define weighted column accuracy

$$\sum_{i=1}^n (CW_i \times CA_i), \quad (2.10)$$

whereby  $CW_i$  is the column weight of the  $i$ th column and  $CA_i$  is the column accuracy of the  $i$ th column. We can utilize Equation 2.4 to calculate the column accuracy,  $CA_i$ .

### Consistency

Similarly, we can also define weighted column consistency

$$\sum_{i=1}^n (CW_i \times CCS_i), \quad (2.11)$$

whereby  $CW_i$  is the column weight of the  $i$ th column and  $CCS_i$  is the column consistency of the  $i$ th column. This column consistency,  $CCS_i$ , can also be calculated using Equation 2.8.

## 2.3. Usability Evaluation

Systems designed for people to use, like front ends, should be easy to learn, easy to remember and highly useful [15]. This is essential to enable users to complete tasks intuitively and efficiently. Grudin [16] points out that a potentially useful system could be unusable if users find it too difficult to learn and interact with, rendering the system's functions useless.

Evaluating the usability of a system is crucial to enable an iterative evaluation process that ensures the interaction design of software with high usability [17]. Many researchers have proposed different methods to evaluate usability, but there is a lack of a standardized set of metrics to compare these methods [26]. Consequently, only certain evaluation techniques are suitable for specific applications. This thesis follows the guidelines provided in the book "User Interface Design and Evaluation" written by Stone et al. [37]. We combine various evaluation methods like user testing and questionnaires to fit our needs in conducting a formative evaluation of our system.

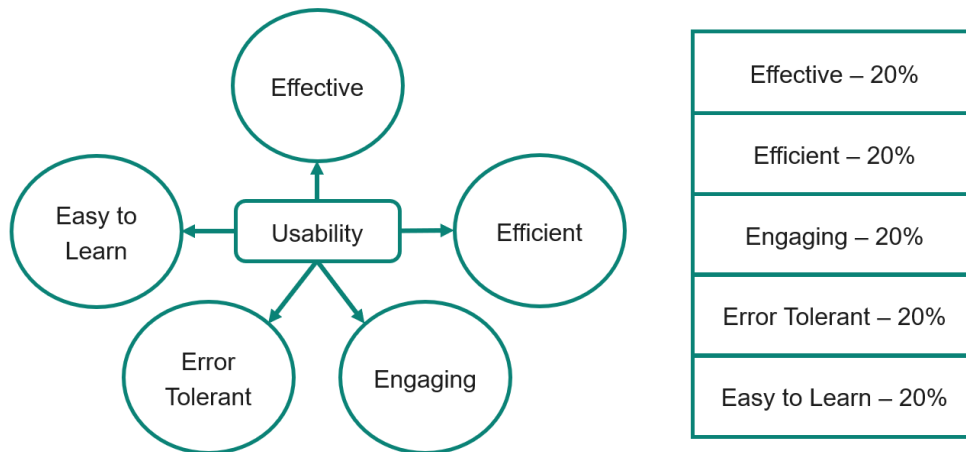
### 2.3.1. Evaluation Setup

#### What to Evaluate

To plan a suitable strategy for usability evaluation, we must first understand what usability means and what should be evaluated. Nielsen [27] defines usability as the measure of how well a user can utilize a system's functionalities. He further explains that usability traditionally comprises five key attributes: learnability, efficiency, memorability, errors and satisfaction. Based on this definition, Quesenbery [31] has come up with *5Es* as shown in Figure 2.4 to describe the dimensions of usability: effective, efficient, engaging, error tolerant and easy to learn. The *5Es*

## 2. Theoretical Background

are easy to remember and have straightforward definitions. **Effective** refers to how accurately and completely users achieve their goals. **Efficient** is how quickly and accurately a particular task can be done. **Engaging** represents user satisfaction with the system. **Error Tolerant** describes how well the system prevents errors and helps users recover from them. **Easy to Learn** refers to the system's ability to support and guide users during initial orientation and subsequent usage. We can prioritize one usability dimension over the other depending on the specific use case.



**Figure 2.4:** Dimensions of usability in balance, adapted from [31]

### Choosing Participants

For systems or tools that the user is expected to use without outside support, each evaluation session should only involve one participant interacting with the system alone to simulate the real-life environment. Such evaluation sessions can be repeated with several different participants to get different feedback and views. It is common for there to be only about five participants in the initial round of evaluation [37]. Usability experts mention that they often learn a lot, even from just a few participants. Virzi [41] shares the same opinion and emphasizes that 80% of important usability problems are discovered just after five subjects.

Choosing suitable participants plays a key role in generating meaningful results for usability evaluations. In an ideal situation, each participant should be a real user who will use the system. Depending on the situation, a representative user or a usability expert may also provide valuable insights. Let us take the evaluation of a public information kiosk for tourist information as an example. The actual users of the system include the general public, such as tourists who may not speak English.

Other representative users like participants who speak English, can be included in the early stages of evaluation to simplify communication and help identify obvious errors. This makes it easier to refine the prototype before testing it with actual users who do not speak English in subsequent evaluation rounds.

### Preparing Task Descriptions

Task descriptions provide instructions that represent tasks which users will perform while interacting with the prototype during the evaluation [37]. These tasks should represent key functions of the system to focus on the most important user interaction and to cover different complexity levels of user testing. After defining the task descriptions, it is important to determine the sequence in which tasks will be presented during the evaluation. This is because it may not be feasible to ask participants to perform all the tasks as planned in a limited amount of time. Thus, certain tasks should be prioritized to ensure that critical aspects of the systems are evaluated. Hix and Harton [19] suggest a list of different type of tasks that can be used:

- Essential tasks that users perform frequently.
- Tasks that are highly significant to the user or the business.
- Tasks that involve newly developed design features.
- Critical tasks that are rarely used but important to be evaluated.

### 2.3.2. Data Collection

The type of data being collected is highly relevant in evaluating a system's usability. There are two types of data: **quantitative data** and **qualitative data**. Quantitative data are numeric data that can be obtained from measurements, whereas qualitative data includes any information that is not numeric.

Table 2.7 below shows an example of quantitative and qualitative data based on the usability dimensions, or the *5Es*. This thesis primarily collects quantitative data. Any qualitative data obtained through questionnaires will be converted into quantitative data.

### Timing and Logging Actions

To validate time-related usability metrics quantitatively, it is necessary to measure the time taken to complete a task. A digital or analog stopwatch is more accurate and suitable as compared to a clock in this case. The facilitator of the user test

## 2. Theoretical Background

Dimension	Quantitative Data	Qualitative Data
Effective	If task is completed completely and accurately	User's view if task is finished correctly or not
Efficient	Keeping track of mouse clicks or keystrokes needed to finish a task	User's view on the difficulty to complete a task
Engaging	Numeric measurement of user satisfaction	User satisfaction check through questionnaires or surveys
Error tolerant	Number of entries or tasks with incorrect data	User's feedback of confidence in using the interface even if they make mistakes
Easy to learn	Time spent for a novice and an experienced user to complete a task	Novice user's report on experience in using the interface

**Table 2.7.:** Example of possible data collection according to usability dimensions

can pause the stopwatch in case of any disruptions, whether from the participant or externally. However, it may be easy for the facilitator to forget to restart the stopwatch again if they are also the timekeeper.

Recording or logging the actions carried out by participants when completing an assigned task can be useful for evaluation. Specialized software can keep track of the number of mouse clicks or keystrokes, which is more advantageous than manually recording it on a piece of paper. This also helps avoid common human errors such as miscounting clicks.

### Questionnaires

Using questionnaires to collect data for an evaluation has its pros and cons. Questionnaires provide the same format for all participants, which facilitates consistent data collection. Thus, it is possible to compare results between participants after conducting such questionnaires. However, it can be difficult and challenging to design a suitable and useful questionnaire as different evaluations require different metrics. There are a few designed questionnaires available to be used as a part of usability evaluation. Examples are *System Usability Scale (SUS)* [8],



### 2.3. Usability Evaluation

*Questionnaire for User Interface Satisfaction (QUIS)* [10] and *Computer System Usability Questionnaire (CSUQ)* [23].

SUS is a *Likert* scale as shown in Figure A.1 in the appendix, where participants respond to a statement indicating the degree of agreement or disagreement on a 5-point scale. As a result, SUS generates a single number representing a composite measure of the overall usability of the system being evaluated. The inventor of SUS also mentions that scores for individual items are not meaningful on their own.

Tullis and Stetson [39] state that the SUS provides more reliable results than QUIS, CSUQ and two other questionnaires. SUS is the easiest to deploy because it consists of only ten questions compared to QUIS, with 27 questions and CSUQ, with 19 questions. Lewis [24] also recommends to use shorter questionnaires to maximize the response rate. Therefore, this thesis utilizes the structure of SUS as the basis for our questionnaire and modifies it as needed to meet our usability requirements.



# 3. Implementation

In this chapter, we explore the prototype development of an in-house software application and prepare suitable data for the formative evaluation of the system. Generating this data before and after the implementation of prototype is a crucial step for the evaluation in the next chapter, as it is the foundation of the overall assessment of data quality.

## 3.1. Prototype

Before the start of this prototype development, the reliability department conducted interviews among experienced engineers. They identified a need for a centralized and user-friendly platform to track physical samples. This in-house software application is called *NEXTREL*, which is an acronym for the term *Next Reliability*. The *NEXTREL* tool is hosted on the company server, where employees can access it via the browser. Figure 3.1 shows the user interface of the current prototype of the in-house software application.

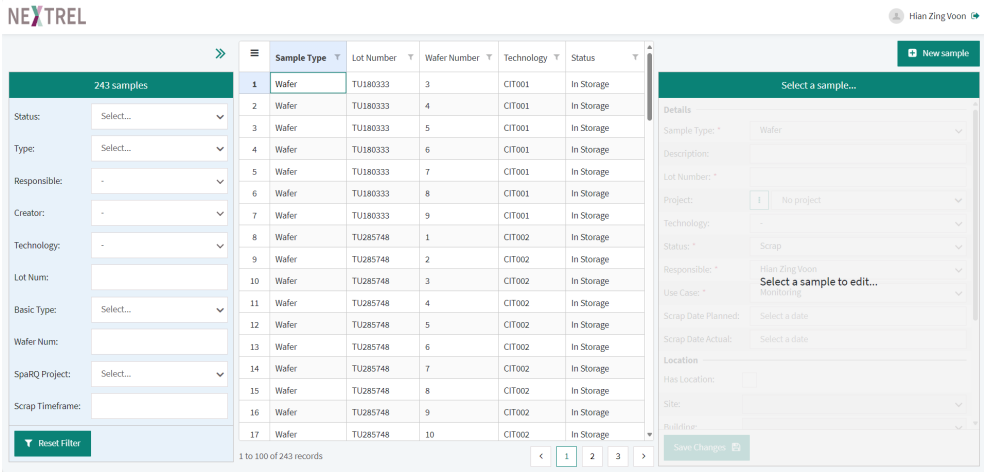


Figure 3.1: User interface of the prototype

The front end of this software is developed using a low-code platform called OutSystems. It allows rapid prototyping and iteration within its environment, making

### 3. Implementation

it well-suited to the agile development process employed within the development team. This results in a faster and more efficient approach to implement necessary changes compared to traditional software development [2]. This is great for prototype development that requires feedback from either users or the project owner for improvements at every iteration. On the other hand, the back end consists of an Oracle database system to create a robust data model to handle complex relationships between different data points. Consequently, this ensures data integrity and allows complex queries to be executed and displayed at the front end.

The user interface is a layout divided into three columns: one for filtering data, one for displaying data, and one for editing data. The sidebar on the left in Figure 3.1 consists of multiple dropdown menus, allowing users to filter and view samples. Based on specific criteria, users can narrow down and refine the contents to look for a particular sample. This is necessary to allow users to quickly search the database and obtain needed information.

In the middle of the interface is a tabular structure, which is called a data grid in the OutSystems platform. This data grid displays filtered or unfiltered data up to 100 records per page. Users can scroll down in the data grid or cycle to another page to look at the following records. Additionally, users can customize the layout of the columns according to preferences as seen in Figure 3.2. They can hide or re-arrange columns in this data grid as needed.

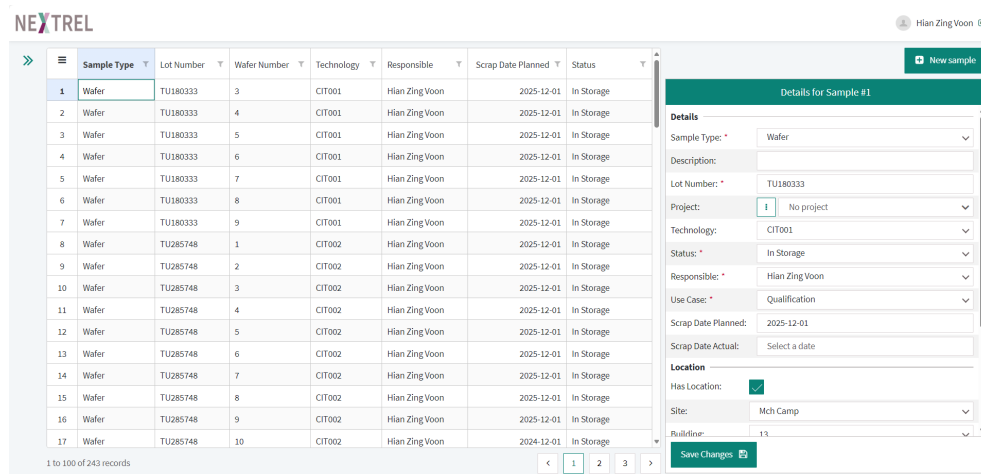


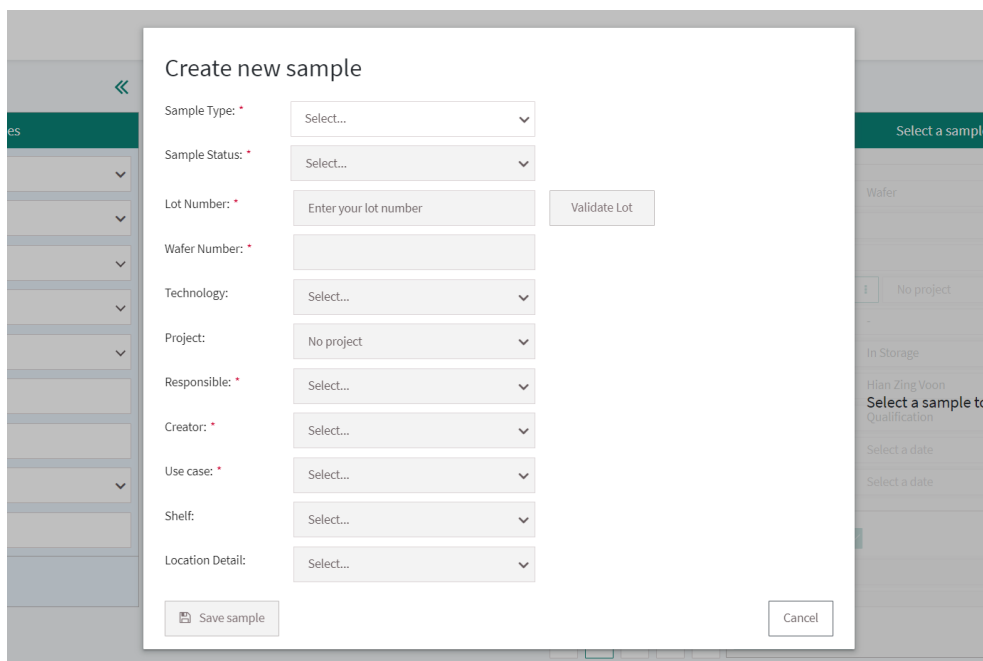
Figure 3.2: Customized layout of data grid along with a column for data edits

On the right side in Figure 3.2 is a dedicated column for editing data. Users can select any record in the data grid and make necessary edits. This section of the interface is initially hidden when users first log in, as shown in Figure 3.1. This serves as a prompt to remind users to select a sample in the data grid and edit it

### 3.1. Prototype

if needed. For instance, users are able to edit details such as "Status", depending on the current state of the physical sample, whether it is in storage or scrapped. To apply these changes, users must click on the "Save Changes" button at the bottom. Otherwise, the changes will revert back to their original form.

Figure 3.3 shows a pop-up to create a new sample in the system, which is triggered when users click on the "New sample" button located above the column for data edits. Initially, most input fields are disabled. Once users complete the required input fields starting with the "Sample Type" dropdown menu, other subsequent fields will be enabled. Most input fields are presented as dropdown menus for users to select from a set range of options. This is done to ensure that users provide accurate inputs, which helps to prevent common human errors like typographical mistakes. Such errors tend to occur when users can enter any input in free text fields. Next to the "Lot Number" input field is a "Validate Lot" button. It is crucial as users are allowed to type any input freely. This button ensures the correct input in the field by allowing users to validate the lot number to be entered.

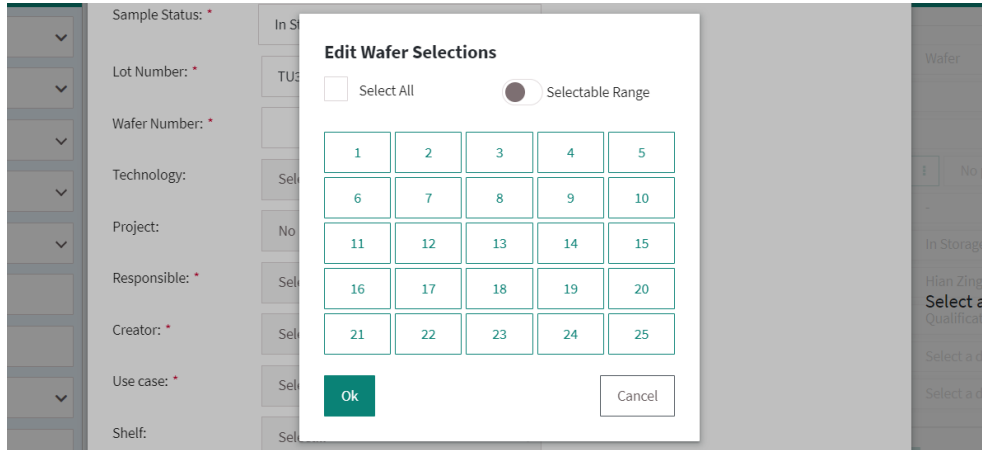
The image shows a 'Create new sample' pop-up window. It has a title bar with a back arrow and the text 'Create new sample'. The form contains the following fields: 'Sample Type' (dropdown), 'Sample Status' (dropdown), 'Lot Number' (text input with a 'Validate Lot' button), 'Wafer Number' (text input), 'Technology' (dropdown), 'Project' (dropdown with 'No project' selected), 'Responsible' (dropdown), 'Creator' (dropdown), 'Use case' (dropdown), 'Shelf' (dropdown), and 'Location Detail' (dropdown). At the bottom left is a 'Save sample' button with a floppy disk icon, and at the bottom right is a 'Cancel' button. The background shows a blurred interface with a table and a 'Select a sample' dropdown.

**Figure 3.3:** Pop-up to create a new sample

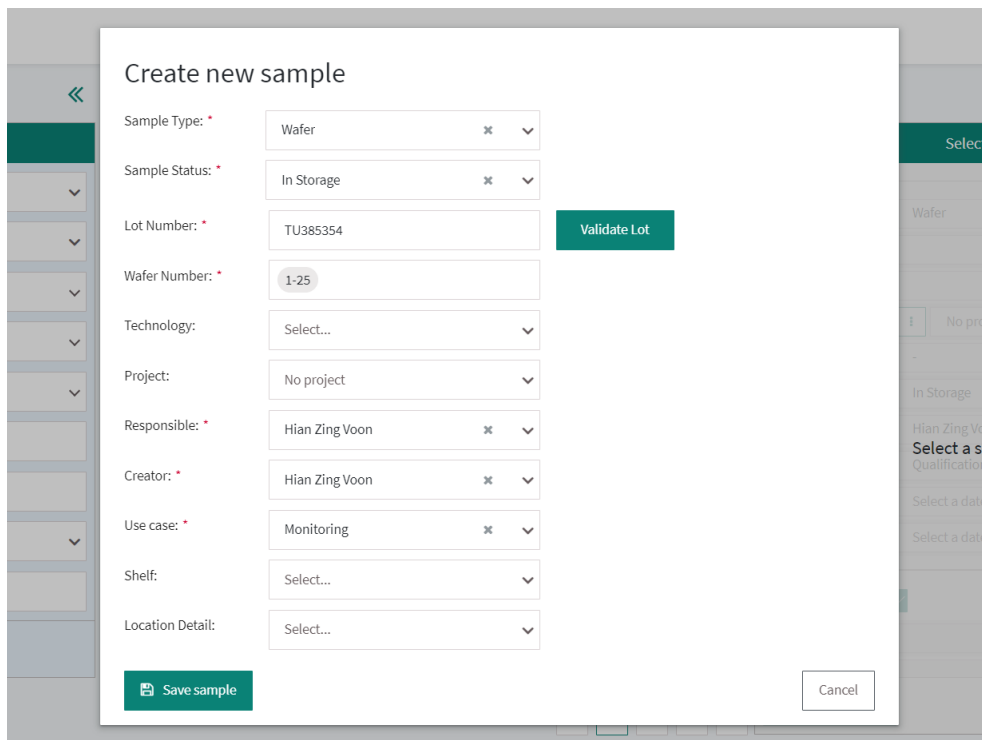
Figure 3.4 shows a custom widget for the "Wafer Number" input field, which appears as a pop-up. This allows users to select multiple wafers at a time, which was deemed more intuitive than a multiple select dropdown menu. Users are also able to select a particular range of wafers if needed.

### 3. Implementation

As seen in Figure 3.5, users may leave optional input fields empty if they are uncertain about the information. However, the "Save sample" button stays disabled until mandatory fields marked with red asterisks are completed. This ensures that users fill in the necessary information before creating a new sample.



**Figure 3.4:** Multiple selection of wafers during creation of a new sample



**Figure 3.5:** Creation of a new sample with mandatory fields completed

## 3.2. Preparation for Data Quality Evaluation

To conduct an evaluation of data quality and compare the current data management tool with the *NEXTREL* prototype, we need sufficient and appropriate data. Currently, engineers primarily use *Microsoft Excel* as the data management tool in the department to handle test samples. *Microsoft Excel* is a spreadsheet software commonly used by many users to organize and store data in the format of rows and columns. This section outlines the preparation needed to conduct data quality evaluation, including the development of a Python script to assess data quality dimensions such as accuracy, completeness and consistency.

### Selection of Status Quo Data

Currently, there exist several *Microsoft Excel* files containing relevant information about samples in the department. These files are stored in network drives, which the engineers can access. We select the most recent file containing thousands of samples and extract the most relevant data. Only data with 100 distinct random lot numbers and data with scrap dates beyond October 2024 are considered. This ensures a fair comparison between the status quo data and the data after the prototype implementation. We also include data without scrap dates to account for human typographical errors, which we want to include in the evaluation. 100 distinct random lot numbers are chosen to represent a fraction of samples that the department receives annually. In the appendix, Figure A.2 shows a section of the status quo data with certain details altered for confidentiality purposes.

### Selection of Data after Prototype Implementation

To assess the data quality generated from the prototype, we must first input the raw data into the system using the interface. Specifically, we use samples with the same 100 lot numbers used previously to generate the status quo data. However, some lot numbers do not yet exist in the system with the current prototype. This could be due to actual typographical errors in the original lot number entries. Thus, we cannot create new samples for them. Despite this limitation, we still include this data in our evaluation to simulate real-world scenarios where some data cannot be entered or are non-existent. A section of this data with modified details can be seen in Figure A.3 in the appendix.

### Python Script Development

In order to objectively measure and assess the data quality dimensions using equations defined in the previous chapter, we develop a Python script with various li-

### 3. Implementation

braries. Some libraries used are *pandas*<sup>1</sup>, *Levenshtein*<sup>2</sup>, *seaborn*<sup>3</sup> and *matplotlib*<sup>4</sup>. The *pandas* library is primarily used to organize data into data structures constructed with rows and columns. This allows the evaluation of previously generated data that is in the same format. We use the *Levenshtein* library to calculate the NED of data, which is essential in evaluating the accuracy of data quality. To visualize the evaluation results, *seaborn* and *matplotlib* are used to create graphs such as box plots and violin plots, which will be presented in the next chapter.

---

<sup>1</sup><https://pypi.org/project/pandas/2.2.2/> [Accessed: Jul. 10, 2024]

<sup>2</sup><https://pypi.org/project/Levenshtein/0.26.0/> [Accessed: Jul. 10, 2024]

<sup>3</sup><https://pypi.org/project/seaborn/0.13.2/> [Accessed: Jul. 10, 2024]

<sup>4</sup><https://pypi.org/project/matplotlib/3.9.2/> [Accessed: Jul. 10, 2024]



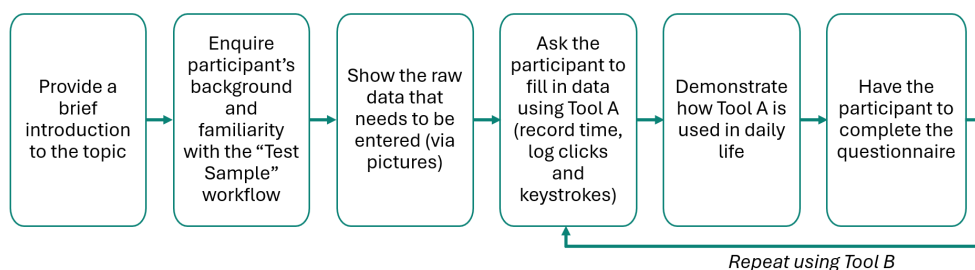
## 4. Evaluation

This chapter will dive deeper into evaluating data quality and system usability. Prior to this, an appropriate study setting must be established to ensure good formative evaluation. For simplicity when referring to tools, this thesis will use *Microsoft Excel* and *Excel* interchangeably in the following chapters.

### 4.1. Study Setting

#### Study Workflow

Before conducting the study, a clear task description is required to evaluate data quality and system usability. Participants first receive a short introduction to this topic and related information for better context. Their background and experience level with the "Test Sample" workflow are assessed before assigning them a task related to the workflow. The task assigned is to create a new sample with the same data in both Tool A and Tool B. Tool A is *Microsoft Excel* and Tool B is *NEXTREL*. Some participants start with Tool A and then use Tool B, while others begin with Tool B before moving on to Tool A. This randomized order inspired by A/B testing helps to reduce the impact of learning effects, which occur when retaking tests [6, 32]. After using either tool, participants are shown data generated by that tool in real-life scenarios before completing a questionnaire consisting of ten questions. Then, this process is repeated using the other tool. Figure 4.1 illustrates an overview of the study workflow.

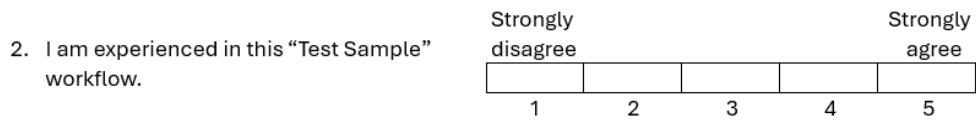


**Figure 4.1:** Study workflow to evaluate data quality and system usability

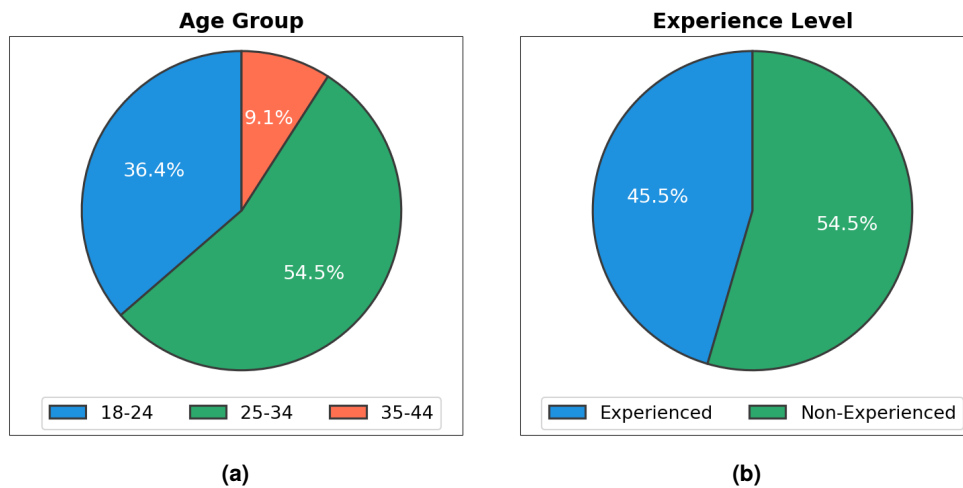
#### 4. Evaluation

##### Background of Participants

We gathered 11 participants to evaluate the system's usability. Out of 11 participants, four are aged 18 to 24, five are aged 25 to 34 and only one is aged 35 to 44. To enquire about a participant's experience level in the "Test Sample" workflow, a direct question is presented to them as shown in Figure 4.2. None of the participants answered "Strongly agree" about their degree of experience. A reason could be that participants are slightly more reserved and do not want to be overconfident. Therefore, for this cohort of participants, we consider those who answered "2" or higher on the scale as experienced individuals. In this case, five participants have experience in the "Test Sample" workflow, while six do not. The age group and experience level data are presented in pie charts in Figure 4.3.



**Figure 4.2:** Question to determine participant's experience level



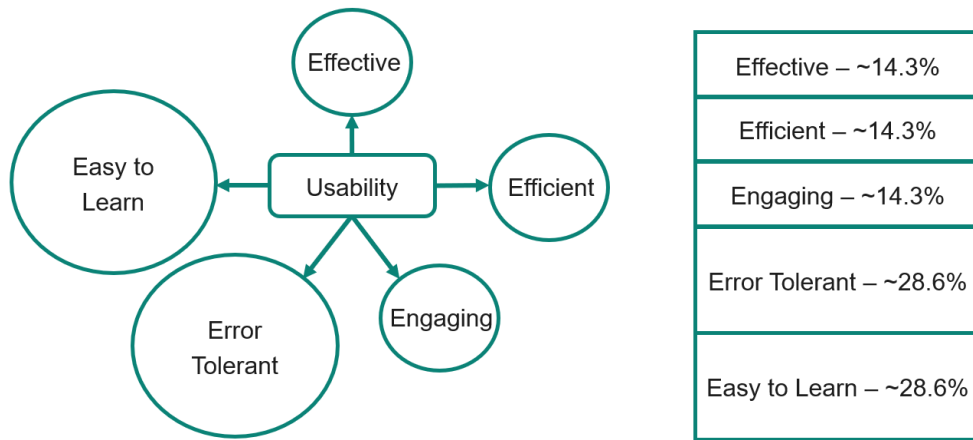
**Figure 4.3:** Participants' background of: (a) age group and (b) experience level in the "Test Sample" workflow

##### Questionnaire

The questionnaire follows the structure of SUS and the usability dimensions described in Chapter Two. Questions one to seven focus on evaluating usability, specifically the "Error Tolerant" and "Easy to Learn" dimensions, as shown in Figure 4.4. Two questions represent each of these two dimensions, which is about

#### 4.1. Study Setting

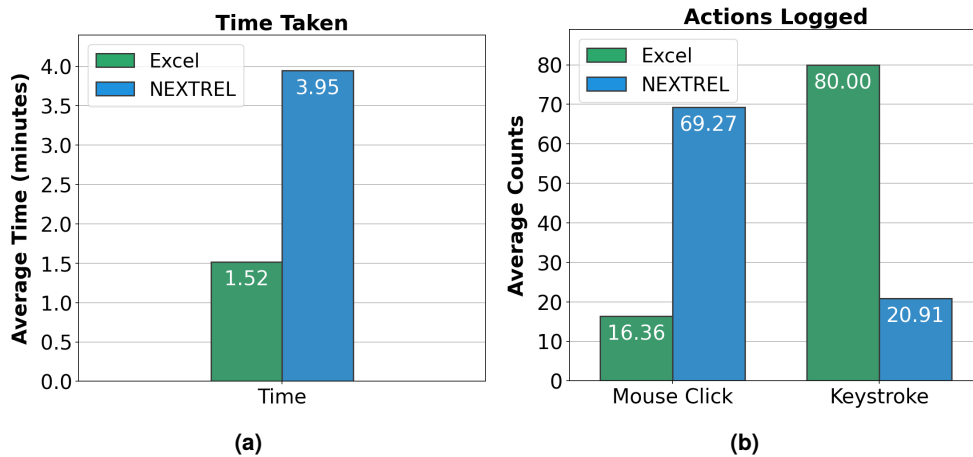
28.6% of the total questions. Whereas questions eight to ten address subjective data quality perceived by users. The set of questions can be seen in Figure 4.6.



**Figure 4.4:** Dimensions of usability with a focus on "Easy to Learn" and "Error Tolerant", adapted from [31]

#### Timing and Logging Actions

On average, creating a new sample with *NEXTREL* takes longer than it does with *Microsoft Excel*. Additionally, using *NEXTREL* requires more mouse clicks but fewer keystrokes than using *Microsoft Excel*. The data is illustrated in Figure 4.5.



**Figure 4.5:** Recorded data of: (a) time taken and (b) actions logged of participants while creating a new sample

#### 4. Evaluation

1. I was satisfied with this tool overall.	Strongly disagree								Strongly agree
	1	2	3	4	5				
2. I found this tool unnecessarily complex.	Strongly disagree								Strongly agree
	1	2	3	4	5				
3. I found it easy to fill in the data.	Strongly disagree								Strongly agree
	1	2	3	4	5				
4. I think that I would need the support of a technical person to be able to use this tool.	Strongly disagree								Strongly agree
	1	2	3	4	5				
5. I had a clear understanding of where each data needs to be entered.	Strongly disagree								Strongly agree
	1	2	3	4	5				
6. I found it easy to understand the correct format for entering the data.	Strongly disagree								Strongly agree
	1	2	3	4	5				
7. I received support from the tool for entering the data.	Strongly disagree								Strongly agree
	1	2	3	4	5				
8. I found the information provided by the tool to be accurate.	Strongly disagree								Strongly agree
	1	2	3	4	5				
9. I found the available data fields to be complete.	Strongly disagree								Strongly agree
	1	2	3	4	5				
10. I found the data in the tool to be consistent.	Strongly disagree								Strongly agree
	1	2	3	4	5				

**Figure 4.6:** Questionnaire based on SUS [8] and usability dimensions [31]

#### 4.2. Data Quality

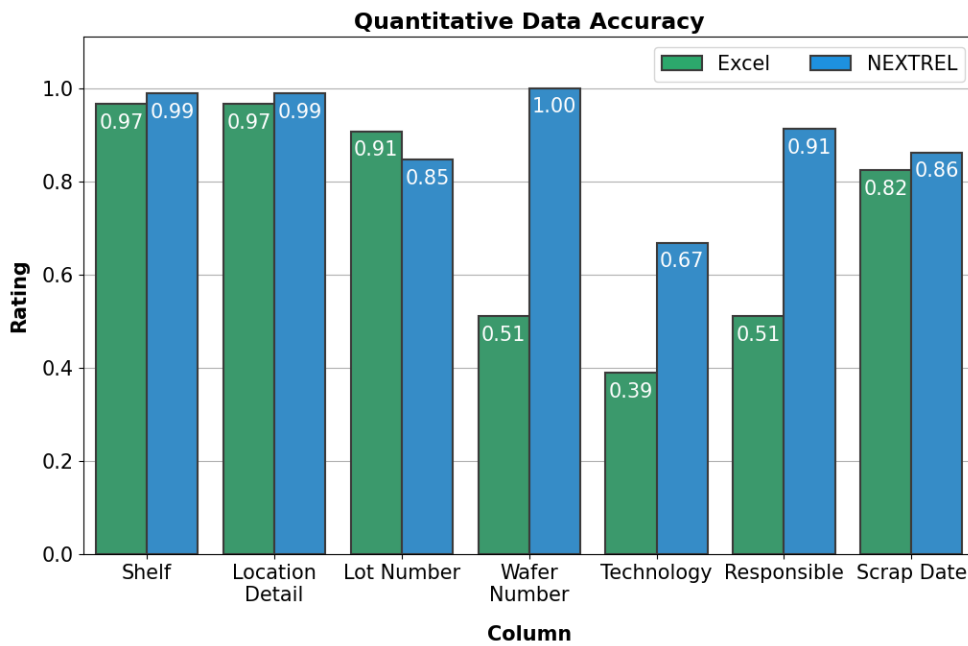
The evaluation of data quality consists of two parts: objective assessment and subjective assessment. Objective assessment focuses on quantitative metrics, whereas subjective assessment concentrates on users' perceived data quality.

### 4.2.1. Objective Assessment

By utilizing the selected data explained in Chapter Three, we can objectively evaluate the data quality generated by *Microsoft Excel* and *NEXTREL* through our custom Python script. To assess data quality across various dimensions, we evaluate seven specific columns: "Shelf", "Location Detail", "Lot Number", "Wafer Number", "Technology", "Responsible" and "Scrap Date". For each of these columns, we assign a data quality rating ranging from zero to one.

#### Accuracy

Data accuracy measures how closely the data in each tool matches the predefined correct values. As illustrated in Figure 4.7 and Table A.1 in the appendix, *NEXTREL* performs better than *Excel* in nearly all columns, with the exception of "Lot Number". This occurs because some lot numbers are not yet available in the system database, while others are mistakes in *Excel* that cannot be mapped. Consequently, this prevents users from creating new samples for those particular lot numbers. Significant differences are observed in columns like "Wafer Number", "Technology" and "Responsible". For example, *NEXTREL* scores 1.00 for "Wafer Number", whereas *Excel* achieves only 0.51. The poor performance of *Excel* in the "Technology" and "Responsible" columns indicates typographical errors.

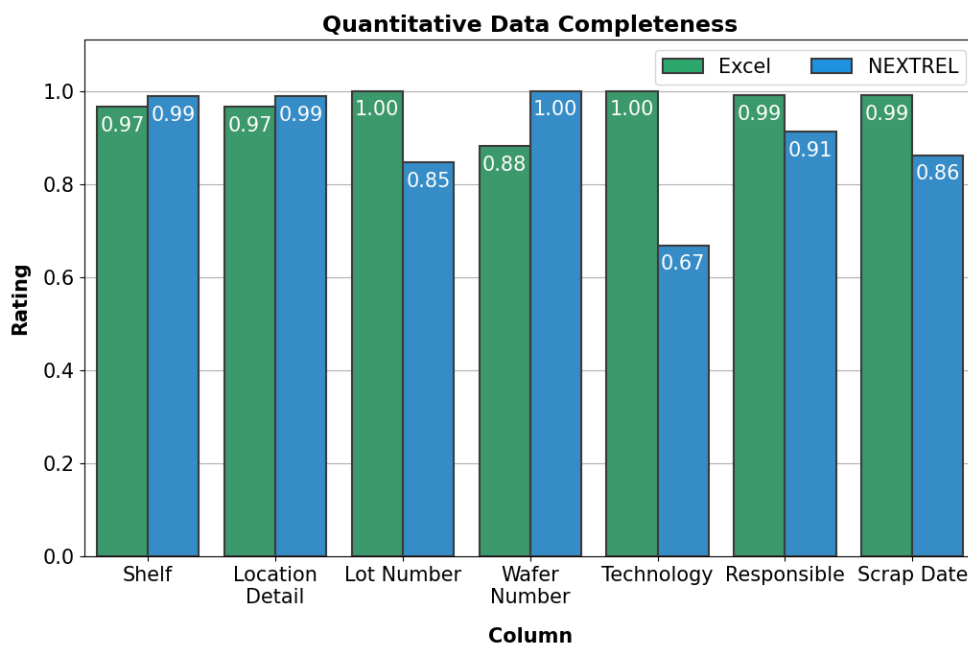


**Figure 4.7:** Comparison of quantitative data accuracy between Excel and NEXTREL

#### 4. Evaluation

##### Completeness

Data completeness evaluates missing values in the data. Figure 4.8 and Table A.2 in the appendix illustrate the ratings for completeness across both tools. *Excel* performs better than *NEXTREL* in 4 columns, which are "Lot Number", "Technology", "Responsible" and "Scrap Date". For *Excel*, most columns nearly achieve ratings of 1.00 except for "Wafer Number". This shows that data generated using *Excel* is mostly complete. The performance of *NEXTREL* is significantly worse than that of *Excel* in the "Technology" column. *NEXTREL* scores 0.67 while *Excel* scores 1.00. This occurs because certain selections of technology are not available in the system database, causing users to create a new sample with an empty field for "Technology" in *NEXTREL*. This data field is not set as mandatory in the prototype.



**Figure 4.8:** Comparison of quantitative data completeness between Excel and NEXTREL

##### Consistency

Data consistency reflects how well the data conforms to the predefined format or rules. Figure 4.9 and Table A.3 in the appendix show the comparison of data consistency between both tools. *NEXTREL* outperforms *Excel* across all columns, most significantly in the "Responsible" column. *NEXTREL* achieves a score of 0.91, while *Excel* scores only 0.23. The low performance of *Excel* in most columns is due to the use of free text data types, which allow users to input any data they

want. In the "Responsible" column, users often enter only the initials of individuals' names instead of their full names. Meanwhile in "Wafer Number", users frequently input multiple wafer numbers in a single entry, which does not fit the desired format.

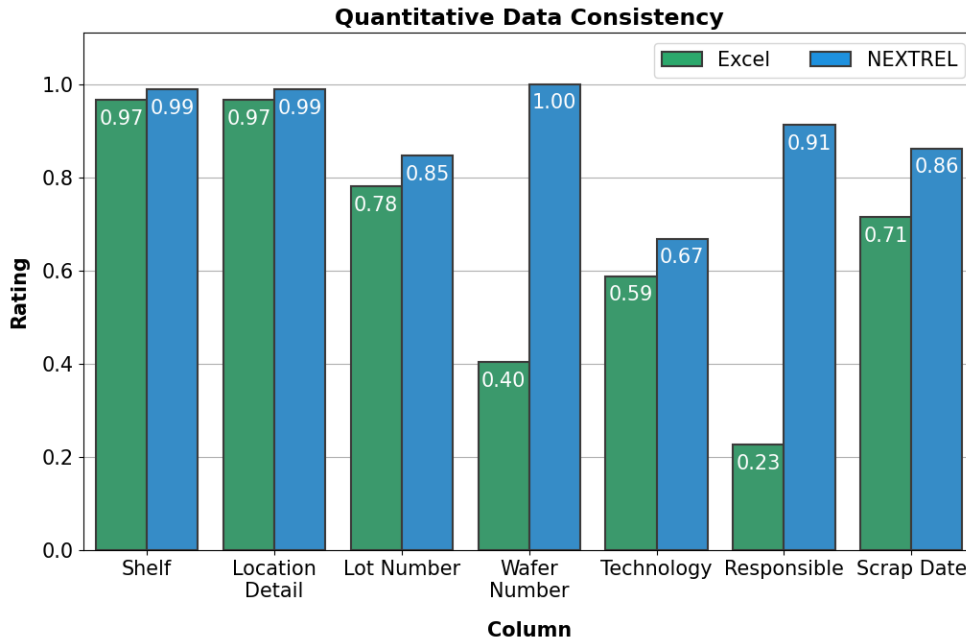


Figure 4.9: Comparison of quantitative data consistency between Excel and NEXTEL

### Weighted Metric

We can summarize the data quality results across all columns for each dimension to ease the comparison between both tools. Table 4.1 shows the weight values assigned to each column that we utilize to create a unified rating for each data quality dimension. The "Lot Number" column holds the highest weight value at 0.25, as it is the most critical data. Without an identifiable lot number, the information of a sample becomes meaningless. The "Wafer Number" and "Responsible" columns carry the second highest weight value of 0.20. Data from these columns is important to provide information regarding which particular wafer is referenced and indicate who is responsible for it. Conversely, "Shelf" and "Location Detail" hold the lowest weight value of 0.05, as the primary focus is on the other columns.

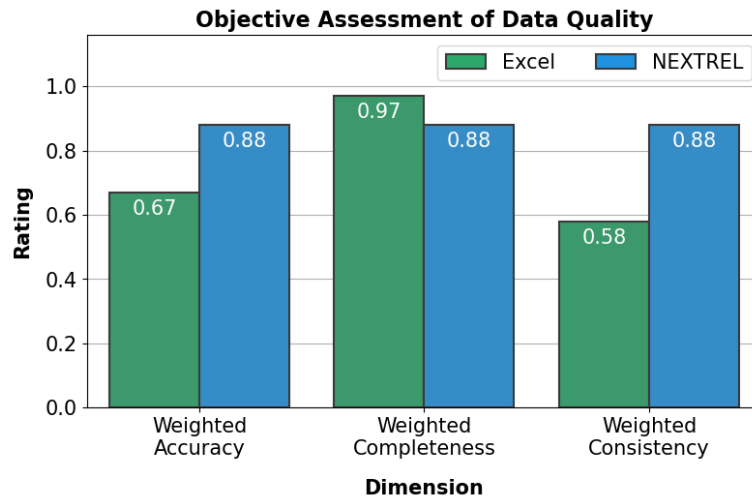
Using the defined weight values, we can apply the equations from Chapter Two to do an objective data quality assessment by calculating weighted metrics of data quality across all seven columns for each data quality dimension. Figure 4.10 shows the result for weighted accuracy, weighted completeness and weighted con-

#### 4. Evaluation

sistency. At a glance, *NEXTREL* performs better than *Excel* in both accuracy and consistency. For weighted accuracy, *NEXTREL* scores 0.88 while *Excel* scores 0.67. This indicates that *NEXTREL* provides more reliable data that users can depend on. In terms of weighted consistency, *Excel* scores even lower than its accuracy rating. This indicates that several data entries are in the wrong format, which causes data inconsistency that can disrupt workflow. However, where *Excel* outperforms *NEXTREL* is in data completeness. *Excel* achieves a score of 0.97 as compared to *NEXTREL*'s score of 0.88. This shows that *Excel* provides more complete data than *NEXTREL*, though at the expense of accuracy and consistency.

Column	Weight
Shelf	0.05
Location Detail	0.05
Lot Number	0.25
Wafer Number	0.20
Technology	0.15
Responsible	0.20
Scrap Date	0.10

**Table 4.1.:** Weight values of columns to be evaluated



**Figure 4.10:** Comparison of objective data quality across three dimensions using weighted metrics between *Excel* and *NEXTREL*

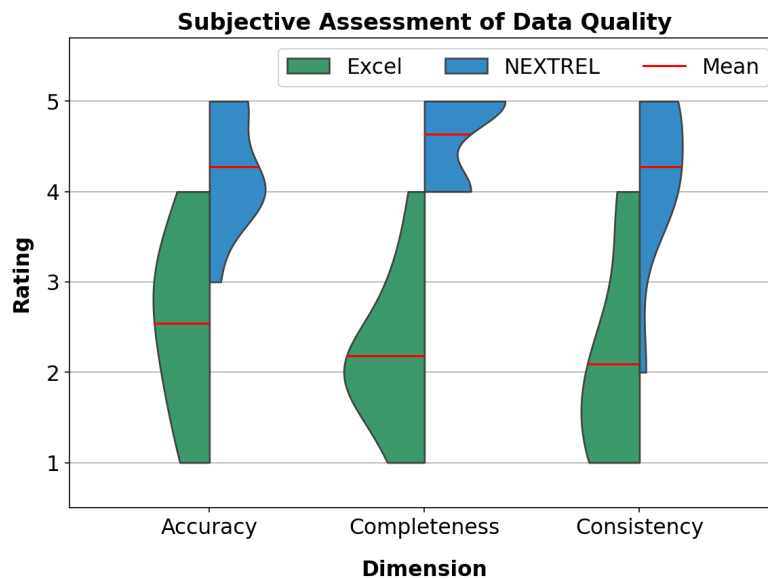


### 4.2.2. Subjective Assessment

A thorough evaluation of data quality involves both objective and subjective assessment. We can evaluate the subjective quality of data generated by both tools through a questionnaire designed for users who tested the prototype to create a new sample. Specifically, we utilize questions eight to ten in the questionnaire as shown in Figure 4.6 from the previous section. These questions aim to understand users' perceptions of the data quality generated by *Excel* and *NEXTREL*.

Figure 4.11 is a modified violin plot that shows the comparison of data quality subjectively between *Excel* and *NEXTREL* across all three dimensions: accuracy, completeness and consistency. We use a violin plot to show the data distribution, which can sometimes be skewed positively or negatively. Such skewed distributions are no longer normal distributions. Therefore, a box plot is not suitable for use in this situation. To provide an easier understanding, we modify the violin plot to show the mean instead of the median and the interquartile range.

The result of this violin plot is a scale from one to five, where one represents "Strongly disagree" and five is "Strongly agree". Users perceive that *NEXTREL* generates higher quality of data than *Excel* across all three dimensions assessed. *NEXTREL* consistently scores no lower than two, whereas *Excel* never scores higher than four. The most significant difference lies in data completeness, with *NEXTREL* scoring no less than four and *Excel* scoring a mean of around 2.



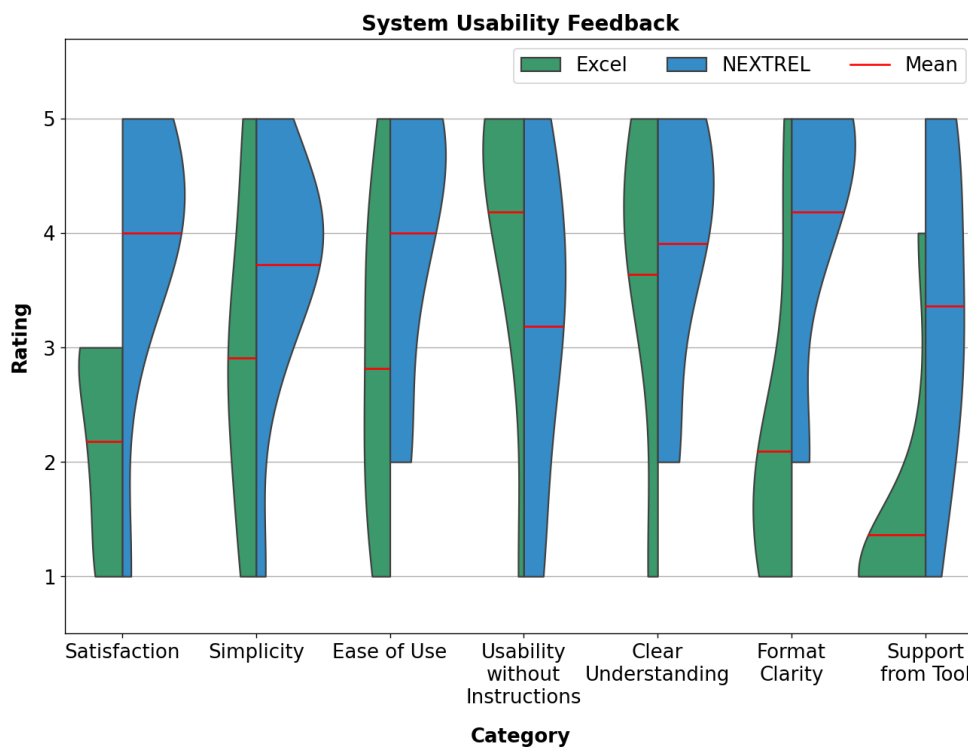
**Figure 4.11:** Comparison of subjective data quality across three dimensions between *Excel* and *NEXTREL*

## 4. Evaluation

### 4.3. System Usability

The usability evaluation of both tools can be conducted through user feedback, which is gathered using the same questionnaire shown previously in Figure 4.6. Questions one to seven focus on usability dimensions, providing insights into how users perceive the system's usability.

The violin plot illustrated in Figure 4.12 shows the comparison of system usability between *Excel* and *NEXTREL*. The questions are simplified and labeled by category on the x-axis, arranged in ascending order according to the questionnaire. The y-axis represents the results on a scale from one to five, where one indicates "Strongly disagree" and five corresponds to "Strongly agree". This is consistent with the questionnaire format.



**Figure 4.12:** Comparison of system usability feedback between *Excel* and *NEXTREL*

To maintain consistency in this plot, the labels of questions two and four have been altered. Previously, the label for question two was "Complexity" and the label for question four was "Usability With Instructions". After the alteration, these labels are now "Simplicity" and "Usability without Instructions" respectively. This adjustment ensures that a lower rating always indicates worse usability, while a higher rating

### 4.3. System Usability

signifies better usability. These two questions were previously worded in a negative way to encourage users to pay closer attention to the questionnaire rather than answering blindly [3].

Overall, *NEXTREL* performs better than *Excel* in many categories. For *NEXTREL*, the categories of "Satisfaction", "Simplicity", "Ease of Use", "Clear Understanding" and "Format Clarity" show a wide distribution at higher ratings with mean scores of around four. This is higher than the mean ratings for *Excel*. One can conclude that users are generally more satisfied with *NEXTREL*. They also find it easier to understand and use the tool as compared to *Excel*. Additionally, users understand what format of data they need to enter when using *NEXTREL*.

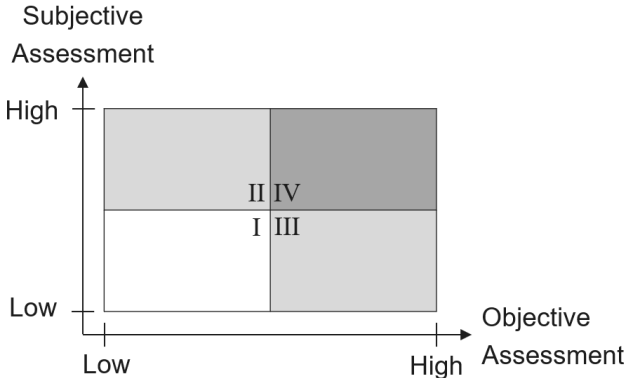
One of the most notable differences is the support provided by tools, which can be seen in the "Support from Tool" category. *NEXTREL* scores an average rating of around three, while *Excel* has ratings that are widely distributed around one. This indicates that users receive more support when using *NEXTREL* to create a new sample than they do with *Excel*. The higher rating for *NEXTREL* in this case may be attributed to the frequent use of dropdown menus, which provide options for users to choose from. However, *Excel* outperforms *NEXTREL* in "Usability without Instructions" category. Users find that they are able to navigate and use *Excel* without any instruction, while they feel they need technical support for *NEXTREL*.



# 5. Discussion

After evaluating data quality and system usability, we can summarize and discuss key findings. In this chapter, we will interpret the results obtained from both objective and subjective assessments of data quality and explore how these relate to the usability of both tools.

According to Pipino et al. [30], the goal of good data quality is for the data to excel in both objective and subjective assessment. As illustrated in Figure 5.1, achieving good data quality means that the outcome of the analysis should fall into *Quadrant IV*. The authors also point out that if the analysis falls into *Quadrants I, II or III*, companies should investigate the underlying causes and implement corrective measures.



**Figure 5.1:** Goal of objective and subjective assessment of data quality, adapted from [30]

From the results obtained after the comparison of both assessments in the previous chapter, we can analyse the data quality of *NEXTREL* and *Excel*. In terms of objective assessment, the average score for weighted metrics across all three dimensions for *Excel* is 0.74, which is lower than *NEXTREL*'s average of 0.88. *NEXTREL* scores consistently across all three dimensions, which is 0.88. This consistency is due to the missing of necessary information in the database that prevents the creation of new samples. However, the missing data are included in the analysis to enable a fair comparison as they are also part of *Excel*'s dataset. An example of missing information occurs when the person responsible for a sample is no longer

## 5. Discussion

part of department, making it impossible to create such samples in *NEXTREL*. This is because the data field for the responsible person is mandatory. Such entries still remain in *Excel*, as the responsible person was part of the department at the time when the data was entered into *Excel*. The way of selecting these data is explained in Chapter Three. For subjective assessment, *NEXTREL* achieves a high average rating above four, whereas *Excel* only has a low average rating of around two across all three dimensions. Based on the available comparisons, one could conclude that *NEXTREL* corresponds to "High" and *Excel* corresponds to "Low". *NEXTREL*'s analysis could then be placed in *Quadrant IV*, whereas *Excel*'s analysis could land in *Quadrant I*. Thus, we may conclude that *NEXTREL* produces data of higher quality than *Excel*.

It is generally quicker for users to create a new sample using *Excel* that results in high data completeness. However, this comes at the expense of less accuracy and consistency. Users require fewer mouse clicks but more keystrokes due to the input type of free text fields, as they must manually type out the information. This is also why data generated by *Excel* tend to be more inaccurate and inconsistent when compared to a standardized tool like *NEXTREL*. *Excel* allows users to enter any information regardless of format, which can be advantageous if one needs to include additional information. This is currently not possible with *NEXTREL*.

With *NEXTREL*, users require more time to create a new sample, but the resulting data have higher accuracy and consistency. Additionally, the data are also highly complete as users are not allowed to create samples with insufficient information. Compared to using *Excel*, users need more mouse clicks but fewer keystrokes because of dropdown menus that help maintain high accuracy and consistency.

Based on the usability feedback, both experienced and non-experienced users consider *NEXTREL* to have higher usability than *Excel*. *NEXTREL* fulfills the usability dimensions to some extent, being effective, efficient, engaging, error tolerant and somewhat easy to learn. However, the feedback indicates that using *NEXTREL* requires training or technical support, regardless of user's age or experience level. This could be a drawback for users who prioritize time above all else, or those who have not been adequately introduced to the tool.

## 6. Conclusion

In this thesis, we conducted a formative evaluation of two data management tools: *Excel* and *NEXTREL*, focusing on data quality and usability. This evaluation aims to identify the strengths and weaknesses of both tools, which provides valuable feedback that we can use to improve the design of the *NEXTREL* prototype as part of an iterative development process. In order to conduct such an evaluation, we examined the dimensions of data quality and the aspects of usability, along with methods to measure them objectively and subjectively.

There are various definitions of data quality, but we focused on the framework proposed by Pipino et al. [30], which is "Data Quality Assessment". The authors mention that there is no "one size fits all" method to measure data quality, so we adopted the definition of "fitness for use". Hence, we concentrated on three key data quality dimensions: accuracy, completeness and consistency. In terms of usability, we used Quesenbery's definition, which comprises the *5Es*: effective, efficient, engaging, error tolerant and easy to learn [31]. We also looked into suitable methods for assessing data quality and system usability, such as using equations for objective measurement and questionnaires for subjective assessment. Additionally, we designed a suitable study workflow with clear task descriptions to evaluate both data quality and system usability. Participants were involved in testing the tools and completing the questionnaires.

Through the evaluations, we conclude that the *NEXTREL* prototype generally performed better in both objective and subjective assessments than *Excel* despite a few exceptions. *NEXTREL* is able to improve data quality and usability as a data management tool to handle samples when compared to *Excel*. This benefits engineers by providing reliable data and minimizing errors in the "Test Sample" workflow, where they test the reliability of samples. However, there is still room for improvement. The interface of *NEXTREL* could be designed to be more intuitive for users to navigate and use without needing training or technical support. Furthermore, using a larger sample size of participants could improve the accuracy of the subjective assessment results.

This thesis opens up more opportunities for exploring the formative evaluation of data management tools. It serves as a solid starting point for conducting more effective evaluations using better methodologies. Future work can investigate addi-

## *6. Conclusion*

tional data quality dimensions like reliability, timeliness and relevance to deepen the understanding of data quality across various contexts. Additionally, we can conduct subgroup analyses to determine the specific features that both non-experienced and experienced users look for in such tools. These evaluations can also then be applied in other fields, such as healthcare, finance and logistics, where high data quality and usability are essential. As analytics, automation and artificial intelligence continue to evolve, the importance of maintaining high data quality will become even more critical, serving as the foundation for innovation across industries.



# Acronyms

<b>CSUQ</b>	Computer System Usability Questionnaire
<b>ISO</b>	International Organization for Standardization
<b>NEXTREL</b>	Next Reliability
<b>NED</b>	Normalized Edit Distance
<b>QUIS</b>	Questionnaire for User Interface Satisfaction
<b>SUS</b>	System Usability Scale



## Bibliography

- [1] F. G. Alizamini, M. M. Pedram, M. Alishahi and K. Badie, "Data Quality Improvement using Fuzzy Association Rules," in *Proceedings of 2010 International Conference on Electronics and Information Engineering*, vol. 1, pp. V1-468-V1-472, 2010.
- [2] S. Alsaqqa, S. Sawalha and H. Abdel-Nabi, "Agile Software Development: Methodologies and Trends," *International Journal of Interactive Mobile Technologies*, vol. 14, no. 11, pp. 246–270, 2020.
- [3] J. J. Barnette, "Effects of Stem and Likert Response Option Reversals on Survey Internal Consistency: If You Feel the Need, There is a Better Alternative to Using those Negatively Worded Stems," *Educational and Psychological Measurement*, vol. 60, no. 3, pp. 361–370, 2000.
- [4] C. Batini, D. Barone, F. Cabitza and S. Grega, "A Data Quality Methodology for Heterogeneous Data," *International Journal of Database Management Systems*, vol. 3, no. 1, pp. 60–79, 2011.
- [5] C. Batini and M. Scannapieca, *Data Quality: Concepts, Methodologies and Techniques*, Springer Berlin Heidelberg, 2006.
- [6] W. C. M. Belzak and J. R. Lockwood, "Estimating Test-Retest Reliability in the Presence of Self-Selection Bias and Learning/Practice Effects," *Applied Psychological Measurement*, vol. 48, no. 7-8, pp. 323–340, 2024.
- [7] M. L. Brodie, "Data Quality in Information Systems," *Information & Management*, vol. 3, no. 6, pp. 245–258, 1980.
- [8] J. Brooke, "SUS: A 'Quick and Dirty' Usability Scale," in *Usability Evaluation In Industry*, CRC Press, 1996.
- [9] C. Cappiello, C. Francalanci and B. Pernici, "Data Quality Assessment from the User's Perspective," in *Proceedings of the 2004 International Workshop on Information Quality in Information Systems*, pp. 68–73, 2004.
- [10] J. P. Chin, V. A. Diehl and K. L. Norman, "Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 213–218, 1988.

## Bibliography

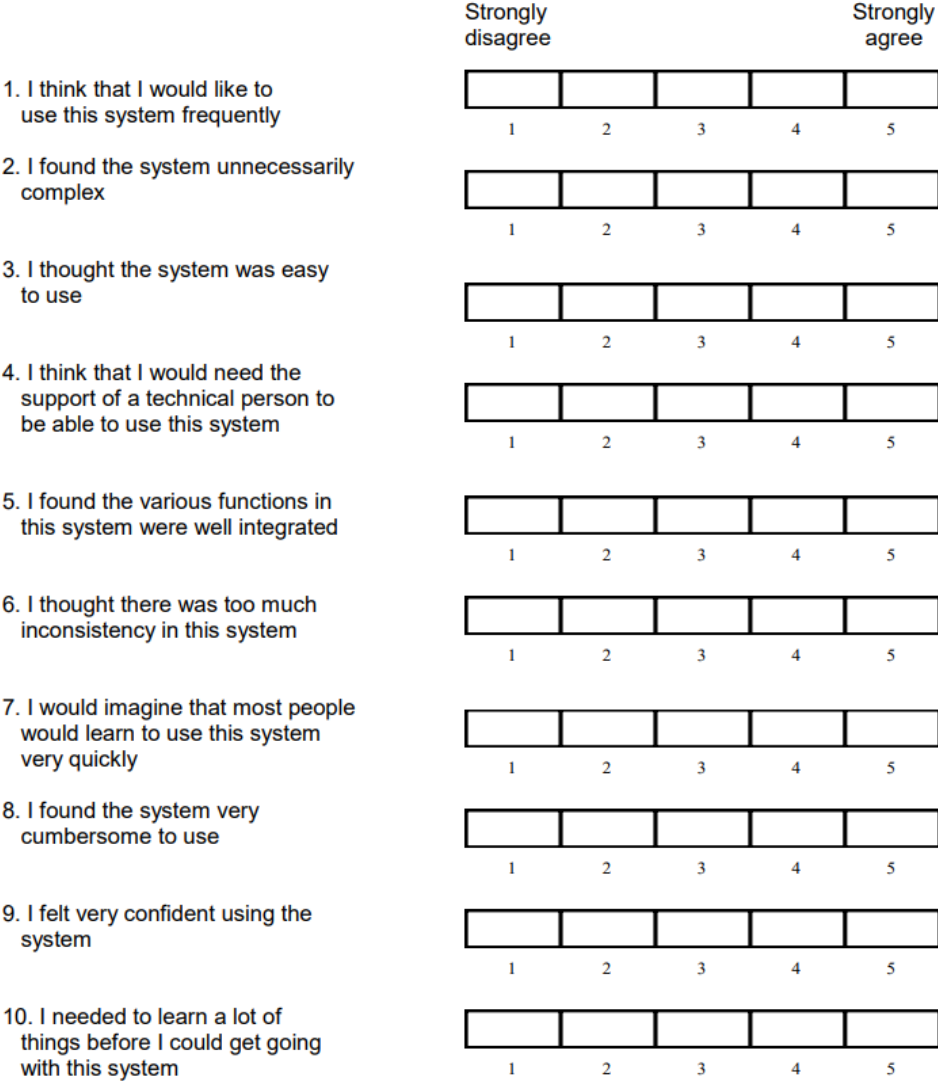
- [11] A. K. Elmagarmid, P. G. Ipeirotis and V. S. Verykios, "Duplicate Record Detection: A Survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 1, pp. 1–16, 2007.
- [12] W. Elouataoui, I. El Alaoui, S. El Mendili and Y. Gahi, "An Advanced Big Data Quality Framework Based on Weighted Metrics," *Big Data and Cognitive Computing*, vol. 6, no. 4, p. 153, 2022.
- [13] D. Gerdeman, "Companies Love Big Data But Lack the Strategy To Use It Effectively," Harvard Business School, 2017. [Online]. Available: <http://hbswk.hbs.edu/item/companies-love-big-data-but-lack-strategy-to-use-it-effectively>. [Accessed: Jun. 17, 2024].
- [14] J. D. Gould, S. J. Boies and C. Lewis, "Making Usable, Useful, Productivity-Enhancing Computer Applications," *Communications of the ACM*, vol. 34, no. 1, pp. 74–85, 1991.
- [15] J. D. Gould and C. Lewis, "Designing for Usability: Key Principles and What Designers Think," *Communications of the ACM*, vol. 28, no. 3, pp. 300–311, 1985.
- [16] J. Grudin, "Utility and Usability: Research Issues and Development Contexts," *Interacting with Computers*, vol. 4, no. 2, pp. 209–217, 1992.
- [17] H. R. Hartson, T. S. Andre and R. C. Williges, "Criteria For Evaluating Usability Evaluation Methods," *International Journal of Human-Computer Interaction*, vol. 15, no. 1, pp. 145–181, 2003.
- [18] H. R. Hartson and D. Boehm-Davis, "User Interface Development Processes and Methodologies," *Behaviour & Information Technology*, vol. 12, no. 2, pp. 98–114, 1993.
- [19] D. Hix and H. R. Hartson, *Developing User Interfaces: Ensuring Usability Through Product & Process*, John Wiley & Sons, 1993.
- [20] J. K. Kies, R. C. Williges and M. B. Rosson, "Coordinating Computer-Supported Cooperative Work: A Review of Research Issues and Strategies," *Journal of the American Society for Information Science*, vol. 49, no. 9, pp. 776–791, 1998.
- [21] S. Kraus, S. Durst, J. J. Ferreira, P. Veiga, N. Kailer and A. Weinmann, "Digital Transformation in Business and Management Research: An Overview of the Current Status Quo," *International Journal of Information Management*, vol. 63, p. 102466, 2022.
- [22] Y. W. Lee, L. L. Pipino, R. Y. Wang and J. D. Funk, *Journey to Data Quality*, The MIT Press, 2006.

- [23] J. R. Lewis, "IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use," *International Journal of Human-Computer Interaction*, vol. 7, no. 1, pp. 57–78, 1995.
- [24] J. R. Lewis, "Psychometric Evaluation of the Post-Study System Usability Questionnaire: The PSSUQ," in *Proceedings of the Human Factors Society Annual Meeting*, vol. 36, pp. 1259–1260, 1992.
- [25] D.-Y. Liu, S.-W. Chen and T.-C. Chou, "Resource Fit in Digital Transformation: Lessons Learned from the CBC Bank Global E-Banking Project," *Management Decision*, vol. 49, no. 10, pp. 1728–1742, 2011.
- [26] A. M. Lund, "The Need For a Standardized Set of Usability Metrics," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 42, pp. 688–690, 1998.
- [27] J. Nielsen, *Usability Engineering*, Morgan Kaufmann Publishers, 1994.
- [28] M. Obitko, V. Jirkovský and J. Bezdíček, "Big Data Challenges in Industrial Automation," in *Industrial Applications of Holonic and Multi-Agent Systems*, pp. 305–316, 2013.
- [29] K. Orr, "Data Quality and Systems Theory," *Communications of the ACM*, vol. 41, no. 2, pp. 66–71, 1998.
- [30] L. L. Pipino, Y. W. Lee and R. Y. Wang, "Data Quality Assessment," *Communications of the ACM*, vol. 45, no. 4, pp. 211–218, 2002.
- [31] W. Quesenbery, "The Five Dimensions of Usability," in *Content and Complexity*, Routledge, 2003.
- [32] F. Quin, D. Weyns, M. Galster and C. C. Silva, "A/B Testing: A Systematic Literature Review," *Journal of Systems and Software*, vol. 211, p. 112011, 2024.
- [33] J.-L. Rassineux and H. Proff, "Digital Maturity Index," Deloitte, 2023. [Online]. Available: <https://www.deloitte.com/global/en/Industries/industrial-construction/perspectives/digital-maturity-index.html>. [Accessed: Jun. 17, 2024].
- [34] W. J. Roesch and S. Brockett, "Field Returns, a Source of Natural Failure Mechanisms," *Microelectronics Reliability*, vol. 47, no. 8, pp. 1156–1165, 2007.
- [35] K. U. Schulz and S. Mihov, "Fast String Correction with Levenshtein Automata," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 67–85, 2002.
- [36] *Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Data Quality Model*, ISO/IEC 25012, 2008.

## Bibliography

- [37] D. Stone, C. Jarrett, M. Woodroffe and S. Minocha, *User Interface Design and Evaluation*, Morgan Kaufmann Publishers, 2005.
- [38] *Systems and Software Engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Measurement of Data Quality*, ISO/IEC 25024, 2015.
- [39] T. Tullis and J. Stetson, "A Comparison of Questionnaires for Assessing Website Usability," in *Proceedings of UPA 2004 Conference*, 2004.
- [40] R. Vaziri, M. Mohsenzadeh and J. Habibi, "Measuring Data Quality with Weighted Metrics," *Total Quality Management & Business Excellence*, vol. 30, no. 5-6, pp. 708–720, 2019.
- [41] R. A. Virzi, "Refining the Test Phase of Usability Evaluation: How Many Subjects Is Enough?" *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 34, no. 4, pp. 457–468, 1992.
- [42] Y. Wand and R. Y. Wang, "Anchoring Data Quality Dimensions in Ontological Foundations," *Communications of the ACM*, vol. 39, no. 11, pp. 86–95, 1996.
- [43] R. Y. Wang, "A Product Perspective on Total Data Quality Management," *Communications of the ACM*, vol. 41, no. 2, pp. 58–65, 1998.
- [44] R. Y. Wang and D. M. Strong, "Beyond Accuracy: What Data Quality Means to Data Consumers," *Journal of Management Information Systems*, vol. 12, no. 4, pp. 5–33, 1996.
- [45] M. Webb, "Single Source of Truth (SSoT)," Techopedia, 2024. [Online]. Available: <https://www.techopedia.com/definition/single-source-of-truth-ssot>. [Accessed: Jun. 17, 2024].

# A. Appendix



**Figure A.1:** The standard *System Usability Scale (SUS)* [8]

## A. Appendix

Shelf	Location Detail	Lot Number	Wafer Number	Technology	Responsible	Scrap Date
4	5	TU889614.03	1,2,3	HSTF1200	Max Turan	Dezember 24
4	5	TU223344	#3-#20	CIT110305	Franz Giebel	Dezember 25
4	4	TU911312	1,4,7,10,13,16,19	CIT120310	Kloppenburg	Dezember 30
4	4	QU778812.04	1,2,3	C9LMG	Kloppenburg	Dezember 30
4	4	QU778812.01	22,23,24,25	C9LMG_GG	Kloppenburg	Dezember 30
		PL887026	4	CARD 7	I.Kranz	Dezember 25
		P99667 UWU	1,2,3	INNOVAT TUM	Fromme	Dezember 25
4	3	GP0210201	13, 15, 17,18	P90TPA	Rebstock	Oktober 40
		GP2269622	9, 10	P90TPA	Rebstock	Oktober 40
4	3	HF151005.00	7	MOSFET - 10V	L.Balz	Dezember 25
4	3	HF888000.02	2	MOSFET - 10V	L.Balz	Dezember 25
4	3	PO118005.09	3	MOSFET - 10V	L.Balz	Dezember 25
		6ABB87HQ22	4,5,6	CPTZ	H.Schaffer	
4	3	HF212007	18	CPTZ		Jan-25
1	5	QU106700	1,3,5,6,23,24	PMOS99R (GaN)	L.Deckert	Dec-25

**Figure A.2:** Section of status quo data with altered details for confidentiality purposes

Shelf	Location Detail	Lot Number	Wafer Number	Technology	Responsible	Scrap Date
4	5	TU889614.03	1	HSTF1200	Max Turan	2024-12-01 00:00:00
4	5	TU889614.03	2	HSTF1200	Max Turan	2024-12-01 00:00:00
4	5	TU889614.03	3	HSTF1200	Max Turan	2024-12-01 00:00:00
-	-	-	1	INNOVATIV_TUM	Iris Fromme	2025-12-01 00:00:00
-	-	-	2	INNOVATIV_TUM	Iris Fromme	2025-12-01 00:00:00
-	-	-	3	INNOVATIV_TUM	Iris Fromme	2025-12-01 00:00:00
w4	4	QU778812.04	1	C9LMG	Sid Kloppenburg	2030-12-01 00:00:00
w4	4	QU778812.04	2	C9LMG	Sid Kloppenburg	2030-12-01 00:00:00
w4	4	QU778812.04	3	C9LMG	Sid Kloppenburg	2030-12-01 00:00:00
4	3	GP021020	13	P90TPA	Hans Rebstock	-
4	3	GP021020	15	P90TPA	Hans Rebstock	-
4	3	GP021020	17	P90TPA	Hans Rebstock	-
4	3	GP021020	18	P90TPA	Hans Rebstock	-
4	3	HF888000.02	2	-	Leon Balz	2025-12-01 00:00:00
1	5	QU106700	1	PMOS99R_(GaN)	-	2025-12-01 00:00:00

**Figure A.3:** Section of data after the prototype implementation with altered details for confidentiality purposes



Column	Data Accuracy	
	Excel	NEXTREL
Shelf	0.97	0.99
Location Detail	0.97	0.99
Lot Number	0.91	0.85
Wafer Number	0.51	1.00
Technology	0.39	0.67
Responsible	0.51	0.91
Scrap Date	0.82	0.86

**Table A.1.:** Quantitative data accuracy between *Excel* and *NEXTREL*

Column	Data Completeness	
	Excel	NEXTREL
Shelf	0.97	0.99
Location Detail	0.97	0.99
Lot Number	1.00	0.85
Wafer Number	0.88	1.00
Technology	1.00	0.67
Responsible	0.99	0.91
Scrap Date	0.99	0.86

**Table A.2.:** Quantitative data completeness between *Excel* and *NEXTREL*

A. Appendix

Column	Data Consistency	
	Excel	NEXTREL
Shelf	0.97	0.99
Location Detail	0.97	0.99
Lot Number	0.78	0.85
Wafer Number	0.40	1.00
Technology	0.59	0.67
Responsible	0.23	0.91
Scrap Date	0.71	0.86

**Table A.3.:** Quantitative data consistency between *Excel* and *NEXTREL*