Expertise
and insight
for the future

Salla Vesterinen

Creation of Automated EazyBI Data Graphs for Project Progress Reporting

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

24 March 2021

Author Title Number of Pages Date	Salla Vesterinen Creation of Automated EazyBI Data Graphs for Project Progress Reporting 39 pages 24 March 2021
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Smart Systems
Instructors	Sami Sainio, Senior Lecturer

The purpose of this study was to create automated EazyBI graphs for Airbus Defence and Space based on provided project data. The goal was to provide summarized and easily understandable data graphs for different projects to be able to see the progress and possible problem areas.

Visualized data graphs are needed to provide an easy but comprehensive view of the project data since results can often go unnoticed or be overlooked if shown in a more traditional, non-visualized manner. There are often difficulties when aiming to provide data optimally since every case is different, and the tools should be chosen accordingly.

The approach taken was to start by going through the different tools and types of data used and received in each project, and agreeing with the project managers on the key progress indicators to focus on. The data received for each project was imported to the chosen Business Intelligence tool, EazyBI from Jira, and could be processed and formatted there before publishing as dashboards in Confluence.

As a result, these new data report graphs should provide necessary information about the progress of each project concisely which meets the needs of the company. The data that can be presented in the new graphs should be more in-depth and dynamically visualized, therefore providing more information than before. Based on this thesis it should also be possible to create further graphs for different teams in the company.

Keywords	business intelligence, EazyBI, reporting, visualization, data

Tekijä Otsikko Sivumäärä Aika	Salla Vesterinen Automatisoitujen EazyBI kaavioiden luominen projektin edistyksen raportointiin 39 sivua 24.3.2021
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikka
Ammatillinen pääaine	Älykkäät järjestelmät
Ohjaaja	Lehtori Sami Sainio

Tämän insinöörityön aiheena on automatisoitujen EazyBI-kaavioiden luominen Airbus Defence and Space -yhtiölle saadusta datasta. Tavoitteena on tarjota helposti ymmärrettävät kaaviot eri projekteille, jotta niistä voidaan nähdä yhteenvetona edistyminen ja mahdolliset ongelma-alueet.

Syy visualisoitujen datakaavioiden tarpeeseen on tarjota helppo mutta kattava näkymä datasta, koska tulokset voivat usein jäädä huomaamatta, jos ne esitetään perinteisemmällä ei-visualisoidulla tavalla. Datan näyttämiseen optimaalisesti liittyy usein vaikeuksia, koska jokainen tapaus on erilainen ja työkalut tulisi valita tapauksen mukaan.

Projekti aloitettiin käymällä läpi eri työkalut sekä projekteissa käytetty ja luotu data. Tämän lisäksi projektipäälliköiden kanssa sovittiin keskeisistä edistymisindikaattoreista, joihin keskittyä. Jokaisesta projektista saatu data importoitiin EazyBI:hin Jirasta, jossa ne pystyttiin käsittelemään ennen julkaisemista Confluenceen.

Tämän tuloksena näiden uusien kaavioiden tulisi antaa tarvittavat tiedot kunkin projektin etenemisestä ytimekkäästi yrityksen tarpeiden mukaisesti. Uusissa kaavioissa esitettävien tietojen tulisi olla perusteellisemmin ja dynaamisemmin visualisoituja, ja siten tarjota enemmän tietoa kuin aikaisemmin. Tämän insinöörityön perusteella pitäisi olla myös mahdollista luoda lisää EazyBI-raportteja yrityksen eri tiimeille.

Avainsanat	liiketoimintatiedon hallinta, EazyBI, raportointi, visualisointi, data

Contents

List of Abbreviations

KPI	Key Performance Indicator
BI	Business Intelligence
OLAP	Online Analytical Processing
MDX	Multi-Dimensional eXpressions
ETL	Extract, Transform and Load
RDBMS	Relational Database Management System
SQL	Structured Query Language
AI	Artificial Intelligence
HTML	Hypertext Markup Language

1 Introduction

This study focuses on the creation of automated EazyBI data graphs for the Secure Land Communication branch of Airbus Defence and Space. The creation of the graphs was based on provided project data. The goal is to provide summarized and easily understandable data graphs for different project groups to be able to see the key performance indicators (KPIs) and possible problem areas through them.

Visualized data graphs are needed to provide an easy but comprehensive view of the project data since results can often go unnoticed or be overlooked if shown in a more traditional, non-visualized manner [1]. There are often difficulties when aiming to provide data optimally since every case is different, and the tools should be chosen accordingly.

Some challenges facing data visualization include the level of understanding and possible technical skills, which is why Business Intelligence tools are advantageous. There are also problems with data visualization, such as oversimplification of data, and overreliance on visuals.

In his book *The Truthful Art: Data, Charts, and Maps for Communication*, Alberto Cairo states that the five qualities of great visualization are truthfulness, functionality, beauty, insightfulness, and enlightening quality. He also discusses biased and dishonest visualization. [2.] With this in mind, the approach taken in the final year project was to start by going through the different tools, types of data used and received in each project, and agreeing with the project managers on the key performance indicators to focus on.

The data received for each project was imported to the Business Intelligence (BI) tool EazyBI from project tracking software called Jira. The data could be processed and formatted in EazyBI before publishing in gadgets or as dashboards in the Confluence workspace tool. EazyBI was chosen in this case because of the ease of which it could be added as a plug-in to the Atlassian collaboration tool Jira, which was already in use. Another option discussed was QlikView, and although it offers more features, EazyBI provided everything needed in this case. These required features included dashboards, key performance indicators, performance metrics, predictive analytics, publishing, trend and problem indicators, and most importantly visual analytics.

As a result, these new data report graphs should provide necessary information about the progress of each project concisely, therefore meeting the requirements of the project set by the company. The data that can be presented in the new graphs should be more in-depth and dynamically visualized, therefore providing more information than before when data visualization was not used.

Based on this study it should also be possible to create further graphs for different teams in the company.

2 Theoretical Background

This chapter deals with the usage of BI tools with a focus on their usage in data analysis. The goal is to provide an overview and compare the distinctive features. This is an important topic because of the widespread use of BI tools. With several diverse types of tools, differentiating between them is key for effectiveness.

BI tools are application software used for processing data. They are widely used because of their increasing importance of data. There have been various developments in unique ways of using BI tools and some of them will be reviewed in this section.

2.1 Background of Business Intelligence Tools

The term Business Intelligence has a long history and there have been many steps for it to have evolved to be what it is today. The next section focuses on the history and then delves into some current use cases and benefits of these tools.

2.1.1 History

The first person to use the term Business Intelligence was Richard Miller Devens in the book ‘Cyclopedia of Commercial and Business Anecdotes’ which was released in 1865. Though the term did not have the same meaning as it does now, Devens used it in reference to data usage for informed business decisions. [3, 210.] This was when companies realized the possibility to make improved decisions by considering the current performance.

It was not till 1958 and Hans Peter Luhn that the term was used as it is recognized today. Working as a researcher for IBM, Luhn published an article in the IBM Journal where he specified a Business Intelligence system to be able to give information to support actions. Luhn further explains how a BI tool can provide the possibility for selective data distribution, therefore, fixing communication errors. [4, 315.]

The foundation of BI was built around the need for improvement in making business decisions using fact-based systems though nowadays it involves gathering data, storing the data, and knowledge management to analyze complex information. This ability to retrieve useful information from complex data is what makes BI tools so important. With an increased necessity for accurate and fast decision making, in a world full of data with the evolution of computers and databases, it is unsurprising that there are increasingly more BI software tools available.

The first business analytics system was introduced by Frederick Taylor through the study of data analysis to increase industrial production efficiency in the late 1800s [5]. When computers were invented, people such as Luhn could continue the implementation of BI tools which eventually grew to be the software used now.

The future of BI tools seems bright because of the clear expansion during the past years which shows no signs of slowing down.

2.1.2 Use Cases and Benefits

There are a plethora of ways Business Intelligence tools are currently used, with varying reasons and benefits. There was a three-part study from The University of Texas 'Measuring the Business Impacts of Effective Data' [6]. It demonstrated the impacts of effective data in 150 Fortune 1000 firms, addressing not only the quality of the data but the usability and intelligence as well. They developed a conceptual model assessing the impact of data on financial performance.

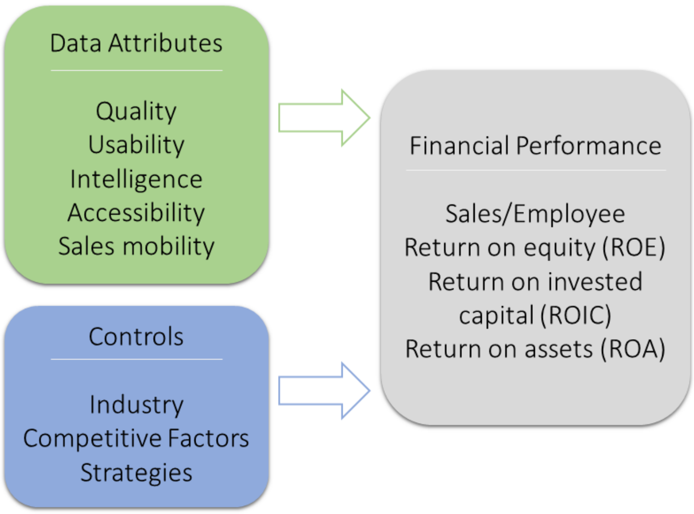

Figure 1. Financial performance impacts of data attributes. Adapted from Barua et al. 2010 [6,4].

The five different data attributes considered in the study are illustrated in Figure 1. Quality impacts how good the data and outcome of any data analysis is. It incorporates data accuracy, scope, timeliness, and recency. Usability is seen in the data presentation, ability to be further processed and manipulated, and consistency across various sources. Accessibility is a data attribute including how the data can be accessed. For example, many companies want remote data access by authorized users. Sales mobility on the other hand includes the possibility for data exchange with customers through portable systems. Lastly, there is intelligence, which is the most popular attribute. This includes how well the data can be rearranged to show data trends and patterns for making decision and action recommendations. [6,4.]

Using these data attributes and looking into the various companies the study concluded that when increasing effective data, the returns far outweigh the costs required [6, 13].

There are plenty of benefits and reasons for business data analytics. Data is what drives continual performance improvements. It encourages decision-making since data can back up decisions with evidence. Data provides a competitive advantage in the modern world because at the end of the day it boils down to one goal: revenue gain and therefore increased profits.[6,11].

BI tools are not only used by technology companies, but also extremely helpful to the healthcare and education industries, and the government [7]. These big and fast-paced industries can use BI data analysis for cost reduction as an example. Data is also helpful in politics and crisis management situations in both response and recovery aspects.

2.1.3 Software

With a market full of different BI software and tools it can be daunting to figure out which one meets the corporate needs. Whether that includes problem identification, marketing strategies, or streamlined communication, they all analyze the entered data.

Microsoft Power BI is the leading Business Intelligence platform designed for business analysts and data scientists. It is a no-code platform with a library of connectors and a drag-and-drop interface. Providing Microsoft Excel integration and deployment pipelines, Power BI gives the ability to stream data analytics in real-time. [8.]

Tableau is another widely used data visualization tool. It aims to simplify raw data into an understandable format and identification of patterns, with a range of features and the creation of real-time visualizations. [7.]

QlikView is a data discovery tool used to create analytics and a dashboard for targeted challenges, providing the ability to uncover data and data relationships. It uses an in-memory data model which is helpful when handling big data sets and allows for analytic calculations with R and Python. [9.]

Google Data Studio is a free, web-based suite offered by Google which works well with data sourced from Google, for example, YouTube and some third-party connectors [10]. IBM Cognos Analytics, on the other hand, is a cloud-based platform with the ability to import data from a wider variety of sources. It uses machine learning. [11.]

EazyBI is another BI tool with a variety of powerful features. EazyBI provides a solution to creating customizable reports, charts, and dashboards with interactive data analysis and visualization. Further calculations can also be made using the MultiDimensional eXpressions (MDX) query language used for Online Analytical Processing (OLAP) with the use of a database management system. [12.]

2.2 Business Intelligence Tool Features

Business Intelligence tools can be used for data accumulation, storage, analysis, knowledge management or a combination depending on the tool used. There are many ways to categorize BI tools, but the focus here is data storage, aggregation, analysis, and visualization.

2.2.1 Data Storage

The efficient use of a BI tool requires data storage with a low response time. Filtering, aggregating, and analyzing data all profit from data structures with the possibility of scanning substantial amounts of data fast. Indexing, partitioning, and column-oriented storage are examples of ways to achieve this [13].

An extract, transform, and load (ETL) tool is needed to first get the data from the source (extract), clean and integrate the data (transform), and last place it into a data warehouse (load). The ETL procedure can be challenging but verify that the data is correct and of quality. Most tools require some form of data manipulation in order to integrate data for use. [13, 22.]

The first part of data storage is the data source with the original data. This could be a spreadsheet, an external source service, or enterprise resource planning software. The real-time capability of a BI tool differs depending on the data extraction possibilities.

The second part of data storage is where the extracted data is held. A data warehouse is usually used to preserve data history and integrity. A Relational Database Management System (RDBMS) is designed to store data using rows and columns to keep its structure, but BI tools typically query the data outside of the RDBMS data warehouse.

The third part of data storage is the analytical database where the BI data analysis happens. It provides a fast and flexible query analysis when comparing to an RDBMS. Most analytical databases store some of the data in memory in a compressed way and have an expressive query language for calculations.

2.2.2 Data Aggregation with Queries

The process of data aggregation means presenting data retrieved in a concise format. Most BI tools go through data with queries. A query is an action to request data from the database and the interface depends on the tool used [13]. With the use of a query language, it is possible to query copious amounts of data to generate only the wanted part. This is done through different code words that vary between languages. Structured Query Language (SQL) is universally used and MDX is another example.

By querying data, it is possible to get specific sets to pinpoint smaller aspects [13]. An example could be getting information on customers and sales peaks by querying the data on purchases without having to manually sort through the data first.

A simple query would be for obtaining specific data from a database with a selection function. On the other hand, to get a single result from a set of values, aggregate functions are used. Examples of aggregate functions are average and count. It is also possible to retrieve all the rows in all the tables of a query with a Cartesian product, also known as a cross-join and a further filter with nested queries.

Data aggregation by BI tools is necessary to discover information from the data retrieved. OLAP provides a multidimensional view of data which can be used to aggregate data from multiple sources to summarize it [13, 24].

2.2.3 Data Analysis

Where data aggregation is helpful to summarize and sort through data, data analysis takes it further to uncover patterns and to discover relationships between data through algorithms. This is done through a combination of statistics, artificial intelligence, and machine learning and keeps evolving with the increasing size of databases. An example of this would be data mining.

Data mining is a method of searching for connections in large data sets in order to predict outcomes. The information based on data through data mining can be used in business analytics for revenue growth, cost cuts, and risk reduction by finding cause and effect statistics. [14.] This is increasingly important for companies since the rapid growth of data obstructs rapid response, which is crucial.

Data analysis is not only used to identify patterns and correlations within data sets but also to make conclusions to predict future outcomes. This is done to make better decisions when faced with multiple options. Although centralizing data and seeing the results is beneficial, the real power of data analysis comes from being able to make decisions based on that data.

Artificial Intelligence (AI) is a field of data science using algorithms for computers to learn on their own. Data analytics and AI bring the possibility of linking data in order to gain insight, whether that be related to business or not [15]. Advances in AI and machine learning can help automate various stages of data analytics and therefore improve aspects such as productivity, speed, and efficiency [16].

2.2.4 Analytical Visualization

Processing information can be taxing, but with the use of charts and graphs to visualize complex data it is easier to communicate results more universally.

Though data visualization is done with data results, it can help identify areas that need attention just like the previously mentioned BI tool features. Using tables and graphs to organize and present data, the results can be understood by more people because of the easily understandable format. The use of data visualization allows for utilization by people who might not understand BI tools, therefore speeding things up.

Dashboards are a way of sorting data and graphs into sets to easily see what is to be conveyed through the data. The amount of data for a single project can be overwhelming and condensing it into a single figure can identify problem areas, clarify factors, speed things up, and help everyone understand the project better.

Depending on the BI tool, these visual reports can also be automated to update when new data is inputted, cutting time and therefore costs as well.

3 Tools and Methods Used

Although there are plenty of BI tools to choose from, while completing this project, a variety of tools and methods were used including Jira, Confluence, and EazyBI which utilizes Mondrian OLAP and MDX. This section provides an overview of all these tools and methods used and why they were chosen for this case project.

3.1 Jira and Confluence

Jira and Confluence are both tools by Atlassian with unique features all intended for collaboration and working more efficiently, by organizing and managing team projects [17]. These tools were chosen for this case, because they were already in use in the company. This allowed for the data in these tools to be used instead of starting anew.

Jira Software is a development tool made especially for agile teams with the ability to plan, track and report things through it. Users can create issues, plan sprints and mark who is working on what with visibility to the whole team. Prioritization and tracking through these areas can therefore improve team performance, with all the data visualized in one place.

Confluence is a workspace for collaborative use for knowledge management, working as a modern intranet. It is equipped with templates to easily start and stay consistent, with options to use provided macros and tools, or for creating new macros.

For this project, Jira version 8.5.3 was used to collect project data automatically from each team. Anything being done by an individual in a team was tracked in Jira using stories, epics, and issues, containing labels and updates. In addition to Jira, Confluence version 7.4.1 was used in this project to publish reports.

3.2 EazyBI

EazyBI is a Business Intelligence tool for analyzing and visualizing data from various sources such as Jira, SQL databases, REST API sources, Google Sheets, Excel, and CSV files. It was created by Raimond Simanovskis in 2011 in order to improve typical BI project tools, for an easy-to-use application. [12.]

EazyBI was chosen in this case because it could easily be added as a plug-in to the Atlassian collaboration tool Jira, which was already in use. EazyBI provided everything needed in this case with the required features including dashboards, key performance indicators, performance metrics, predictive analytics, publishing, trend and problem indicators, and most importantly visual analytics. [12.]

EazyBI is used by many companies for creating automatized and interactive custom reports, charts, and dashboards with data analysis and visualization [12]. With a drag-and-drop report builder, new graphs can be made with only a few clicks, with the possibility for user-defined calculations.

EazyBI has a plethora of report features to use. For creating graphs, EazyBI makes it possible to import data (from multiple sources), use ready-made demo reports, create customized reports, define new calculated measures, and export/import the report definition (to easily copy someone else's graph). Once the graph is done it can be placed in Confluence or Jira, be shared as a link, or downloaded as a PDF or Excel file. Then whilst viewing these reports, it can be viewed as a table and the filters can be modified. It is also possible to hover over the data to see values (when viewing a graph), to sort for example descending values (when viewing a table), to select things to be inspected more closely, to "drill through" to the Jira issues link, and to choose to see a section (when viewing a timeline). Lastly and crucially, the graph updates itself based on the data it receives.

For this project EazyBI, version 6.0.1 was used as a plug-in for the Atlassian tools Jira and Confluence. Version 6.0.1 was updated before starting this project because of its more detailed error messages, processing, and various other attributes [12]. The EazyBI plug-in was used in this project to create visualized data reports for various company projects.

EazyBI uses Mondrian Online Analytical Processing (OLAP) reporting engine and implements the Multidimensional eXpressions (MDX) query language. With EazyBI it is possible to not only use MDX functions provided by Mondrian but also define additional MDX functions that are used in calculated member formulas. [12.]

3.3 Mondrian OLAP and MDX

EazyBI was the chosen BI tool used for this project, and therefore the Mondrian OLAP and the MDX query language were used.

Mondrian OLAP is an open-source reporting engine written in Java. It is used to aggregate data in a memory cache by reading data from RDBMS. There are four layers in a Mondrian OLAP system: the presentation, dimensional, star, and storage layers. The presentation layer is about what the end-user sees and what they can do to interact. The dimensional layer is for parsing, executing, and validating MDX queries. The star layer is for the aggregate cache, and lastly, the storage layer is the RDBMS which provides aggregated data. [18.]

MDX is a query language for querying the data cube and for adding logic to the cubes. In EazyBI and this project, MDX was used to define customized and calculated measures and member properties from a data cube.

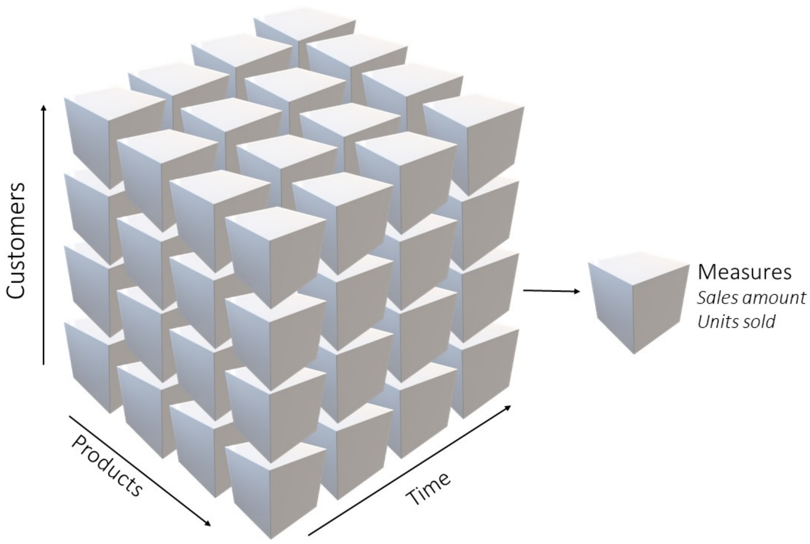

Figure 2. Multi-dimensional data cube. Adapted from [19].

The **data cube** as illustrated in Figure 2, stores data imported to EazyBI and is divided into dimensions and further into measures. This data cube in EazyBI is also not limited to the traditional three dimensions, but can include as many measures as needed, with the option to also create multiple cubes. Separate cubes might be created because all measures in a cube have the same dimensions.

A **dimension** is for example *customer*, with further hierarchy levels of *country* or *name*. With each level below, they are all aggregated into the levels above it.

Members are items in the dimensions. As an example, the dimension *country* can include a member such as *Finland.* Each dimension also always includes a default member.

Measures are typically numeric values accessed through a dimension member level or aggregated at higher levels. This could be *sold units*, for example.

Dimension members also have **properties** that hold information about members. The dimension *customer* could have a member *person* and, in turn, each member could have a property like *address.* [19.]

4 Processing Data

There are multiple aspects to understand and think about when it comes to creating graphs based on received data. Graphs vary greatly based on the type of data used, and not only how they are imported and mapped but also how they are implemented makes a difference. This section focuses on these aspects in addition to some shared challenges when it comes to visualizing data and some qualities that make data visualization great.

4.1 Data Source

There are two main types of data: quantitative and qualitative. Quantitative data is numbers and things that can be objectively measured, for example temperature and prices. Qualitative data on the other hand is characteristics and descriptors that can be observed but not measured, textures and colors being examples. [20.] These two main types of data can be further grouped into more data types.

In the case of this project, each team would update project data into Jira which would then be imported into EazyBI. This data included mainly quantitative data such as test case amounts and results. In addition to marking a test case completed, Jira would also save information like who did it and at what time. This data which was imported into EazyBI could then be aggregated and analyzed further to be able to generate an educated guess of when a project will be finished, by calculating the speed of test cases being completed and comparing that to the number of test cases that are still open.

One key functionality of EazyBI used as a plug-in with Jira is the fact that each time new data, which is regularly imported to EazyBI, is added, EazyBI also updates all the EazyBI reporting graphs done. This makes creating automated data graphs possible. Other tools also allow this, like for example QlikView and PowerBI, but the tool and source should be chosen solely based on requirements for each individual project depending on the data type, data source, and needs for the result.

4.2 EazyBI

If the chosen BI tool is EazyBI, the Data Administrator is responsible for choosing what data is being imported from various source applications, such as Jira, SQL, Excel, or Google Spreadsheets [12].

In the case of this project, data was imported from Jira, after which the data cube was named and created. The available options depend on the source, but sample reports can be imported, and the frequency of data import can be chosen (the default is daily). Import status transitions and corresponding measures can be chosen, for them to be used and calculated in some measures. For example, *issues* can be *created, in progress, open, reopened, resolved, tested*, or *closed*. There are also further import options that include importing the history, age, and resolution intervals of an estimation of remaining hours.

User group dimensions should also be chosen, with the default being that all user groups are created as user group dimension members, but it can be chosen to only show limited fields. Finally, it can also show available custom fields from a list recognized by EazyBI though there are advanced settings for further additions.

Value type custom fields can be imported into additional issue cube dimensions. Numeric custom fields on the other hand can be imported as measures (the sums of these fields can be analyzed by other dimensions). Date custom fields can be imported as measures (e.g., issues with corresponding periods can be compared). Custom fields can also be imported as properties. For numerical date custom fields, historical changes in the customized field values can also be imported. [21.]

If the data is imported from a spreadsheet, then data mapping is needed. This is choosing what data corresponds to what dimension, with the numerical facts describing the dimensions being the measures. Though data mapping can become complex, there are some simple rules to follow. There must be at least one measure (with decimal or integer values), and only one “time” dimension (though there can be multiple columns with DateTime values). When going through the hierarchical order (e.g., *Country* > *State* > *City* > *Customer*), the top-level needs to be on the left of the spreadsheet, and the values imported as properties need to be specific for each member. [22.]

After choosing and saving the options for what data is imported, the import itself can start. The source selection, cube, status, and actions can all be seen in the EazyBI Source Data section. Though an import can be manually done at any time, for efficient use the next import follows regular import date frequency that can be specified in the creation process.

The data cube where all the data is imported into can also be emptied and reimported, but so can individual dimensions if changes are needed to be done in them. It is important to note that when deleting a data cube, all the calculations, reports, and dashboards are also deleted from the account, so reports should be exported before deleting the data cube. [21.]

After successfully completing the data mapping and import without errors, it is possible to start creating graphs with the BI tool used, whether it is EazyBI or another tool.

4.3 Challenges in Data Visualization

Visualized data graphs are needed to provide an easy but comprehensive view of the data since the results can often go unnoticed or be overlooked if shown in a more traditional, non-visualized manner. There are often difficulties when aiming to provide data optimally since every case is different, and there are some things to keep in mind when creating a visualized data report, including accuracy, clarity, use, readability, automation, and updates. Because making a visualized graph is in part summarizing information into a short and concise graph that should be quick to process for anyone seeing it, it is important to know what is being indicated in it.

There are some challenges facing data visualization itself, with the level of understanding and possible technical skills being in the forefront, which is why using BI tools is advantageous. However, there was a study made which found the effectiveness and usage of BI practices to vary largely in various firms, which was reflected in technological challenges and missed opportunities [23].

There are also other challenges with visualizing data, like the oversimplification of data, and overreliance on visuals. These challenges and more can be found in each step of data analytics from data collection, data normalization and storage, to analytics and reporting [24]. Another challenge in visualizing data is the visualization of proportionality and uncertainty because visualizing data in the most typical ways does not work in every case, and the use of confidence intervals are needed. Then regarding accuracy, it is important to show the correct information and to show everything accurately so that there is no room for misreading. This includes colors that can be easily identified and seen, clearly named labels, and comments which should all be clear to both viewers and peers. Some EazyBI specific issues to keep in mind are overlapping data, labels, and commenting the MDX code.

In the case of this project, one of the key functions of EazyBI graphs used as an Atlassian product plug-in is the automation of the data graphs. As an example, it might be useful to create a calculated member "current year" for the Time dimension, instead of using a certain year like "2021". This happens because when the year changes, the graph will automatically update the display to change the year, instead of always showing the chosen year "2021".

4.4 Great Data Visualization

In his book *The Truthful Art: Data, Charts, and Maps for communication*, Alberto Cairo stated that the five qualities of great visualization are the truthfulness, functionality, beauty, insightfulness, and enlightening quality [2]. All five of these aspects are things to keep in mind when visualizing data to accomplish it well.

The first quality, truthfulness, is what Cairo repeats in his work and emphasizes the importance of. Is any data obscured or not visible? The obligation of the data visualizer is to the viewer, and the communication needs to work in a truthful manner for it to work. The second quality, functionality, is quite self-explanatory and can be increased by user testing. The third quality, beauty, is descriptive in the aesthetical sense, and even though representing data in a beautiful way is not the priority when visualizing data, it is what makes a great graph stand out. The fourth quality of great data visualization, according to Cairo, is insightfulness. For a graph to be insightful, it should display the relevant data, since too much is unnecessary. Merely visualizing data by replicating figures might not display all the relevant data, revealing important trends and relationships. The last quality in Cairo's list is enlightening quality, and even though it appears repetitive, to Cairo it means that the previous four qualities are successfully combined to produce something new and a valuable revelation. [2.]

Other than thinking about what qualities make great data visualization, there are steps that can be taken to achieve it. In an article for the Business Information Systems journal, Mathrani writes about seven different practices influencing BI effectiveness. These include aligning BI processes to achieve goals, organizational changes for proactive decision-making, training, data management strategies (for quality, secured and trustworthiness), upgrading technologies, cloud connectivity (for storage and added computing power), and collaborative BI tools (for ease of scheduling, report generation and sharing). [23.]

5 EazyBI Report Implementation

The reason for needing visualized data graphs is to provide an easy but comprehensive view of the data since the results can often go unnoticed or overlooked if shown in a more traditional non-visualized manner. There are often difficulties when aiming to provide data optimally since every case is different, and the tools should be chosen accordingly. In the case of this project the goal was to provide summarized and easily understandable data graphs for different projects to be able to see the progress and possible problem areas.

The approach in this case was creating graphs with the BI tool EazyBI. Several types of tables and graphs can be made with it, allowing the user to create quite complex, flexible, and versatile graphs with some simple steps, with the possibility of changing it further with optional MDX calculations.

This section goes through the ways of creating EazyBI reports, while remembering both the challenges in data visualization and how to attain the qualities that make it great.

5.1 Making an EazyBI Graph

After importing data to the MDX data cube in EazyBI, a user can start working on the data graphs. Once data is imported and mapped (and after having made sure all the data is imported and nothing is obscured), either an empty or demo report can be chosen and further modified. A simplified demonstration of creating a graph in EazyBI is illustrated in Figure 3.

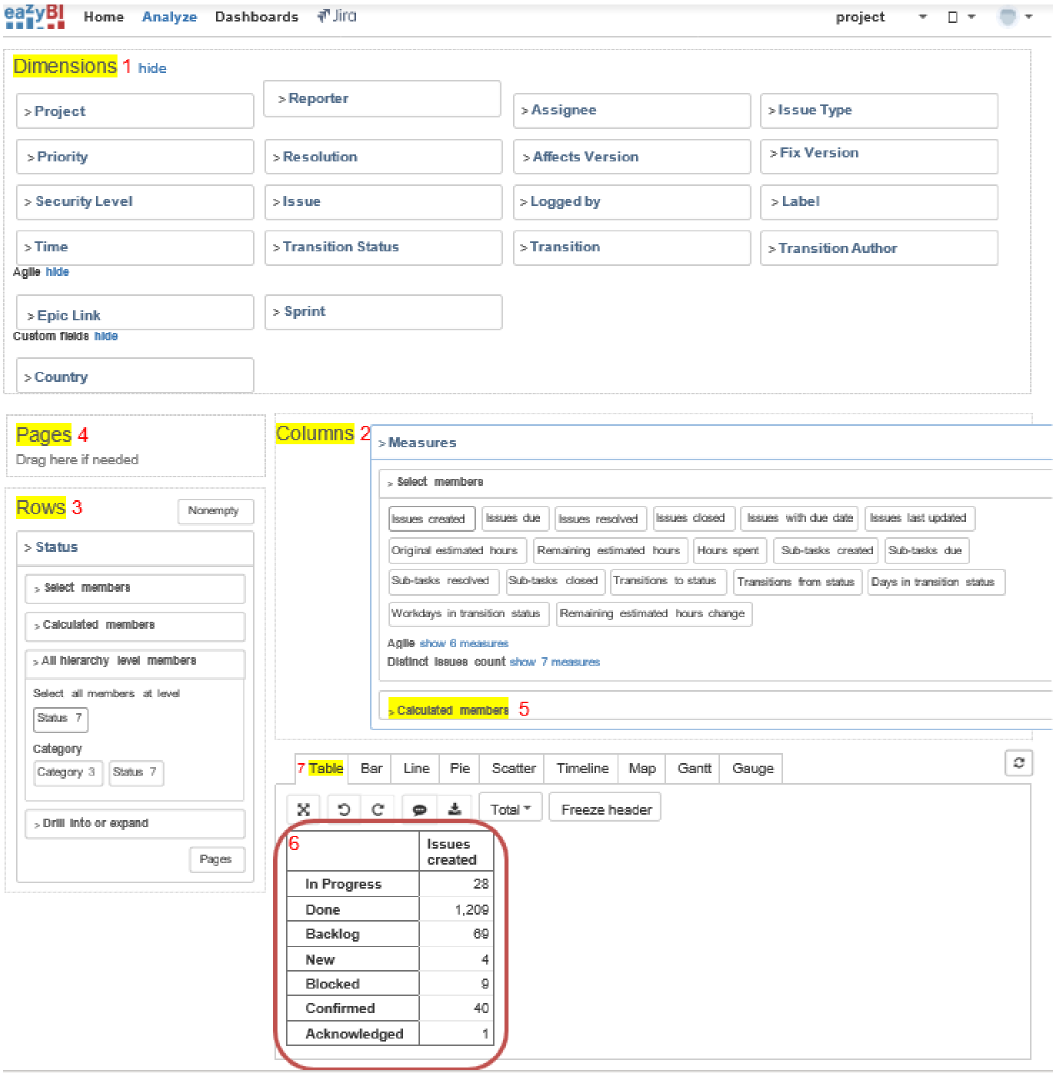

Figure 3. Example of a simple EazyBI graph.

After an empty or demo graph has been opened, the process of either modifying or creating a new graph looks akin to the example in Figure 3. The dimensions (1) section shows all the dimensions stored in the data cube with all the further hierarchy levels contained in them. The relevant dimensions are to be moved to either the Columns (2), Rows (3), or Pages (4) sections (x-axis, y-axis, and filter respectively) using the drag-and-drop functionality.

After choosing dimensions they can be opened to show all the members and properties of an individual dimension. After choosing the needed members or creating new calculated members (5), the members are shown in the resulting graph (6). There can be multiple dimensions in each section and multiple members can be chosen from each dimension, with the order in which they are chosen being the order in which they appear.

After having chosen the data, the type of graph can be chosen (7) from the list, with some graph types having further options (Timeline: line, spline, column, area). There are options to modify the data shown by for example hiding empty rows/columns, getting total values, changing values to percentages, or even calculating new values based on the ones shown.

The last thing that must be done is saving and naming the graph. If copying and modifying an existing report, saving it with a new name creates a separate graph instead of overwriting the previous one.

The table in Figure 3 has the dimension “Measures” in the Column section with the “Issues created” member chosen and the “Status” dimension in the Rows with all members at level “Status” having been chosen. This results in a table with all “Status” options in individual rows with a column for “Issues created” showing how many issues have been created grouped by the status. To take it further, the dimension “Time” could be moved into the “Pages” sections, which would result in a drop-down menu with the chosen “Time” members, which would filter the whole graph.

5.2 Graph Types

The decision of which graph style to use should be done once the data is known and what specifically needs to be shown. A graph should be insightful and only display relevant data in a way that does not oversimplify the data, consequently making it untruthful. EazyBI gives nine different options of graph types to choose from, though most can be further modified or combined. These options are *table, bar, line, pie, scatter, timeline, map, Gantt*, and *gauge* with for example the *timeline* graph having further options such as *line, spline, column,* or *area* timeline [25].

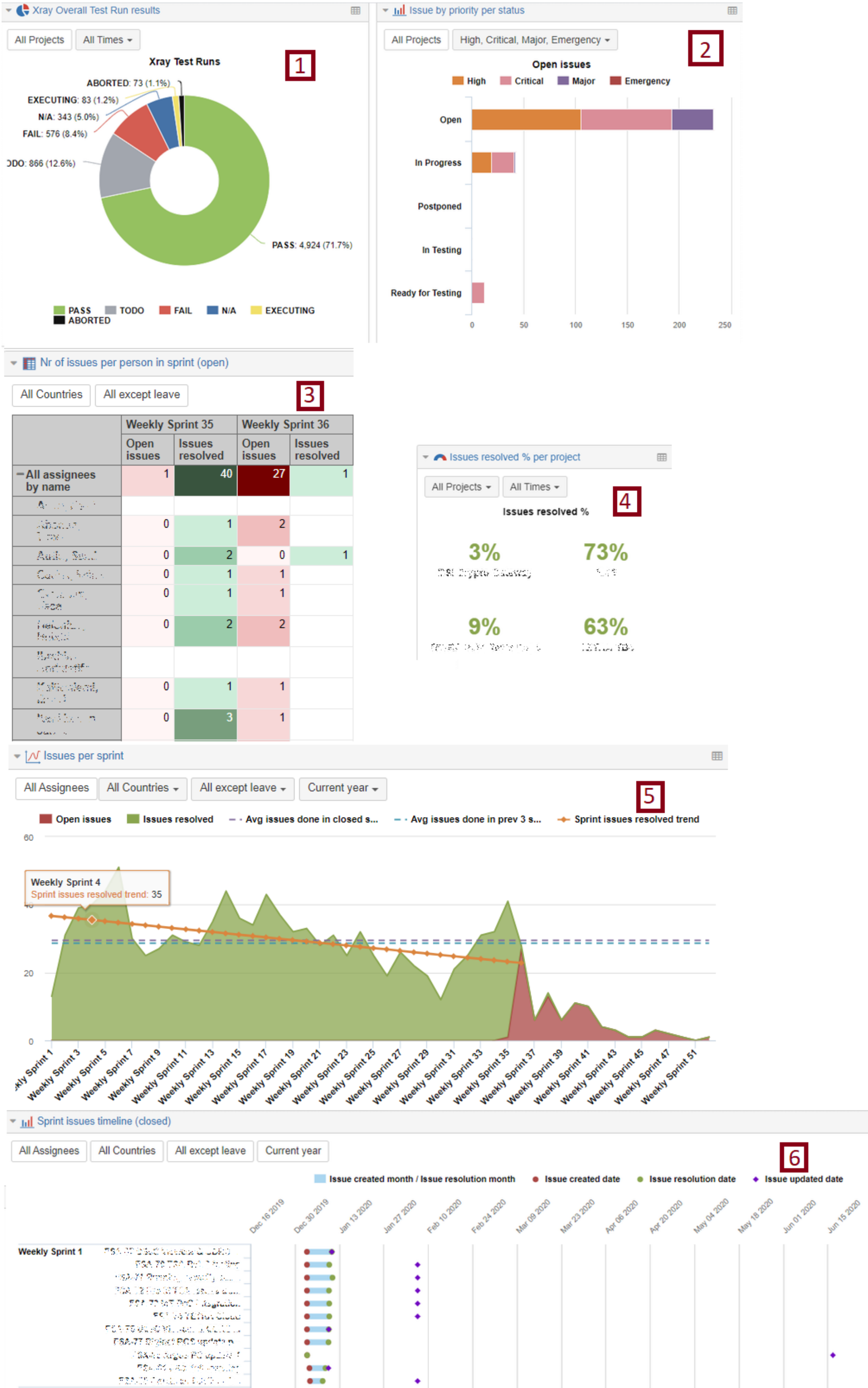

Figure 4. Example of EazyBI graph types.

Figure 4 shows examples of typical graph types made with EazyBI. Some of information has been redacted because of sensitive data. The first graph is a pie chart with each section separated by test run results, and the second graph shows issues by priority per status as a stacked bar chart with a flipped axis. The third graph is a simple table displaying both open and resolved issues per person in a sprint and colored accordingly. The fourth graph is a gauge chart showing issues resolved per project, changed to show percentages instead of the typical gauge. The fifth graph is a timeline displaying the number of issues with the sprints being the measure of time. The issues are separated by their status and stacked as an area chart, with the added trend and average lines. Lastly, the sixth graph is a Gantt chart showing the dates when an issue was created, resolved, and possibly updated on a timeline categorized for each sprint.

There are plenty of other visual changes that can also be modified through EazyBI that can provide a more functional outcome. Other than modifying or filtering the data, for example, changes can be done by swapping the axis, font sizes, altering colors, or modifying the scale.

5.3 MDX Calculations

MDX calculations are used to get further information out of the data imported to the data cube in EazyBI [26]. This is done by creating new calculated members into existing dimensions for example by defining a new calculated member “Profit” to be the cost amount measure subtracted from the sales amount measure.

Define calculated member formula

[Measures].[Issues resolved %] =

```
CASE WHEN
  [Measures].[Issues created] > 0
THEN
  [Measures].[Issues resolved]
  /
  [Measures].[Issues created]
END
```

Formatting: ##% Integer percentage

Listing 1. Example of a calculated member formula definition.

Listing 1 shows an example of a calculated member formula definition in EazyBI and what it looks like when creating one. The code takes the dimension “Measures” and creates a new calculated member “Issues resolved %”. It is a simple case structure where it first takes an existing member “Issues created” from the same dimension and checks that it is greater than 0. If it is, the calculation of “Issues resolved” divided by “Issues created” is executed with the result being the value of the new calculated member. The result is also specified to be formatted as an integer percentage.

```
[Measures].[ Project predicted date ] =
Cache(
Case when
-- resolved
  ([Measures].[Issues resolved],
   [Time].CurrentHierarchy.DefaultMember) > 0
   AND
-- unresolved issues
   ([Measures].[Open issues],
    [Time].CurrentHierarchy.DefaultMember) > 0
-- project has started
   AND NOT IsEmpty(
    [Measures].[Project start date])

Then
  DateAddDays('Today',
  -- add days to reach this amount from Now
  -- (in how many days the currently open issues will be resolved
  -- based on previously resolved issues from project start till today)
  Cache(
  -- all currently open issues
    ([Measures].[Open issues],
     [Time].CurrentHierarchy.DefaultMember)
      /
  -- divided by days performance
    Cache(
      -- currently resolved issues
       ([Measures].[Issues resolved],
        [Time].CurrentHierarchy.DefaultMember)
      /
      -- days till today from project start
      DateDiffDays(
       [Measures].[Project start date],
       'Today')
    )
  ) + 1
)
End)
Formatting  mmm dd yyyy Month Day Year
```

Listing 2. Calculated member formula: project predicted date.

Listing 2 shows more complex MDX code that defines a new calculated member, “Project predicted date”, for the dimension “Measures”. First, the code checks that the members “Issues resolved” and “Open issues” in the current hierarchy of the “Time” dimension have a value greater than zero, and that the measure “Project start date” is not empty. Then it adds days to reach this amount from “Today” and sums all the currently open issues and divides it by the day’s performance, which in turn is calculated by dividing currently resolved issues by the number of days to today from the start of the project.

The result is in the format Month Day Year and tells in how many days the currently open issues will be resolved based on previously resolved issues in the project from its start until today.

5.4 Graph Publication

There are several ways a graph made in EazyBI can be published. They can be single graphs and charts or grouped into dashboards. Different ways to publish these graphs result in possibly different views, updating possibilities, and varying restrictions.

With EazyBI used as an Atlassian plug-in for Jira, a straightforward way to publish the graphs is using the EazyBI macro gadget provided in both Jira and Confluence for both single graphs and whole dashboards. These macro gadgets allow the viewer to also modify the graph view and save that in a single Confluence page without changing the graph from EazyBI.

If wanting to publish somewhere else, an embedded iframe can be used. Embedding the graph into an iframe can be done by copying the iframe code from the relevant EazyBI graph page and pasting it into a Hypertext Markup Language (HTML) code block. This can also be done on a Confluence page with benefits such as a longer loading time before giving an error message and the possibility of modifying the EazyBI graph through Java code applied to the Confluence page.

The graph data can also be downloaded straight from EazyBI as a CSV or XLS (for tables) or CSV, XLS, PNG, or PDF (for graphs). Another option is also to simply share the link straight to EazyBI.

When viewing an EazyBI report once it is published, it is possible to:

- View it as a table (by clicking on the table icon in the top right corner of the graph)
- See the last data import time (by hovering over the top right corner of the graph)
- Filter through the data from drop-down menus (if the creator of the graph has enabled this)
- Set and reset the scale (by clicking and dragging the axis)

- Set and reset zoom (by clicking and dragging the cursor in a data timeline)
- Drill across the data (by clicking on the key)
- Drill through the issues to get the relevant Jira links (by clicking on the data)
- Order and download the data (if the creator of the graph has enabled this)

Making any of the mentioned modifications does not save the changes, which means after changes are done and the page is refreshed, the page will display the graphs in their original form. Changes can only be done through EazyBI, though how the graph is displayed in a Confluence page can be changed through the EazyBI macro gadget. This can be done by adding the graph to Confluence with the gadget, modifying the view, and saving it. After this, the graph is displayed differently than in EazyBI, but only in that specific macro.

5.5 JavaScript Modification

If an EazyBI graph is published onto a page using an iframe, added JavaScript code can further modify what the graph looks.

Weekly allocation combined issues with hidden labels

Assignee	Weekly Sprint 40 Issue ids and labels	Weekly Sprint 41 Issue ids and labels	Weekly Sprint 42 Issue ids and labels	Weekly Sprint 43 Issue ids and labels	Weekly Sprint 44 Issue ids and labels	Weekly Sprint 45 Issue ids and labels
[illegible]	FSA-[illegible] (onsite) FSA-[illegible] (onsite)	FSA-[illegible] (onsite) FSA-[illegible] (onsite)	FSA-[illegible] (development)	FSA-[illegible] (development)	FSA-[illegible] (development)	FSA-[illegible] (development)
[illegible]	FSA-[illegible] (testing) FSA-[illegible] (testing) FSA-[illegible] (testing) FSA-[illegible] (testing) FSA-[illegible] (testing)	FSA-[illegible] (testing) FSA-[illegible] (testing) FSA-[illegible] (testing)	FSA-[illegible] (swdelivery) BACKLOG			
[illegible]		FSA-[illegible] (onsite)	FSA-[illegible] (onsite)	FSA-[illegible] (onsite)		FSA-[illegible] (onsite)
[illegible]	FSA-[illegible] (onsite) FSA-[illegible] (onsite)	FSA-[illegible] (testing) FSA-[illegible]	FSA-[illegible] (leave)	FSA-[illegible] (remote)	FSA-[illegible] (testing)	FSA-[illegible] (testing)
[illegible]	FSA-[illegible] (swdelivery)	FSA-[illegible] (swdelivery)	FSA-[illegible] (swdelivery)	FSA-[illegible] (swdelivery)	FSA-[illegible] (swdelivery)	FSA-[illegible] (leave)
[illegible]						
[illegible]	FSA-[illegible] (testing) FSA-[illegible] (training) FSA-[illegible] (swdelivery)	FSA-[illegible] (testing) FSA-[illegible] (swdelivery)	FSA-[illegible] (testing)	FSA-[illegible] (testing)	FSA-[illegible] (testing) FSA-[illegible] (training)	FSA-[illegible] (testing) FSA-[illegible] (remote)

Figure 5. EazyBI table colored using JavaScript.

Some information in Figure 5 has been redacted because of sensitive data but issue identifications and labels per assignee for each week are shown. It is an EazyBI table that has been published in a Confluence page with an iframe and colored using the JavaScript code in Listing 3 which was added to the same Confluence page in an HTML code block.

```
<script type="text/javascript">

<!-- needs "async" in function that sleep is called upon -->
function sleep(ms) {
  return new Promise(resolve => setTimeout(resolve, ms));
}

//HIGHLIGHTING BASED ON ACTIVITY TYPE (LABEL)
function words() {
     $.each($("#gadget-4814768635904").contents().find('td.string_cell'),
function(){
          // It's a big table, make font smaller
          $(this).css("fontSize", "70%");
         // The inside item background will be replaced afterwards with another
color
          var actionBlock = $(this).text().replace(/(FSA-.*?BACKLOG)/g,"<div
style='background-color: #e6d967; border: 2px solid #e6d967; padding:
0px'>$1</div>");
          actionBlock = actionBlock.replace(/(FSA-.*?BLOCKED)/g,"<div
style='background-color: #c7402c; border: 2px solid #c7402c; padding:
0px'>$1</div>");
          actionBlock = actionBlock.replace(/(FSA-.*?\(leave\))/g,"<div
style='background-color: #6babff; padding: 0px'>$1</div>");
          actionBlock = actionBlock.replace(/(FSA-.*?\(Training\))/ig,"<div
style='background-color: #bd9191; padding: 0px'>$1</div>");
          actionBlock = actionBlock.replace(/(FSA-.*?\(onsite\))/ig,"<div
style='background-color: #339966; padding: 0px'>$1</div>");
          actionBlock = actionBlock.replace(/(FSA-.*?\(remote\))/ig,"<div
style='background-color: #9bc799; padding: 0px'>$1</div>");
          actionBlock = actionBlock.replace(/(FSA-.*?\(preparations\))/ig,"<div
style='background-color: #4d6b53; padding: 0px'>$1</div>");
          // Turn IDs into links
          actionBlock = actionBlock.replace(/(FSA-[\d]*)/g,"<a href='https://slc-
toolset.common.airbusds.corp/jira/browse/$1'>$1</a>");
          $(this).html(actionBlock);
     });
}

// Does not seem to work for everyone
$(window).on("load", async function() {
     await sleep(2000);
     //highlight();
     words();
});

$(document).on("scroll", async function() {
     await sleep(500);
     //highlight();
     words();
});
</script>
```

Listing 3. Iframe code for coloring an EazyBI table.

The JavaScript code in Listing 3 is applied to the page when loaded and when scrolling, which is specified at the end of the code. It also uses a sleep function specified in the beginning so that the code is not constantly running. The function “words()” is the highlighting part of the code which works by finding specific words like “leave” and replacing the cell in which it appears in the table with a new color and applying these changes to the specified EazyBI graph “#gadget-4814768635904”.

6 Reporting Project Execution for the Testing Teams

The illustrated case is about creating automated graphs for Airbus Defence and Space based on the project data accumulated into EazyBI from Jira. This was done with the goal to provide summarized and easily understandable data graphs for different projects to be able to see the progress and possible problem areas and to illustrate how to overcome the challenges of data visualization.

This project began by going through the different tools and methods used in the case: EazyBI, Jira, Confluence, and MDX. Then the project continued with going through the types of data used and received in each project and agreeing on the KPIs to focus on with the project managers.

These main KPIs agreed upon were the defect trend and test execution status. Both had additional requirements of traceability, data accuracy, versatility for use in multiple projects, ease of use, longevity, and a display of great data visualization qualities.

As a result, these new data report graphs should provide necessary information about the progress of each project concisely which meets the needs of the company. The data that can be presented in the new graphs should be more in-depth and dynamically visualized, therefore providing more information than before.

6.1 Graph Execution

This section provides an overview of the execution of the two main EazyBI graphs that were made to show the KPIs.

6.1.1 Defect Trend Graph

The first KPI is shown in the defect trend graph, which sums up all the defects of a particular project per week divided by severity (fatal, severe, low). A request was for it to also contain a limit for fatal and severe type defects, which were to be shown as a stacked area, and for low type defects to be a separate line.

To create this graph, the dimensions *Issue Type*, *Measures*, and *Severity* were needed in *Columns*. *Project*, and *Affects Version* dimensions were needed in the *Pages* section, and *Time* was included to both *Pages* and *Rows*. The *Issue Type* dimension was used to select only *Defect Reports* from the data. *Measures* was used to select *Open Issues* and to create the limit line. *Severity* was used to divide these issues by their severity status. *Rows* held *Time* and inside the dimension *Week* was selected on the *Weekly* hierarchy level to display the data per week, even though it was a timeline.

The pages allowed the graph to be more versatile because adding dimensions into the *Pages* section creates drop-down menus for the user to further parse through the data after it has been published. For this graph the project, version, and time could be chosen, with the defaults being *All Projects*, *All Versions*, and *All Times* shown.

The calculated member formula for *Open Issues* which was created into the *Measures* dimension is shown in Listing 4.

```
[Measures].[Open issues]=
CASE WHEN [Issue].CurrentMember.Level.Name <> 'Issue' THEN
  Cache(
    NonZero(Sum(PreviousPeriods([Time].CurrentHierarchyMember),
      Cache([Measures].[Issues created]
          - [Measures].[Issues resolved])
    ))
    + [Measures].[Issues created]
    - [Measures].[Issues resolved]
  )
WHEN [Time].CurrentHierarchyMember IS [Time].CurrentHierarchy.DefaultMember
THEN NonZero([Measures].[Issues due])
ELSE
  -- optimized formula for drill through Issue
  NonZero(IIF(
      DateBeforePeriodEnd(
        [Issue].CurrentMember.get('Created at'),
        [Time].CurrentHierarchyMember) AND
      NOT DateBeforePeriodEnd(
        [Issue].CurrentMember.get('Resolved at'),
        [Time].CurrentHierarchyMember),
    ([Time].CurrentHierarchy.DefaultMember,
      [Measures].[Issues created]),
    0
  ))
END

Formatting: #,### Integer
```

Listing 4. Calculated member formula: open issues.

A straightforward way of calculating open issues is subtracting issues that have been resolved from issues that have been created, but other checks for the time were also added to get accurate data.

After having all the data and calculated members, the graph was formatted into a stacked timeline showing values instead of percentages. Empty columns were hidden, and the scale was changed for this project from the axes option so that the default for the Y-axis has a step value of 10.

The defect trend graph provides an awareness of the relevant KPI data giving an insightful look into progress regarding the defects because it visualizes the trend. It does not oversimplify the data or leave any relevant data overlooked.

6.1.2 Cumulative Test Execution Status

The second KPI is shown in the cumulative test execution status graph, which sums up the test results of a particular project per week divided by the execution status (pass, fail, executing, to-do, and N/A).

To create this graph, the dimensions *Measures* and *Xray Test Execution Status* were used in *Columns.* The *Xray Test Plan* dimension was used in the *Pages* section, and *Time* was used in *Rows.*

In the columns, *Xray Test Execution Status* was used to divide the issues by their status. A new calculated member was created into *Measures* called *Cumulative Xray Tests executed by status* and it is the only measure chosen from the *Measure* dimension. The calculated member function is shown in Listing 5.

```
[Measures].[Cumulative Xray Tests executed by status]=
Cache(Sum(
      Desscendants([Xray Test].CurrentMember, [Xray Test].[Test]),
      CASE WHEN
            [Measures].[Cumulative Xray Tests executed] > 0
      THEN
            CASE WHEN
                  [Xray Test Execution Status].CurrentMember.Level.Name = "Status"
            THEN
                  CASE WHEN
                        Tail(Order([Xray Test Execution Status].Members,
                        [Measures].[Xray Test last status date])).Item(0) IS
                        [Xray Test Execution Status].CurrentMember
                  THEN
                  1
                  END
            ELSE
            1
            END
      END
      )
)
Formatting: default
```

Listing 5. Calculated member functions: cumulative xray tests executed by status.

The calculated member *Cumulative Xray Tests executed by status* shown in Listing 5 takes the descendants of the current member test, checks that they are greater than 0 and that it is in the hierarchical level *Status,* and sums up.

The only thing in *Rows* is the *Time* dimension which was used to display the data for only the last 6 months or the last 10 weeks in some graph version depending on the project. A new calculated member was used, instead of simply choosing the months so that the graph would modify itself without the need of someone to change the time. This simple piece of MDX code is shown in Listing 6.

```
[Time].[Last 6 months]=
Aggregate(
      [Time].[Month].DateMembersBetween('6 months ago', 'today')
)
Formatting: default
```

Listing 6. Calculated member function: the last 6 months.

The dimension *Xray Test Plan* is the only thing in *Pages* allowing the viewer to select the project they want to be displayed in the graph, with the default being *All Xray Test Plans*.

The formatting of the graph was made to be a timeline displayed as a stacked and vertical bar graph. Another version of the final graph was also made showing the percentages of tests with *Columns* having the *Measure* dimension with new calculated measures for counting the percentages for passed, failed, and to-do tests with the formatting being ##.##% decimal percentages.

The cumulative test execution status graph provides an awareness to the relevant KPI data giving an insightful look into the state of test execution. This is done in a functionally informative manner by dividing the data by status.

6.2 Outcome and Use

The graphs made for Airbus Defence and Space regarding the main KPIs are used in various projects' weekly flash reports, summarizing the status of the projects. The graphs were to be quickly understandable, accurate, and showing the main KPIs, thus also showing possible problem areas. Because the same graphs are used in various projects, they need to be dynamically visualized with options for modifications to meet different projects' slightly varying needs.

The first EazyBI graph detailed in this thesis is the defect trend graph shown in Figure 6, and the second graph for cumulative test execution status is shown in Figure 7.

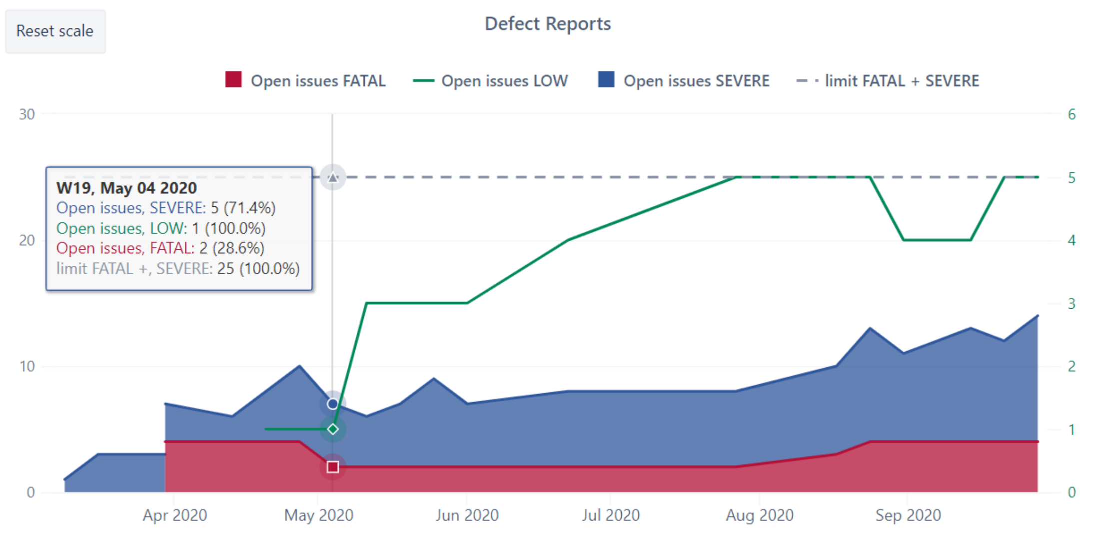

Figure 6. Defect trend graph.

The defect trend graph shows all open issues which are labeled as defects and separated by the severity. Fatal and Severe severities have been stacked, in order not to obscure any data, and a dashed line is used as a limit line. Low severity issues have been added as a separate line with its own axis since there tend to be fewer low severity issues in the projects using the graph. The way it is done in this case works, but there are cases where it might obscure the proportionality of the figures making it harder to understand correctly. Figure 6 shows a text box highlighting week 19. It also shows the exact number of issues during that week and by clicking on the data the user is also able to see the linked data providing much needed functionality to these graphs.

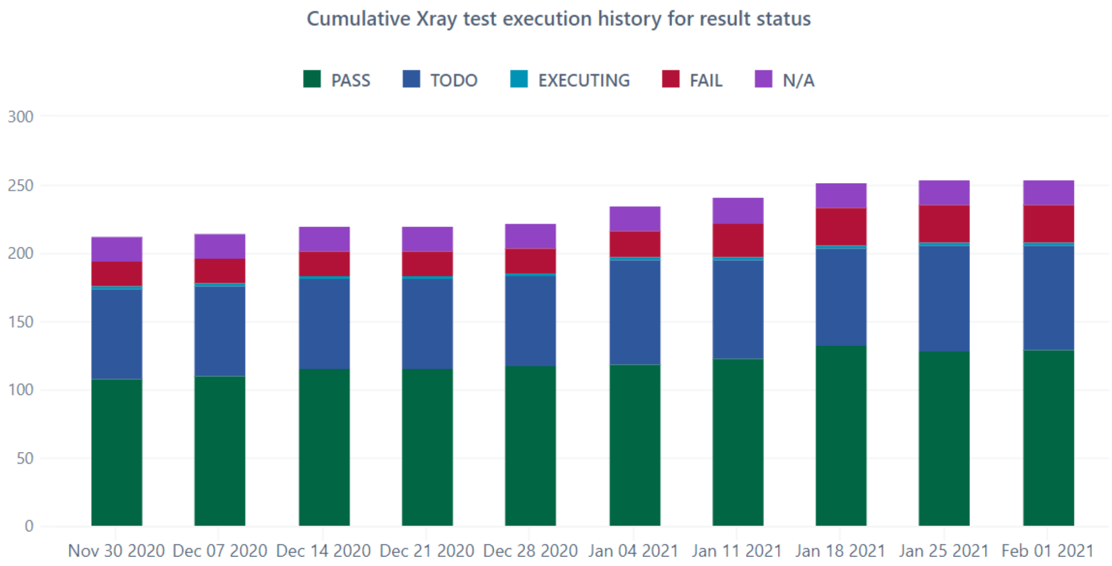

Figure 7. Cumulative test execution status graph.

The cumulative test execution status graph in Figure 7 shows all test executions divided by the status and stacked as bars per week in the allocated time frame which is 10 weeks. The amounts are cumulative meaning that an issue made in for example week 1 is also counted in the number of issues in week 2, though the status can change. It displays the data in a concise manner without sacrificing functionality.

Both graphs are made to automatically update every 15 minutes with data gotten from Jira and can be switched to show the same data as a table as shown in Table 1.

Table 1. Cumulative xray test execution table.

		Cumulative Xray test execution history for result status				
		PASS	TODO	EXECUTING	FAIL	N/A
− Last 10 weeks	+ W49, Nov 30 2020	109	66	2	18	18
	+ W50, Dec 07 2020	111	66	2	18	18
	+ W51, Dec 14 2020	116	66	2	18	18
	+ W52, Dec 21 2020	116	66	2	18	18
	+ W53, Dec 28 2020	118	66	2	18	18
	+ W01, Jan 04 2021	119	77	2	19	18
	+ W02, Jan 11 2021	123	73	2	24	20
	+ W03, Jan 18 2021	133	71	2	28	18
	+ W04, Jan 25 2021	129	77	2	28	18
	+ W05, Feb 01 2021	130	76	2	28	18
All others						

Table 1 shows the same data as the cumulative test execution status graph in Figure 7 and the contrast between these two illustrates exactly why the graphs are helpful: seeing the numbers as visualized bars is a quicker way to see the progress of the said project.

6.3 Challenges and Future Development

Challenges encountered during the data visualization for this case were mainly related to data, changes, or EazyBI functionalities. Handling all the data while keeping it accurate and making sure everyone involved understood what is being displayed was an integral part of the process of creating graphs. Constantly evolving and changing graphs based on current needs and observations made this harder.

EazyBI version 6.0.1 was used in this project mainly due to the more detailed error messages, and processing power, though some problems did arise further into the project. EazyBI errors seen in the process of making these projects varied quite a bit but mainly regarded the sheer amount of data that needed to be queried for these graphs, which resulted in slower processing.

Even with some challenges, the main KPI graphs were completed. Though these graphs were completed, there are plenty of potential future developments that could be made. There are new graphs that are already being made, and the possibility of branching these graphs for other teams is also a possible endeavor, though it is important to keep the qualities of great data visualization in mind.

When it comes to updates on the completed graphs, they will need to be updated depending on the needs of the project, changes in the project data, changes in any data sources, and possible changes if there is a significant increase in data to keep it relatively fast.

7 Conclusion

The purpose of this thesis was to create automated EazyBI graphs for Airbus Defence and Space based on provided project data. The goal was accomplished by creating summarized and easily understandable data graphs for different project groups to be able to see the KPIs and possible problem areas.

The data received for each project was imported to EazyBI from the project tracking software Jira. The data was then processed and formatted in EazyBI before publishing in several ways and formats in the Confluence workspace tool. As a result, these new data report graphs provide necessary information about the progress of each project, therefore meeting the requirements of the project the company has set. The data that can be presented in the new graphs are more in-depth and dynamically visualized, therefore providing more information than before when data visualization was not used.

Visualized data graphs were needed to provide an easy but comprehensive view of the data since results can often go unnoticed or be overlooked if shown in a more traditional, non-visualized manner.

Though data visualization faces some challenges, business intelligence tools are advantageous, and the case of this project illustrates how to overcome the difficulties of data visualization, such as the oversimplification of data, and overreliance on visuals. The examples provided also explain the different qualities of great data visualization including truthfulness, functionality, insightfulness, and enlightening quality.

The graphs are in use by Airbus in multiple projects, allowing the people involved to be able to see the KPIs such as the status, defects, and tests executed by the status. These graphs can be used flexibly allowing for further data growth, updates, and the input of the viewer. Based on this project it should also be possible to create further graphs for different teams in the company.

References

1. Unwin, Antony. Why Is Data Visualization Important? [e-journal]. HDSR, issue 2.1; 2020.
   URL: https://hdsr.mitpress.mit.edu/pub/zok97i7p/release/3. Accessed 8 March 2021.
2. Cairo, Alberto. The Truthful Art: Data, Charts, and Maps for Communication [e-book]. New Riders, Pearson Education; 2016.
   URL: https://drive.google.com/file/d/0B0IAplzpBxwfeEVMdDZrNm5mbUU/view. Accessed 24 February 2021.
3. Kirkland, Frazar. Cyclopedia of Commercial and Business Anecdote [e-book]. New York: D. Appleton and Company; 1868.
   URL: https://books.google.mk/books?id=vqBDAAAAIAAJ&printsec=frontcover#v=onepage&q&f=false. Accessed 27 October 2020.
4. Luhn, Hans P. A Business Intelligence System [e-journal]. IBM Journal; 1958.
   URL: https://web.archive.org/web/20080913121526/http://www.research.ibm.com/journal/rd/024/ibmrd0204H.pdf. Accessed 27 October 2020.
5. What Is Business Analytics? [online]. NDMU; 2018.
   URL: https://online.ndm.edu/news/analytics/what-is-business-analytics/#:~:text=The%20first%20formalized%20system%20of,that%20ultimately%20boosted%20industrial%20production. Accessed 18 March 2021.
6. Barua, Anitesh; Mani, Deepa; Mukherjee, Rajiv. Measuring the Business Impacts of Effective Data [PDF]. The University of Texas; 2010.
   URL: http://www.datascienceassn.org/sites/default/files/Measuring%20Business%20Impacts%20of%20Effective%20Data%20I.pdf. Accessed 26 October 2020.
7. What Is Business Intelligence? [online]. Tableau; 2021.
   URL: https://www.tableau.com/learn/articles/business-intelligence. Accessed 18 March 2021.
8. Why Power BI [online]. Microsoft; 2020.
   URL: https://powerbi.microsoft.com/en-us/why-power-bi/. Accessed 27 October 2020.
9. Qlik [online]. Qlik; 2021.
   URL: https://www.qlik.com/us/. Accessed 18 March 2021.

10. Data Studio Help [online]. Google Data Studio; 2021.
URL: https://support.google.com/datastudio/answer/6283323?hl=en.
Accessed 18 March 2021.
11. IBM Cognos Analytics [online]. IBM; 2021.
URL: https://www.ibm.com/products/cognos-analytics. Accessed 18 March 2021.
12. EazyBI [online]. EazyBI; 2020.
URL: https://eazybi.com/. Accessed 27 October 2020.
13. Waas, Florian; Wrembel, Robert; Freudenreich, Tobias; Thiele, Maik; Koncilia, Christian; Furtado, Pedro. On-Demand ELT Architecture for Right-Time BI: Extending the Vision [e-journal]. International Journal of Data Warehousing and Mining, pp. 21-38; 2013.
URL: https://www.igi-global.com/gateway/article/78285. Accessed 24 February 2021.
14. Data Mining [online]. SAS; 2020.
URL: https://www.sas.com/fi_fi/insights/analytics/data-mining.html. Accessed 27 October 2020.
15. Joshi, Amit M. Data Analytics & Artificial Intelligence: What It Means for Your Business and Society [online]. IMD; 2018.
URL: https://www.imd.org/research-knowledge/articles/artificial-intelligence-real-world-impact-on-business-and-society/. Accessed 24 February 2021.
16. Williams, Chris. Artificial Intelligence for Data Analytics (AIDA) [Research project]. The Alan Turing Institute; 2020.
URL: https://www.turing.ac.uk/research/research-projects/artificial-intelligence-data-analytics-aida. Accessed 24 February 2021.
17. Deliver faster with seamless collaboration [online]. Atlassian; 2021.
URL: https://www.atlassian.com/software/confluence/jira-integration. Accessed 24 February 2021.
18. Hyde, Julian. Layers of a Mondrian System [online]. Pentaho; 2006.
URL: https://mondrian.pentaho.com/documentation/architecture.php. Accessed 15 January 2021.
19. Main Concepts [online]. EazyBI; 2021.
URL: https://docs.eazybi.com/eazybi/getting-started/main-concepts. Accessed 15 January 2021.
20. Noyes, Jane; Booth, Andrew; Moore, Graham; Flemming, Kate; Tuncalp, Özge; Shakibazadeh, Elham. Synthesising Quantitative and Qualitative Evidence to

Inform Guidelines on Complex Interventions: Clarifying the Purposes, Designs and Outlining Some Methods. BMJ Global Health; 2019. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6350750/. Accessed 24 February 2021.

21. Data Import [online]. EazyBI; 2021. URL: https://docs.eazybi.com/eazybi/data-import. Accessed 15 January 2021.
22. Data Mapping [online]. EazyBI; 2021. URL: https://docs.eazybi.com/eazybi/data-import/data-mapping. Accessed 15 January 2021.
23. Mathrani, Sanjay. Critical Business Intelligence Practices to Create Meta-knowledge [e-journal]. Business Information Systems, Vol. 36, No. 1, 2021. URL: http://www.inderscience.com/storage/f911184121056327.pdf. Accessed 24 February 2021.
24. Paul, Ehrlich. Data Analytics [e-journal]. Troy Vol. 31, issue 6; 2014. URL: https://search-proquest-com.ezproxy.metroplia.fi/docview/1535657904?pq-origsite=primo. Accessed 24 February 2021.
25. Analyze and Visualize: Create Charts [online]. EazyBI Documentation; 2021. URL: https://docs.eazybi.com/eazybi/analyze-and-visualize/create-charts. Accessed 24 February 2021.
26. Analyze and Visualize: Calculated Measures and Members [online]. EazyBI Documentation; 2021. URL: https://docs.eazybi.com/eazybi/analyze-and-visualize/calculated-measures-and-members. Accessed 24 February 2021.