BBM475 Management Information Systems
2020 Fall, Term Project

BUSINESS INTELLIGENCE (BI) PROJECT

Delivery Date: 10 Jan 2021

Team Members:
Eren Gülüm 21627265

Mehmet Serkan Tan 21627634

1. PROJECT OVERVIEW

BI Project Purpose : In this project, we tried to bring the large and difficult to understand dataset to a level that can be understood by everyone. We tried to find answers to a number of questions using different methods and techniques. We know that no matter how large the data we have, it can be useful when filtering, associating and visualizing correctly. We supported Business Intelligence processes by converting data into information.

BI Tool Used : Microsoft Power BI

Table 1. Goal(s), questions, and metrics used in BI Project

Goal	Question	BI Feature Used
1.Efficiency in Project Development	(MIS) 1.1 What is the best/most efficient team size for project development?	Dashboards, Charts such as Pie Chart and Bar Chart, Power BI Filtering and Querying features, Grouping
1.Efficiency in Project Development	(MIS) 1.2 Which development methodology has more efficiency/suitability ?	Pivot Tables, Charts, Grouping , Power BI Filtering and Querying features
1.Efficiency in Project Development	(DSS)1.3 Suppose we are working on a M1-sized, Enhancement type project for the communication sector. Our team size is 5-8. What will happen / what will be the impact if we use 3-4 or 9-14 team group size instead of 5-8?	Dashboards, Pivot Tables, Charts, Grouping , Power BI Filtering and Querying features
1.Efficiency in Project Development	(MIS) 1.4. Which Development Techniques can be considered more efficient?	Bar Chart, Pie Chart, 100% Stacked Bar Chart, Power BI Query, and Filtering, Slicers, Matrices
1.Efficiency in Project Development	(MIS) 1.5.How the team experience is related to other metrics and how experience affects efficiency?	Bar Chart, Pie Chart, 100% Stacked Bar Chart, Power BI Query, and Filtering, Slicers, Matrices, Grouping

1.Efficiency in Project Development	(DSS) 1.6 What Would Have Happened or Been the Impact If A Least or Most Experienced Team Were Employed?	Bar Chart, Pie Chart, Power BI Query, and Filtering, Slicers, Grouping

Note: We decided to use the Goal-Question-Metric method. As it is a comprehensive goal, we set different questions for the same goal as two teammates. So we worked on 4 different MIS questions and 2 different DSS questions for one goal.

2. DATASET DESCRIPTION

Dataset Name **:** ISBGS

Link to Data Source **:** -

Purpose of Data Use **:** ISBGS is a dataset that contains data of the various projects. The data set we use includes data from 7,518 different projects. There are many different metrics from technical details to demographic details. Thanks to these metrics, different examinations can be made on software projects and many questions can be answered.

Table 2. Dataset attributes used in BI Project and their descriptions

Attribute	Description
Team Size Group	Categories Max Team Size by into groups to increase number of projects selected, as:
Project Elapsed Time	A metric representing the time passed during the project (Total elapsed time for the project in calendar months)
Speed of Delivery	Functional Size Units per elapsed month calculated as Functional Size / Project Elapsed Time
Manpower Delivery Rate	Functional Size Units per person per elapsed month calculated as Functional Size / Project Elapsed Time * Max Team Size
Defect Density	Defects per 1000 FP calculated as Total Defects Delivered * 1000 / Functional Size Measures the quality of software in terms of defects delivered in unit size of software. It is defined as the number of Defects per 1000 Functional Size Units of delivered software, in the first month of use of the software. It is expressed as Defects per 1000 Functional Points.
Relative Size	For major FSMs, categories the Functional Size (or where blank, the Adjusted Function Points) by relative sizes as: 1. XXS Extra-extra-small => 0 and <10 2. XS Extra-small => 10 and <30 3. S Small => 30 and <100 4. M1 Medium1 => 100 and <300 5. M2 Medium2 => 300 and <1000 6. L Large => 1,000 and < 3,000 7. XL Extra-large => 3,000 and < 9,000 8. XXL Extra-extra-large => 9,000 and < 18,000 9. XXXL Extra-extra-extra-large => 18,000.
Total Defects Delivered	Total number of defects reported in the first month of use of the software. This column shows the total of defects reported (Minor, Major and Extreme). Where no breakdown is available, the single value is shown here.

Development Methodologies	Methodologies used during development. (e.g.: Agile, JAD, Waterfall etc.). These methods have not been recorded as being specific to a particular project activity and therefore may apply to any part of the development lifecycle. For ISBSG purposes a methodology applies to the whole project development process. This is distinct from techniques, which apply to activities within the development process.
Industry Sector	This is a derived field that attempts to summarise Organisation Type of the project into a single value of a defined set.
Year of Project	Year of Project, derived from implementation date (if known), or from other project dates such as: · Project end date · Project start date · Estimated implementation date · Data compilation date
Development Type	This field describes whether the development was a new development, enhancement or re-development.

3. QUESTIONS ANSWERED

Table 3. Questions answered and attributes used in BI Project

Type	Question	Attributes Used	Responsible Student
MIS	1.1. What is the best/most efficient team size for project development?	Project Elapsed Time, Speed of Delivery , Manpower Delivery Rate, Defect Density, Relative Size, Total Defects Delivered	*Eren Gülüm*
MIS	1.2. Which development methodology has more efficiency/suitability ?	Development Methodologies, Speed of Delivery, Project Elapsed Time, Year of Project, Relative Project Size, Industrial Sectors	*Eren Gülüm*
DSS	1.3. Assume that we have a M1-sized ... sector .. type project. Project is developed by .. architecture and the our team size range was 5-8. What would happen/What would be the impact if we have used 3-4 or 9-14 or 15-20 team group size instead of 5-8.	Team Size Group, Project Elapsed Time, Speed of Delivery, Defect Density, Manpower Delivery Rate,Total Defects Delivered	*Eren Gülüm*
MIS	1.4 Which Development Techniques can be considered more efficient?	Development Technique, Elapsed Time, Data Quality rating, UFP rating, Functional Size, Development Methodologies, Software Architecture, Software Defects, Effort	*Mehmet Serkan Tan*
MIS	1.5.How the team experience is related to other metrics and how experience affects efficiency?	IT Team Experience, BA Team Experience,	*Mehmet Serkan Tan*

		Programming Languages, Industries	
DSS	1.6 What Would Have Happened or Been the Impact If A Least or Most Experienced Team Were Employed?	IT Team Experience, BA Team Experience, Project Manager Experience, User Satisfaction Survey, Meet Stated Objectives, UFP Rating	*Mehmet Serkan Tan*

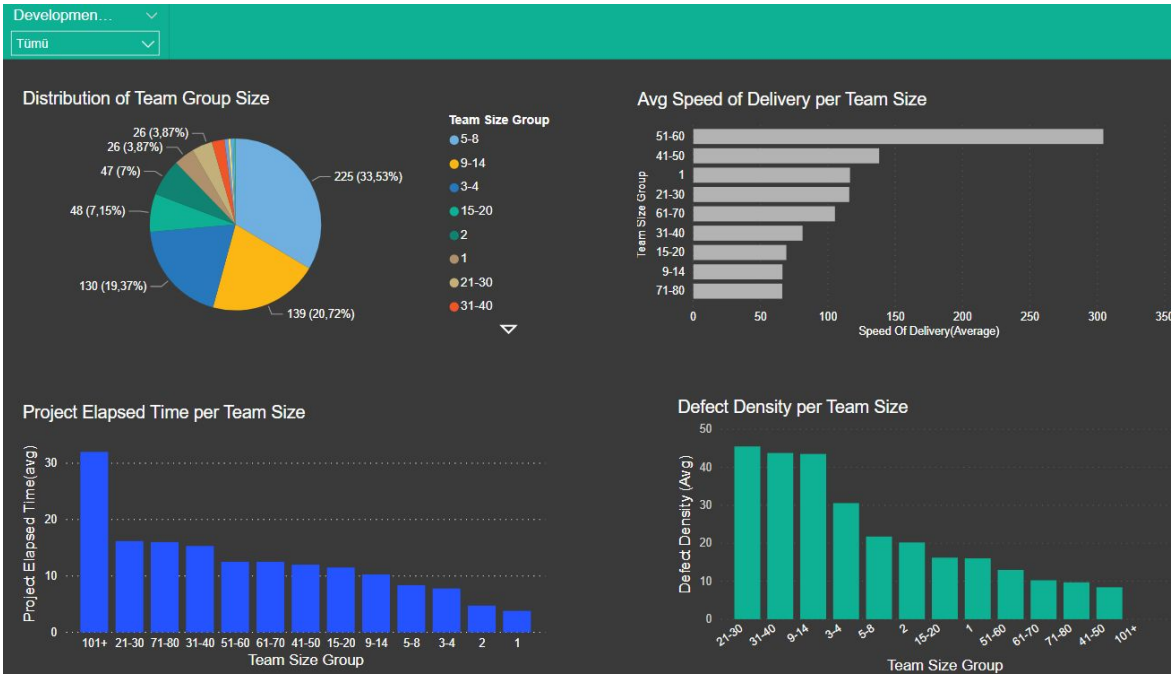
3.1. Questions Answered by Eren Gülüm - 21627265

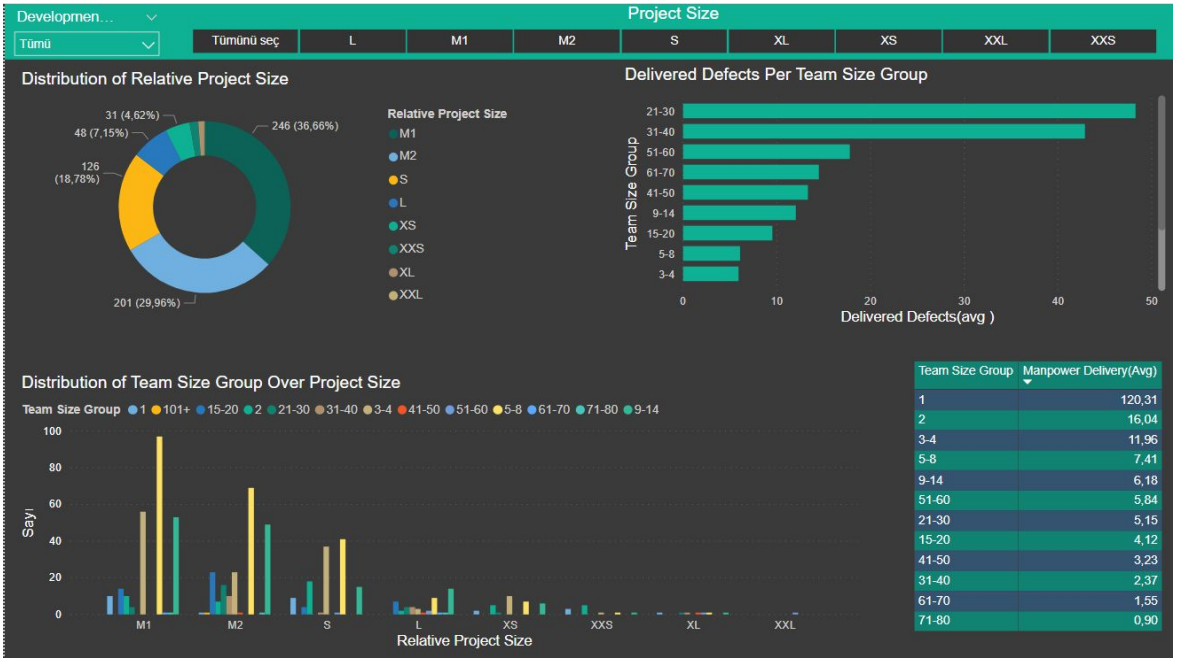
3.1.1. (MIS) Q-1.1: What is the most efficient team size for the project development?

Purpose of question : We think that different team sizes have different effects. So we will try to find out what is the most suitable team size for different type of projects

BI feature used : Dashboards, Pivot Tables, charts such as Pie Chart and Bar Chart, Power BI Filtering and Querying features

BI output description :





When all the graphs are examined, we can see that the most used group size is groups of 5-8 people. However, it seems that the biggest reason for this is the size of the project. For medium-sized projects (M1 and M2), we can assume groups of 5-8 people are in the first place for much more.

In other words, if there is a medium-sized (m1 or m2) project preparation plan, we can say that groups of 5-8 people are preferred by far. In a project of M2 size, if the number of people is insufficient, it can move to a team of 9-14 people.

For a small-scale project (S), groups of 5-8 or 3-4 people seem to be very popular. When we look at the Defects chart and the Manpower Delivery Rate table, we see that groups of 3-4 people give better results. For this reason, it would be better to start with a group of 3-4 people for small-scale projects. If the needs are not met later, a team of 5-8 people can be formed by hiring new employees.

We see that groups of (L) 9-14 people are more popular in large-scale projects. We can say that if a team of 9-14 people exceeds the budget, they can work with a team of 5-8 people (Approximately 20% of L projects have been developed with a team of 5-8 people). In projects where teams of 9-14 people are insufficient, it is seen that teams of 15-20 people have a significant ratio (about 15%). When we examine the Speed of Delivery and Defect Density graphics, we can say that groups of 15-20 people offer faster Delivery and have less defect density.

Implementation details :

To answer the this MIS question, we used 7 different metrics from the data set. Since we have a large and cumbersome data set, we first performed query operations according to this question.

Used dataset metrics:

- Team Size

- Project Size
- Project Elapsed Time
- Number of Defects After Delivery
- Defect Density
- Manpower Delivery Rate
- Speed of Delivery

Instead of using the entire data set to get meaningful results, we decided to separate the part with meaningful and high quality data. Later, we worked on this part. We performed our data preparation processes through the Power BI Query Tool.

First of all, we deleted the records with insufficient quality from the data set in order to reach smoother results (data with ISBSG rating code D was not used). Later, in each of the 7 metrics (columns) we specified, the rows that were empty (null) or incorrect were deleted. After these transactions, we have 671 project data left. We think that 671 pieces of data will be sufficient for this question. Thanks to this query, we have a data set that has 7 metrics and has no null or errors.

Later, this data was grouped according to Team Group Size. A new query was created by taking the number of each group size (in how many projects it is used in) and the average (average) of the other metrics for each metric. Finally, we used visualization tools to implement Power BI graphs and charts.

Please explain in one or two paragraphs how you think your BI output could be used to answers Q-1.1, and how you interpret the BI output:

When all the graphs are examined, we see that groups of 5-8 people are in the first place for much more medium-sized projects (M1 and M2) in the data set. For a small-scale project (S), groups of 5-8 or 3-4 people seem to be very popular.

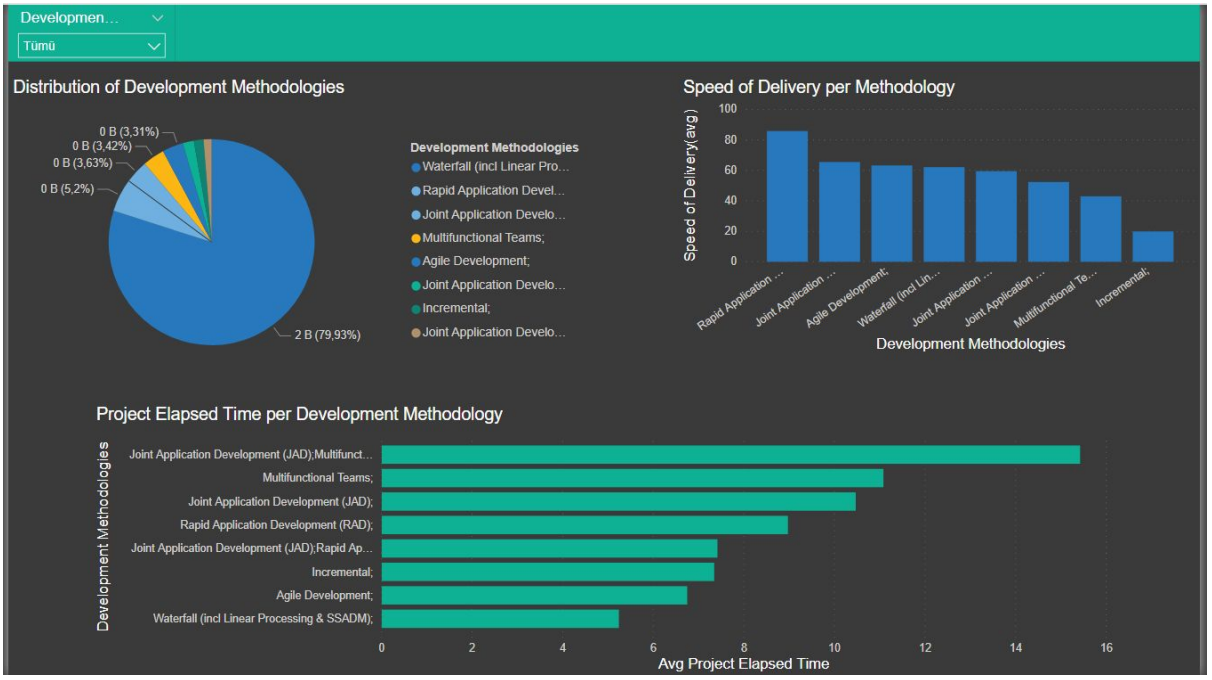
We see that groups of (L) 9-14 people are more popular in large-scale projects. We can say that if a team of 9-14 people exceeds the budget, it can work with a team of 5-8 people. For detailed information and comments, you can review what we wrote in the BI output description section.

3.1.2. (MIS) Q-1.2: Which development methodology has more efficiency/suitability?

Purpose of question :Basically, we need to find out which Development methodology is more efficient. Since the development methodology may vary according to the need and many other situations, we must first examine by specifying certain metrics.

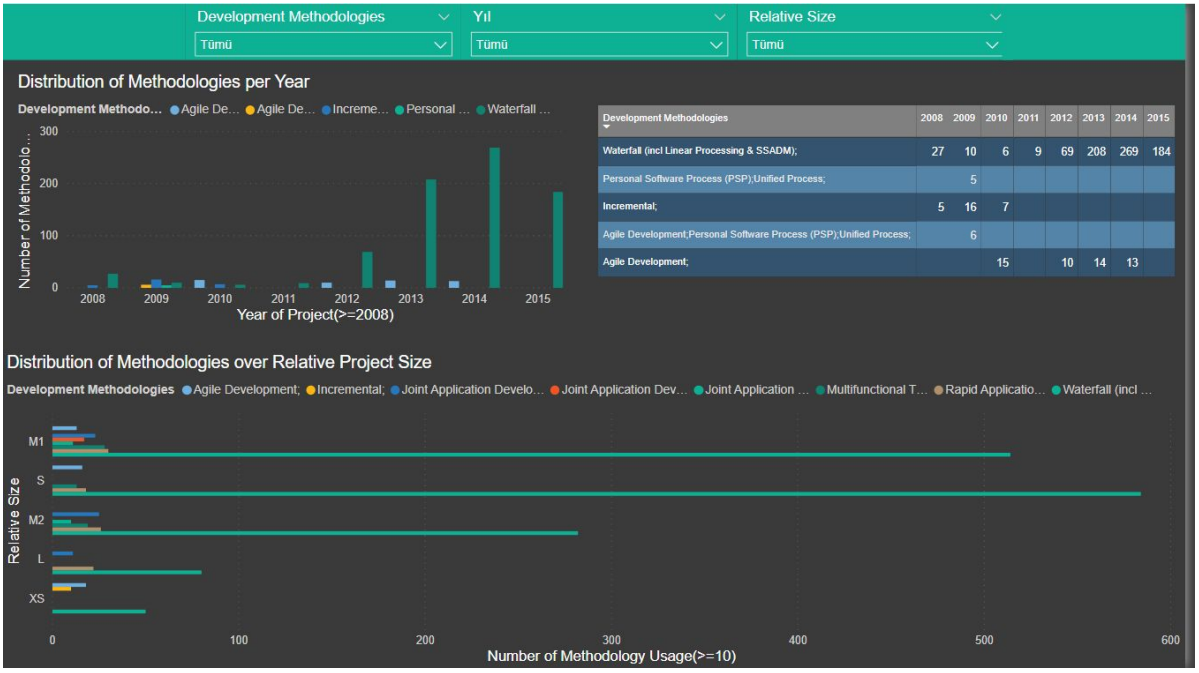
BI feature used : Pivot Tables,Charts, Power BI Filtering and Querying features

BI output description :



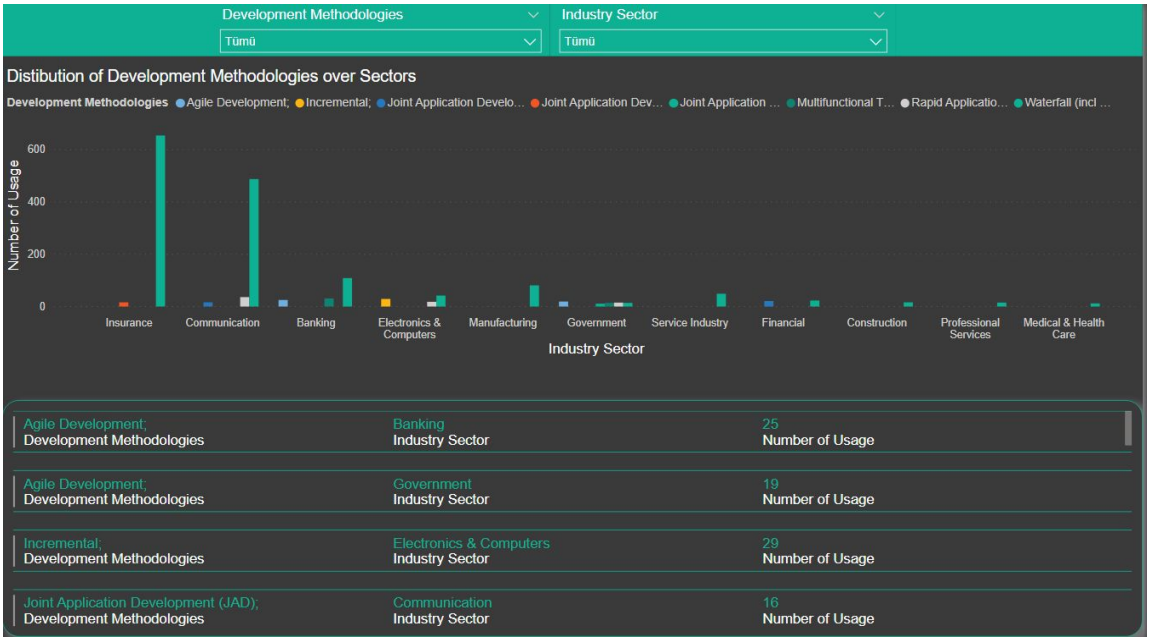
While the Waterfall model ranks 4th in Delivery Speed, it has the least average time in Project Elapse Time, which is the time span covering the entire project, so the project completion time appears faster in the Waterfall model. It is followed by Agile and then Incremental methodology. Since the two graphs (Delivery speed and Elapsed Time) we have examined above are meaningless without knowing the project sizes, we will examine the project sizes in the next step. When we examine the Distribution of Methodologies over Relative Project Size chart:

1. We see the clear leadership of the Waterfall model in all sizes.
2. While the Incremental model is preferred as an alternative for very small projects (XS), Rapid Application Methodology is used as an alternative in small projects (S).
3. In medium sized projects (M1 and M2), the variety is more. Rapid Application and Agile Methodology are preferred as an alternative to the Waterfall model in projects of this size (M1 and M2).



So what about the trends?

While creating the Distribution of Methodologies per Year chart, we based on the methodologies used in projects of 5 and above in 2008 and later. Thus, we can better see what the trends have been in recent years. Waterfall model continues its leadership in recent years, but we see that especially the Incremental methodology has decreased significantly after 2011. Parallel to this, we can see that the Agile development methodology has become widespread in 2009 and after. So according to the trends, Agile Development is in the fashion of recent years. Unfortunately, we cannot see any further since the data is up to 2015.



In addition to all of these, in which sector we will develop a software, it may affect the type of methodology we will use. For this reason, we prepared the Distribution of Development Methodologies over sectors graph.

When we examine the graph:

1. - In the field of Insurance and Communication, we see that Waterfall methodology is very dominant. JAD and RAD methodologies are also used in communication sector.
2. - In banking projects, we can see that in addition to Waterfall, Multifunctional Teams and Agile Development methods are also used.
3. - In the Electronics & Computer sector, we can see that Incremental and Rapid Application Development (RAD) methodologies are also used seriously.
4. - In goverment projects, the leadership belongs to Agile Development methodology with a slight difference. This is followed by RAD, JAD methodologies and Multifunctional Teams.
5. - We see that JAD and Waterfall methodologies are used almost at the same rate in financial projects.
6. - In sectors such as Medical & Health Care, Construction, Manufacturing, we see that the Waterfall model has become traditional and is unrivaled.

Implementation details: We performed our data preparation processes through the Power BI Query Tool. Just like in Q3.1, we deleted the records with insufficient quality from the data set in order to reach smoother results (data with ISBSG rating code D was not used). Used dataset metrics:

- Industry Sector
- Development Methodology
- Project Size
- Year of Project
- Project Elapsed Time
- Speed of Delivery

Later, in each of the 6 metrics (columns) we specified, the rows that were empty (null) or incorrect were deleted. Finally we started to use this dataset to obtain some bar charts, pie charts, tables and other important visualization features thanks to the Power BI. Results and charts can be seen in the pictures above.

<Please explain in one or two paragraphs how you think your BI output could be used to answers Q-1.2, and how you interpret the BI output.>

After the query operations, we lost most of the data we have. However, we think that we have obtained very consistent results with the remaining data. Meaningful graphs about which methodologies can be used in which situation have emerged from Power BI. As we can see from the results, the methodology preference of projects in different sizes and sectors is also differs.

Someone who is unsure about the methodology can get a rough idea by looking at the resulting graphics. Since the choice of methodology is very specific, we cannot say that the results that emerge here are directly correct. Because the development methodology type will also change according to the need and specificity.

3.1.3. (DSS) Suppose we are working on a M1-sized, Enhancement type project for the communication sector. Our team size is 5-8.

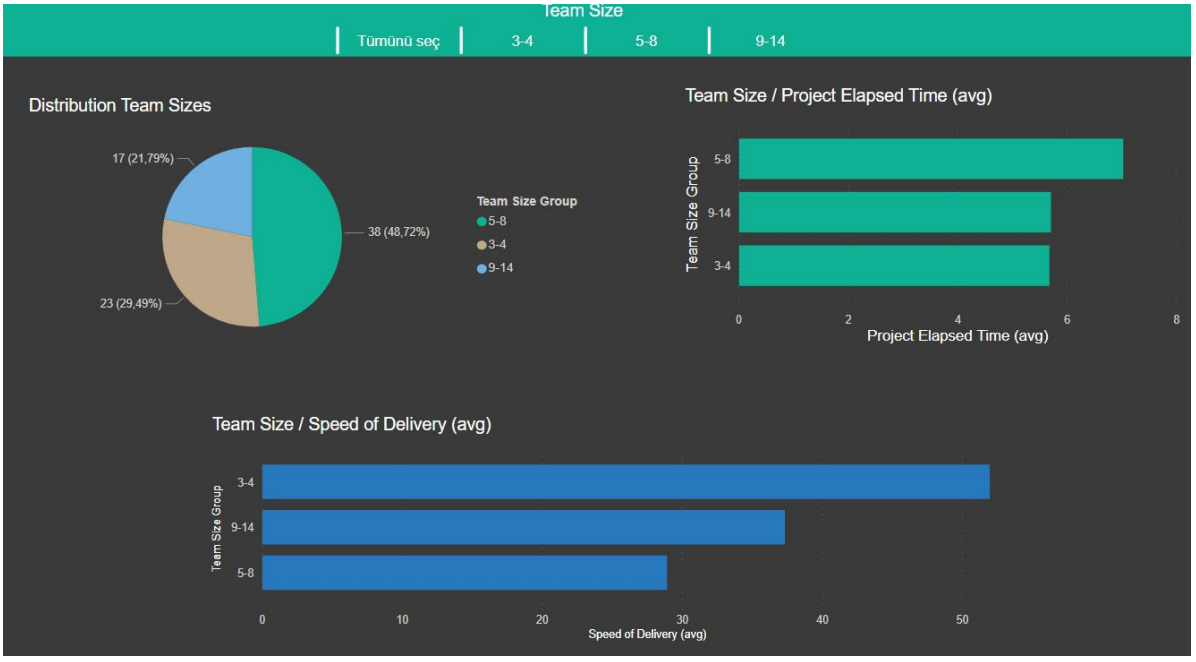
What will happen / what will be the impact if we use 3-4 or 9-14 team group size instead of 5-8?

Purpose of question : Our aim here is on what the effects would have been if we did the project with a team of 3-4 or 9-14 people instead of doing the project with a team of 5-8 people. We will try to reach a general conclusion by examining other projects that exactly match the specified features.

BI feature used : Dashboards, Pivot Tables, Charts, Power BI Filtering and Querying features

BI output description :

We used 7 charts in total to answer this question. 1 Pie Chart and 6 bar charts (3 horizontal and 3 vertical) are used. You can find the graphs below.



When we examine the first graph, we see that groups of 5-8 people are generally preferred in similar projects. In other words, companies working on very similar projects generally preferred groups of 5-8 people. In this respect, we can say that the number of groups of 5-8 people determined in our project is correct. However, before reaching a final decision, we need to add a few more points. That's why other charts were created

When we look at the Average Project Elapsed Time chart, we see that groups of 5-8 people finish the job more slowly. On this chart, the other two groups seem to be doing better.

We can also look at the 3rd chart (Speed of Delivery per Team Size) to verify this. Teams of 3-4 people have interestingly better Delivery speed. Groups of 9-14 people follow this, while groups of 5-8 people take the last place.

To further elaborate, let's include different metric values. When we examine the Defect Density and Defects after Delivery graphics, we find that there is a lot of defect in groups of 9-14 people.
we can see. This is a negative situation. The best value in these two graphs was again obtained from groups of 3-4 people. As the group size grows, the number of errors and defects also increases.

The reason for this may be the communication problem between people in large groups. Considering that the data dates back to 2015, tools such as Jira that strengthen team communication were probably not available on the market or were not widely used.

Chart of Manpower Delivery Rate per Team Size:
This chart was expected. All of the projects in the current data set (with queries applied) are the same size. Therefore, the larger the team size, the less jobs per person will be.
In general, we see that the workload per person in groups of 3-4 people is at a serious level in projects of this size.

Implementation details **:** In order to answer such questions, we need to carefully examine projects with the same criteria by separating them from the data set. Therefore, the following queries were performed on the dataset, respectively:
First of all, projects with D Data Quality were deleted from the data set. Then,

- Project Size: M1
- Sector: Communication
- Development Type: Enhancement
- Team Size: 3-4, 5-8 veya 9-14 kişi

Projects with these four features were extracted from the dataset. 79 different projects were found that fit these criterias. After the new data set we will be working on was created in this way, grouping was made according to team sizes. During grouping, the data set was updated by calculating the average value of each numeric metric that we will use.

Used dataset metrics:

- Team Size
- Project Size
- Industry Sector
- Development Type
- Number of Defects After Delivery
- Defect Density
- Manpower Delivery Rate
- Speed of Delivery

Finally, we created charts and charts on this dataset using Power BI visualization tools. We used lots of bar charts and pie charts because they are easy to understand and fit our purpose. You can see the axes of the graphics and other details in the pictures above.

<Please explain in one or two paragraphs how you think your BI output could be used to answers Q-1.2, and how you interpret the BI output.>

When we examine the output graphics in general, we can say that a group of 3-4 people is more effective than 5-8 employees in a similar type and size project. In groups of 3-4 people, we see that speed of Delivery is faster than others, and at the same time defect density is minimal. In this case, according to these data, better results could have been achieved if the project was started with a group of 3-4 people instead of a group of 5-8 people.

The negative side will be seen on the Manpower Delivery Rate chart, so the workload will be very high in such a small group. When this situation started to cause problems (employee unhappiness due to workload or unwillingness at work, etc.), the group size could be increased to 5-8.

3.2. Questions Answered by Mehmet Serkan Tan - 21627634

3.2.1. (MIS) Q-1.3: Which Development Techniques can be considered more efficient?

Project Elapsed Time by Technique

Purpose of question : Development techniques are indispensable for any sized professional software project. They provide a workflow and help to structure and express software projects.

BI feature used : In this question, other attributes are compared with development techniques. 13 most used development techniques have been used to improve on statistical reliability. All visuals are dynamic (i.e., the mere change affects all related visuals). The interactivity of visuals is important for enhanced comparisons. The report is composed of slicers (i.e., list, dropdown) charts, plots, matrices, and bar graphs.

BI output description : An attribute, Elapsed Time indicates the amount of time that had been spent to successfully finish the project. It is extremely meaningful to compare development techniques by elapsed time since it may reveal the most time-efficient development techniques. In the report, an observer can select development techniques, and set the elapsed time to make specific queries. In addition, different visuals are provided to easily interpret the data. Finally, the total number of projects can be seen which is particularly important when making specific queries.

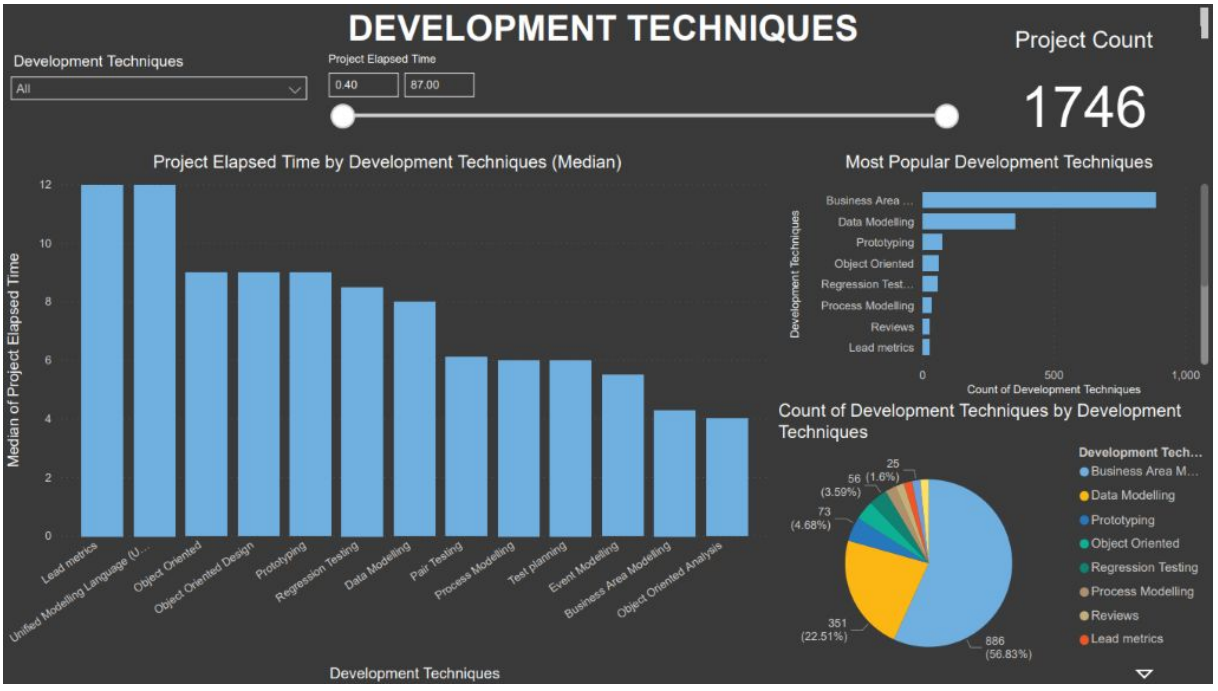


Figure 1. <BI output for answering Q-3.1.1>

Project Rating by Development Technique

Purpose of question : Data Quality and UFP Ratings are important measurements that give a general idea of project quality. It is reasonable to use ratings to determine which development techniques are more efficient.

BI feature used : In the report, there are two 100% stacked column charts that compare development techniques by rating. The observer can compare groups of different development techniques and can infer the most efficient development techniques by their ratings. Besides, matrices have been used to give textual information about the quantity of the ratings.

BI output description : Two different ratings can be measured by the column charts. Each column has the rating distribution (i.e., A B C). The most 15 used techniques are listed for brevity. Observers can select specific independent variables to localize their solution.

Implementation details : Data quality and UFP ratings are the only ratings provided by the dataset. However, the density of the columns is quite satisfying, and it did not require too much preprocessing. But the rating D was removed since the quantity is too few.

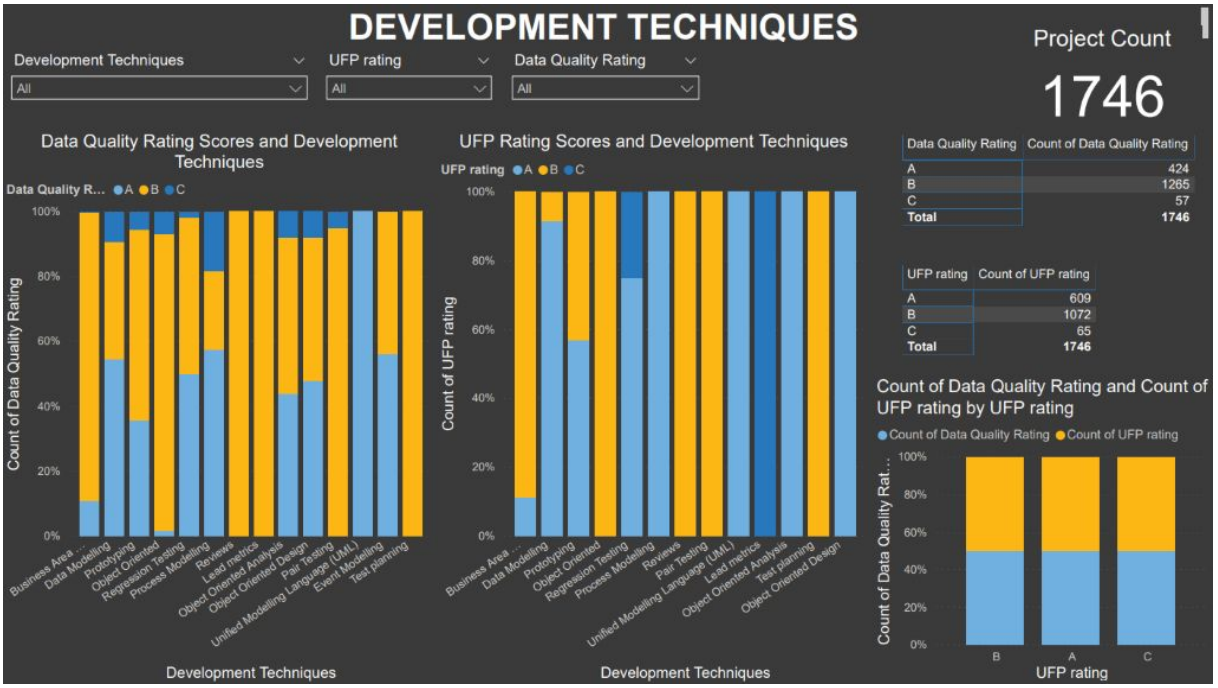


Figure 2. <BI output for answering Q-3.1.2>

Functional Size by Technique

Purpose of question : Software projects vary by their size. The metric called functional size helps determine the complexity and functionality. So that the development techniques are compared to determine which are more suitable for large, middle, small sized projects.

BI feature used : A bar chart and pie charts are utilized and the average of functional sized is queried for each development technique.

BI output description : The produced bar chart is sorted by the functional size on the x axis. It clearly shows us, UML is the best fit when the project complexity is at the highest. Then it is followed by Object Oriented Modelling. As the functional bar size is narrowed towards less complex projects all the models become preferable for projects. At the same time, the pie chart can be viewed to understand the distribution quantitatively.

Implementation details : An extensive preprocessing was required to fulfill the task for example, some of the entries in the functional size contained no data, and they were replaced with 0. Then removed from the statistics.

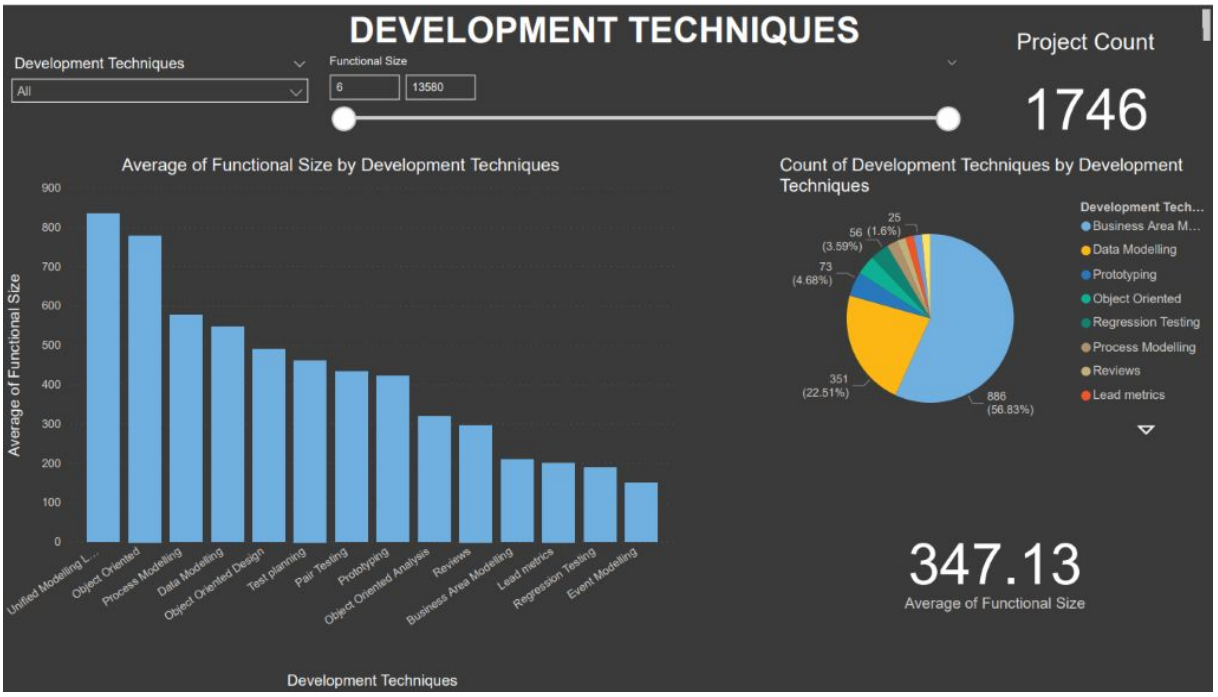


Figure 3. <BI output for answering Q-3.1.3>

Development Methodologies by Technique

Purpose of question : Development methodologies are essential and quite useful for corporate level software projects. Companies can exploit and follow the steps of a selected technique and successfully structure their projects.

BI feature used : The most ten used methodologies are selected. The observer can interactively select specific variables to get more insight. a 100% stacked column chart is used to show the distribution of the methodologies by each development technique. In addition, to visualize values, a matrix is created to show the popularity of these methodologies.

BI output description : The stack column chart is implemented for the most popular ten methodologies. It can easily be seen that waterfall is dominant in all applied development techniques. It is followed by Joint Application Development. On the other hand, it is notable that Agile Methods are far behind the dominant techniques. Interestingly, the second most popular development technique, Data Modelling uses mostly Joint Application Development, and the Waterfall method is minority.

Implementation details : The observer can view specific techniques, and methodologies by altering the slicers at the top of the page. Unfortunately, the data quality of development methodologies attribute is not satisfactory. However, a few dominant techniques let the data presentable, and statistically accurate.

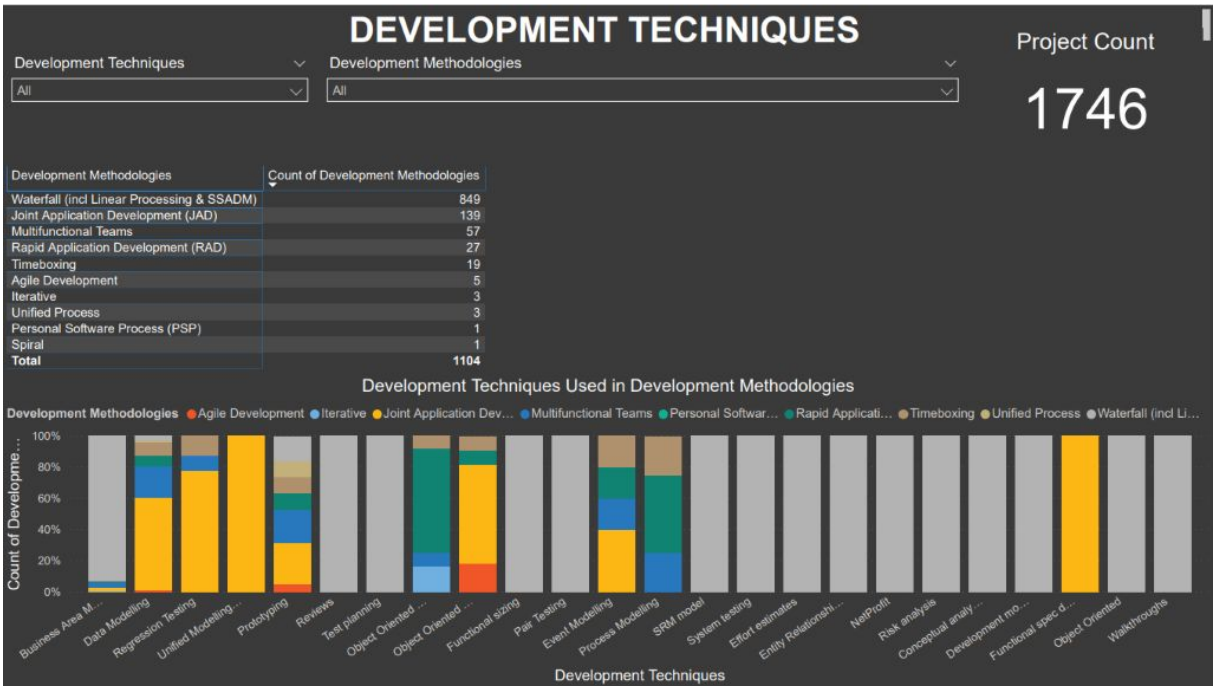


Figure 4. <BI output for answering Q-3.1.4>

Software Architecture by Technique.

Purpose of question : Software architectures give many insights about the software such as its purpose, intended users, scale, and platform etc. In this way, analysts can utilize the relationship between development techniques and software architectures to ease the complexity of required work and to make more concrete assumptions. Besides, project managers and architects can decide which technique to use for their project that follows one of the listed architectures.

BI feature used : In the report, analysts can interactively select architectures and techniques to view its spread in the dataset. Stacked column charts are used to show the popularity of each architecture by a development technique. Lastly, Numerical data is provided to view the spread more easily.

BI output description : The bar chart shows the architectures preferred in the most popular development techniques. When stand-alone architecture is not included, the bias is towards the client server architecture. However, Business Area Modelling is dominated by multi-tier architecture. In the BI report if analyzed carefully, the stand-alone architecture is dominated in every technique except the Business Area Modelling.

Implementation details : Almost 400 project entries do not have architecture attributes so that they were removed from the report.

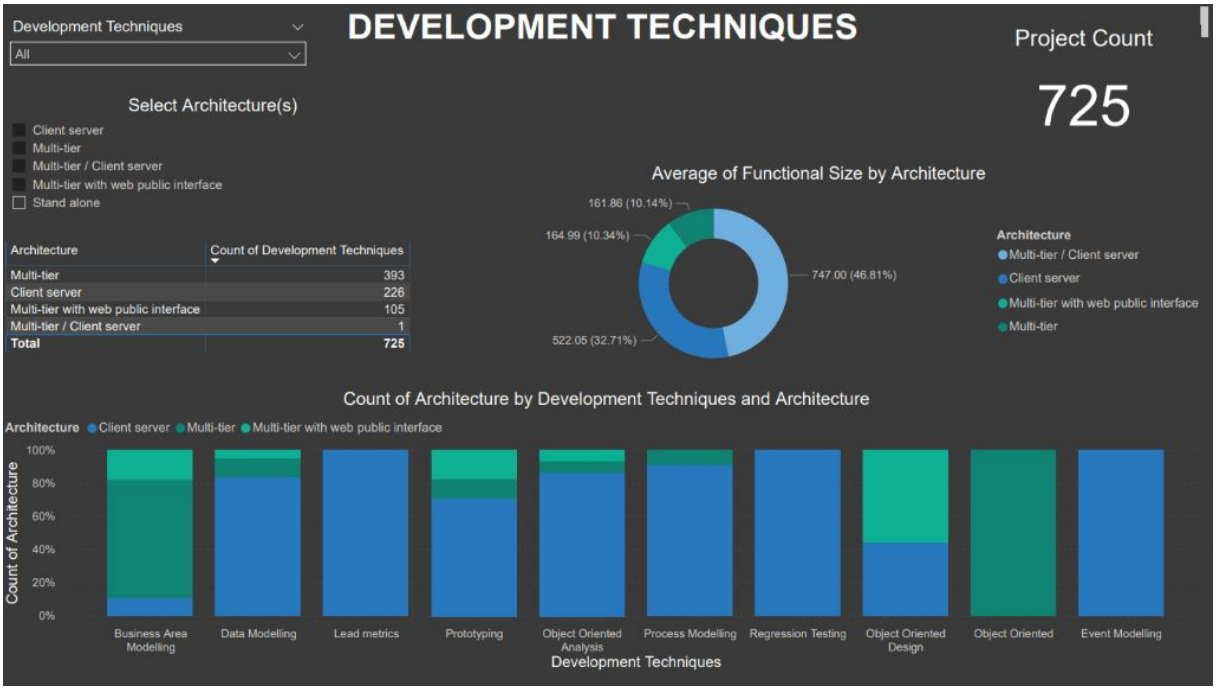


Figure 5. <BI output for answering Q-3.1.5>

Software Defects by Technique

Purpose of question : Software defects, including bugs, operational level errors, system design anomalies are an important factor in a software development cycle. Defects can cause late deliveries, can be costly and damage the reputation of a company. Development methodologies play an important role for designing the software with the lowest cost. Managers can decide which techniques are better for minimizing the number of defects with the aid of historical data.

BI feature used : Defects are investigated in three groups according to their severity levels i.e., Extreme Defects, Major Defects, and Minor defects. The analyst can set the bar of tolerable defects to compare each development technique. Besides, an average of total defects is provided to get a general idea about the technique. Lastly, a pie chart is provided to view the ratio of each defect in the whole ecosystem.

BI output description : The second most popular development technique, Data Modeling has the most total defect amount, it is followed by a less popular technique, Regression Testing. The most occurring defect types are Minor (50%) then Major (31%), and Extreme (18%). Data modelling has the defect ratio; 16%, 12%, 9% with respect to increasing severity.

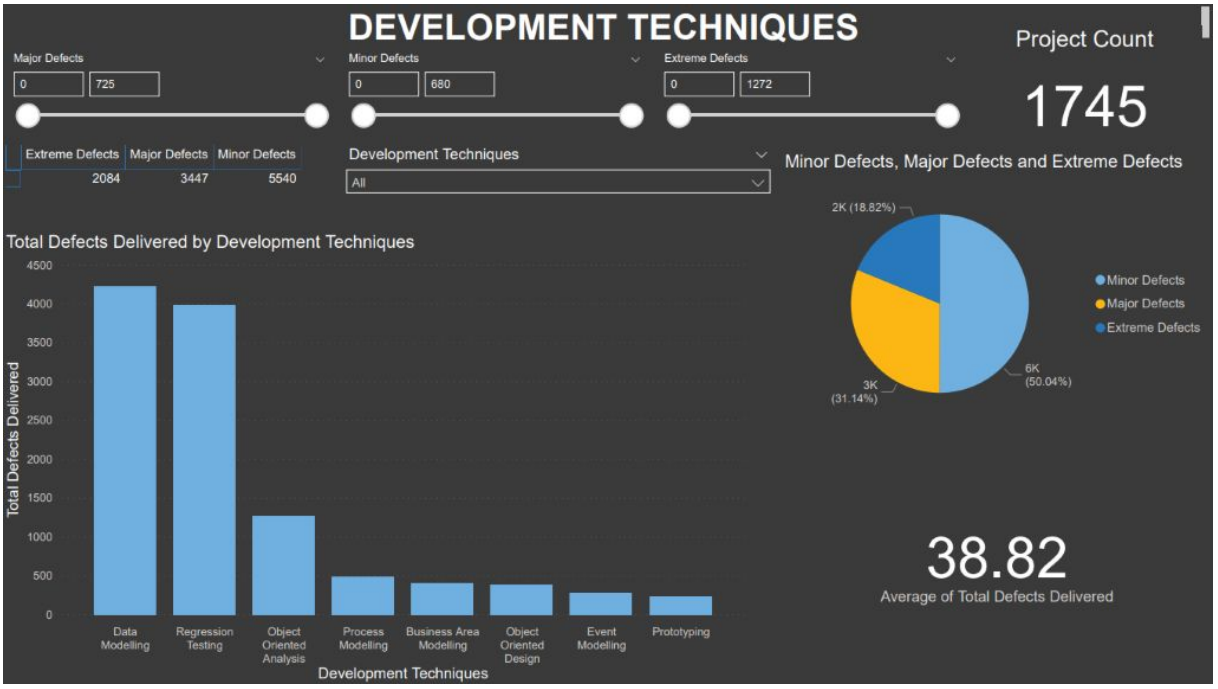


Figure 6. <BI output for answering Q-3.1.6>

Effort by Technique

Purpose of question : An essential efficiency metric of a software project is the amount of effort exerted as building the software. By looking at the average and median of the effort, the most efficient development techniques can be selected. Seven metrics are used to measure the effort in hours. (i.e., build, design, implementation, specify, plan, test, and lastly unrecorded)

BI feature used : A bar graph is provided to show which development techniques are most exhaustive. As well as the average effort values in the whole ecosystem. To make focused observations, desired functional size and development techniques can be set to specific values.

BI output description : The unrecorded efforts can indicate the development technique does not let a structured development environment and/or setting. Build effort is the most occurring effort type in the projects.

Implementation details : The average of the efforts have been taken. Empty entries were removed.

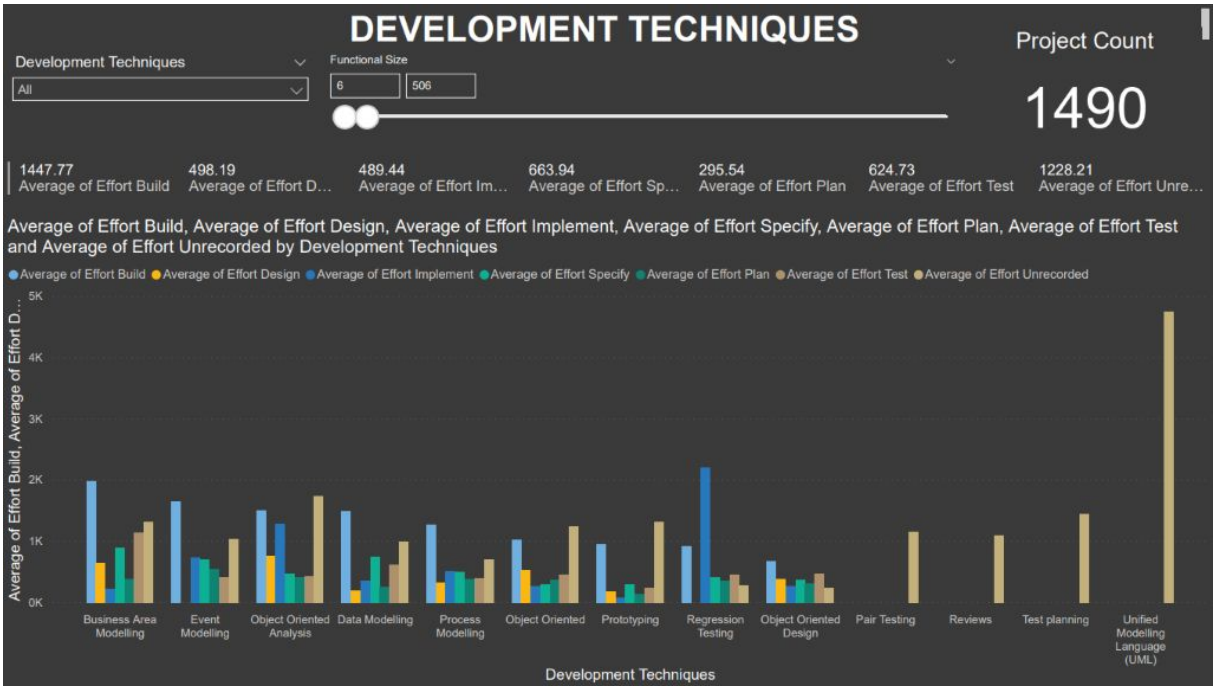


Figure 7. <BI output for answering Q-3.1.7>

3.2.2. (MIS) Q-1.4: How the team experience is related to other metrics and how experience affects efficiency?

Team Experience by Employee Role

Purpose of question : Because of the existing of multiple independent variables, an overview of the team experience is required. Visuals about the ratio between the team experiences will make the report more comprehensive.

BI feature used : The spread of both IT and BA teams are analyzed. Pie charts used to show each experience span. At the same time, the analyst can view the ratios by the functional size of projects. Also, the relationship between employee roles can be viewed better by using relative (i.e., XXS, XS, S, M, L, XL, XXL) size metric provided by the dataset.

BI output description : The employees in the IT sector that have few years of experience are quite few. However, in the BA sector the distribution is more uniform.

Implementation details : Analysts can view the distribution with the slicers. Too big functional size records were deleted since it leads to inconsistency in data.

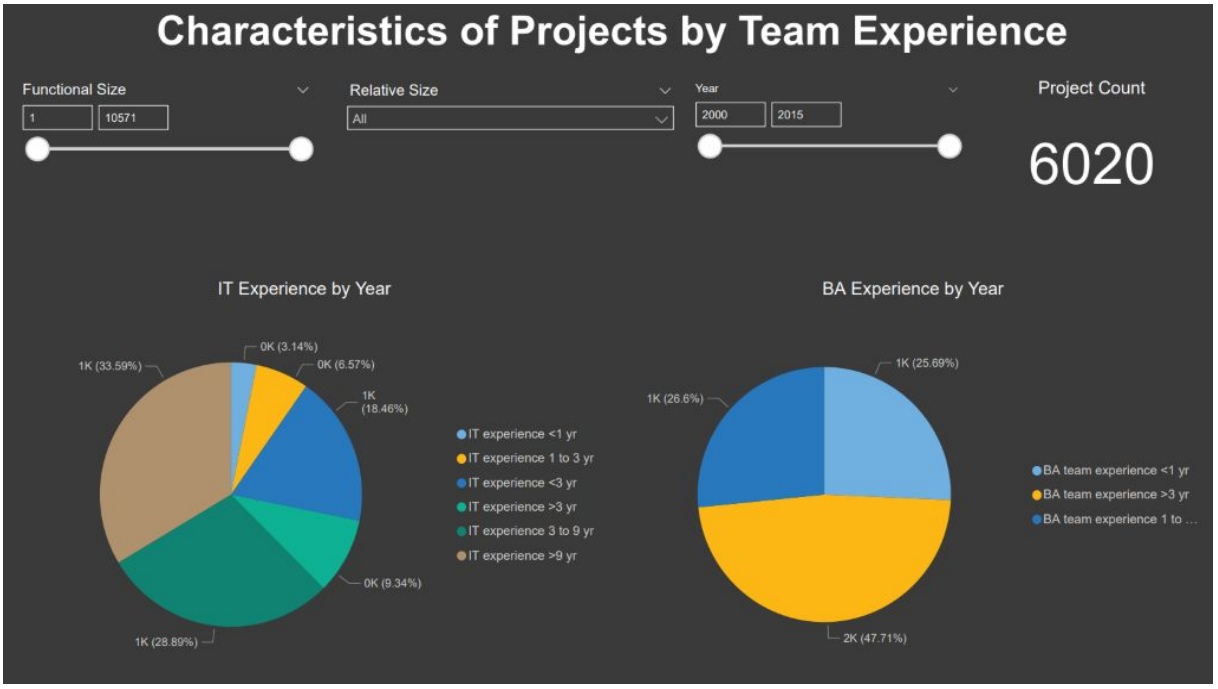


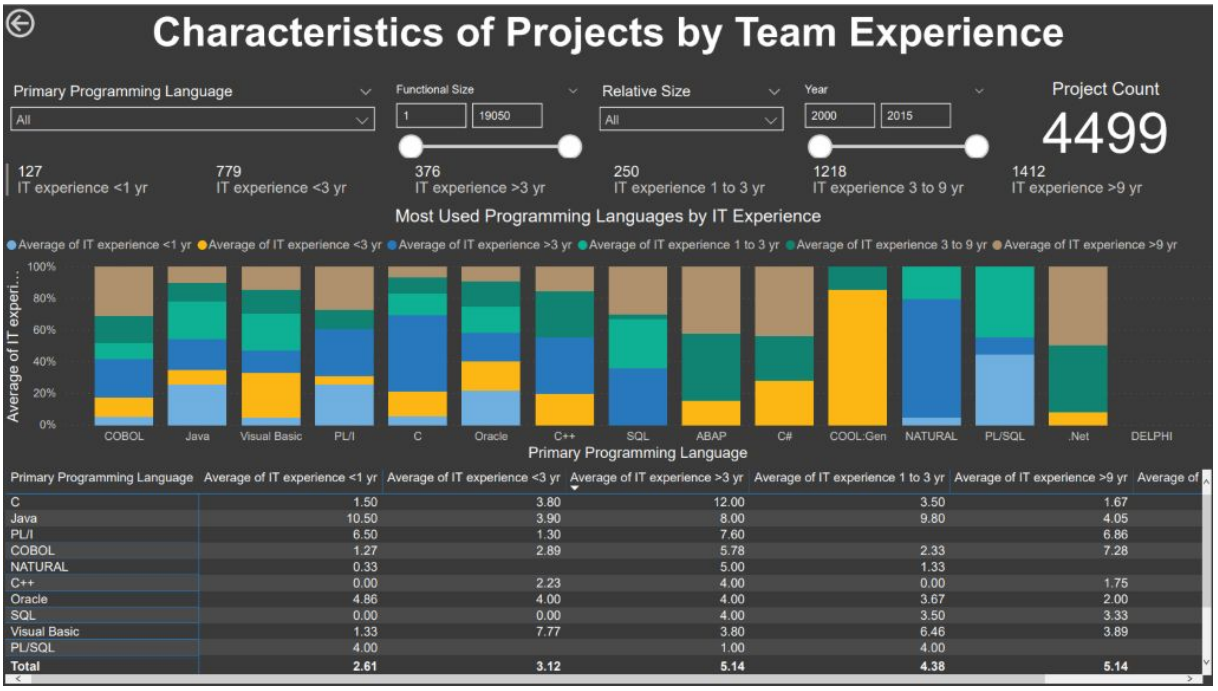
Figure 1. <BI output for answering Q-3.2.1>

Programming Languages Preferred by Teams

Purpose of question : Programming languages are an important metric for teams. Depending on the teams' specification different design choices can be preferred. So, this question can help get better answers for observers who strive for conventional solutions.

BI feature used : A 100% stacked bar chart is used to view the popularity of each experience level. Also, a table presents the results quantitatively. Multiple slicers are used to enhance the search experience.

BI output description : More experienced teams tend to use COBOL (relatively an old language) while, less experienced teams tend to use Java. Low level but popularly used languages such as C/C++ are mostly used by More experienced teams. Database languages are used quite equally by every experience level. When the year slicers set to recent years, also the more experienced teams tend to use new generation languages such as Java, Oracle.



Primary Programming Language	Average of IT experience <1 yr	Average of IT experience <3 yr	Average of IT experience >3 yr	Average of IT experience 1 to 3 yr	Average of IT experience >9 yr	Average of
C	1.50	3.80	12.00	3.50	1.67	
Java	10.50	3.90	8.00	9.80	4.05	
PL/I	6.50	1.30	7.60		6.86	
COBOL	1.27	2.89	5.78	2.33	7.28	
NATURAL	0.33		5.00	1.33		
C++	0.00	2.23	4.00	0.00	1.75	
Oracle	4.86	4.00	4.00	3.67	2.00	
SQL	0.00	0.00	4.00	3.50	3.33	
Visual Basic	1.33	7.77	3.80	6.46	3.89	
PL/SQL	4.00		1.00	4.00		
Total	**2.61**	**3.12**	**5.14**	**4.38**	**5.14**	

Figure 2. <BI output for answering Q-3.2.2>

Industries and Teams

Purpose of question : Team overall experience by industry sector is another valuable contribution to see which sectors are more dynamic or employ more experienced workers.

BI feature used : A pie chart shows the distribution of industries, and bar plots show industries by team experiences. Slicers are used to make the report more interactive.

BI output description : When we look at the five biggest industries, the least experienced group in the IT sector is as much employed as the rest of the industries. However, it is just the opposite in the BA sector. At the same time, companies tend to employ more experienced employees from all the sectors. Most experienced employees work in the Government industry sector.

Implementation details : Data processing was a quite challenging task for this question. The columns were handled independently.

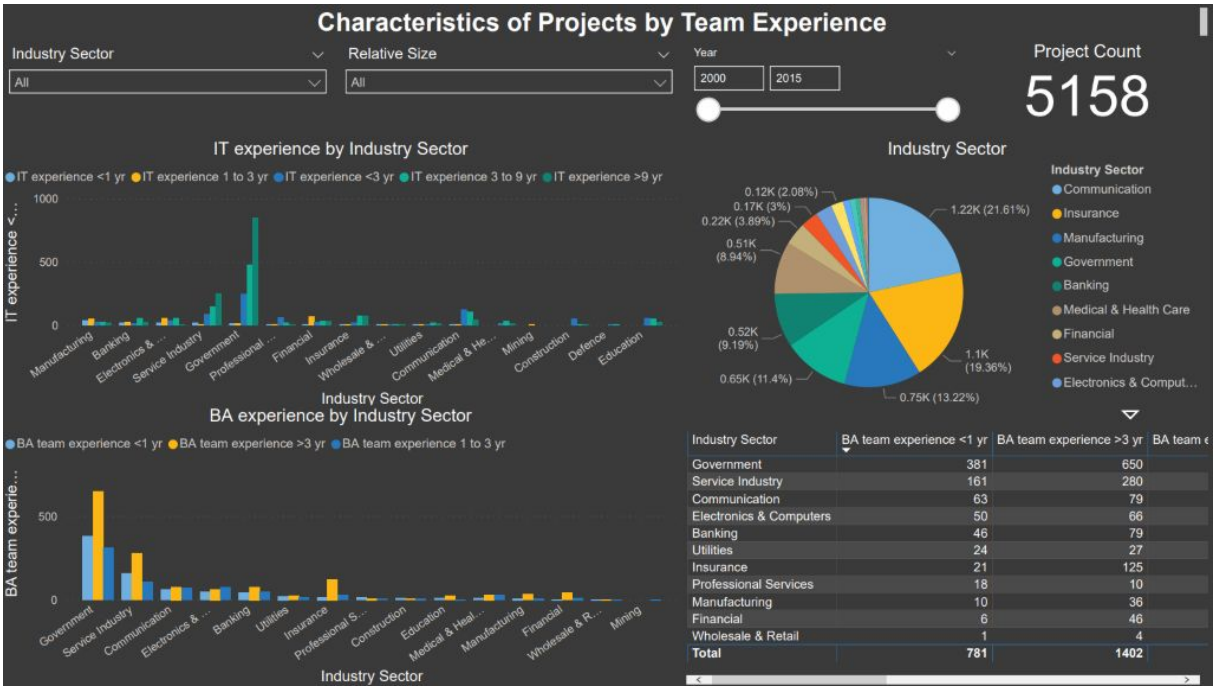


Figure 3. <BI output for answering Q-3.2.3>

Delivery Rate and Teams

Purpose of question : Delivery rate shows the amount of work done by employees. In the dataset delivery rate is spread by each experience level. Hence, the total work done by each experience level is compared.

BI feature used : Stacked area chart is used to show delivery rate of change between each bin for delivery rate. A bar plot shows the total delivery in each bin. Grouping is used to create the bins.

BI output description : Both in the BA, and IT sector, more experienced employees have a higher delivery rate. The bin no. 1 is the IT sector employees' highest delivery rate range. It's ~1 for the BA sector.

Implementation details : A new measure is created to classify the ranges of delivery rates by 10. For example, the delivery rate in between 0-9 is the bin no. 1 and so forth.

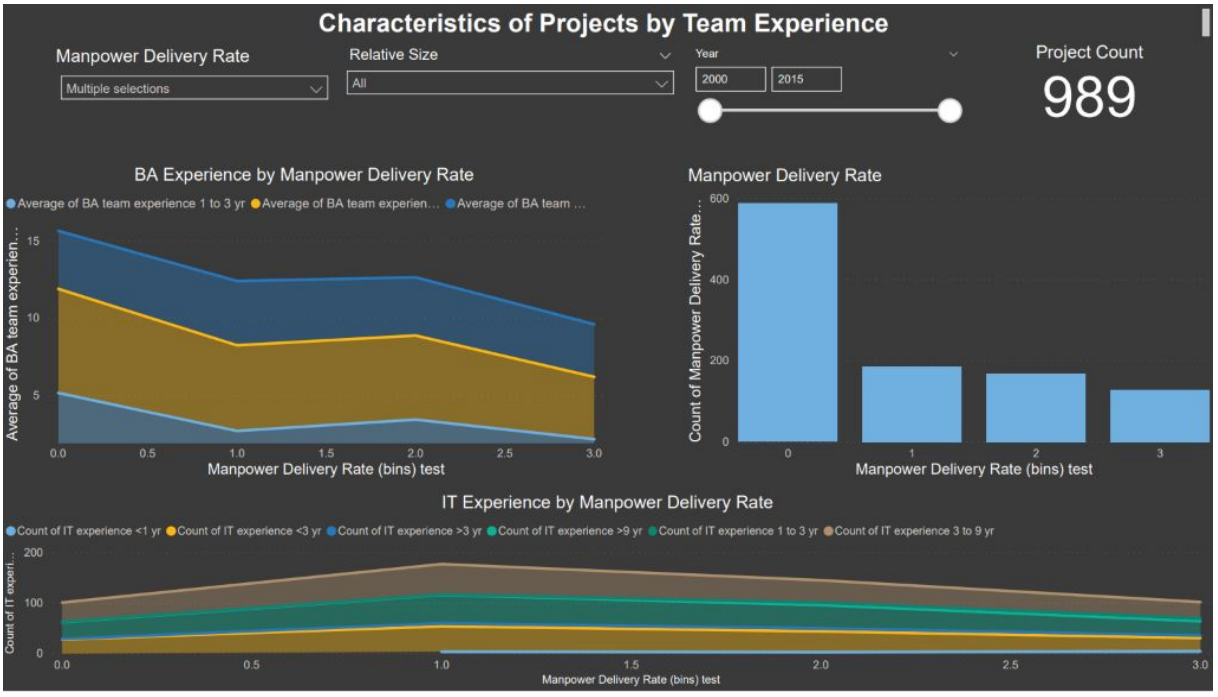


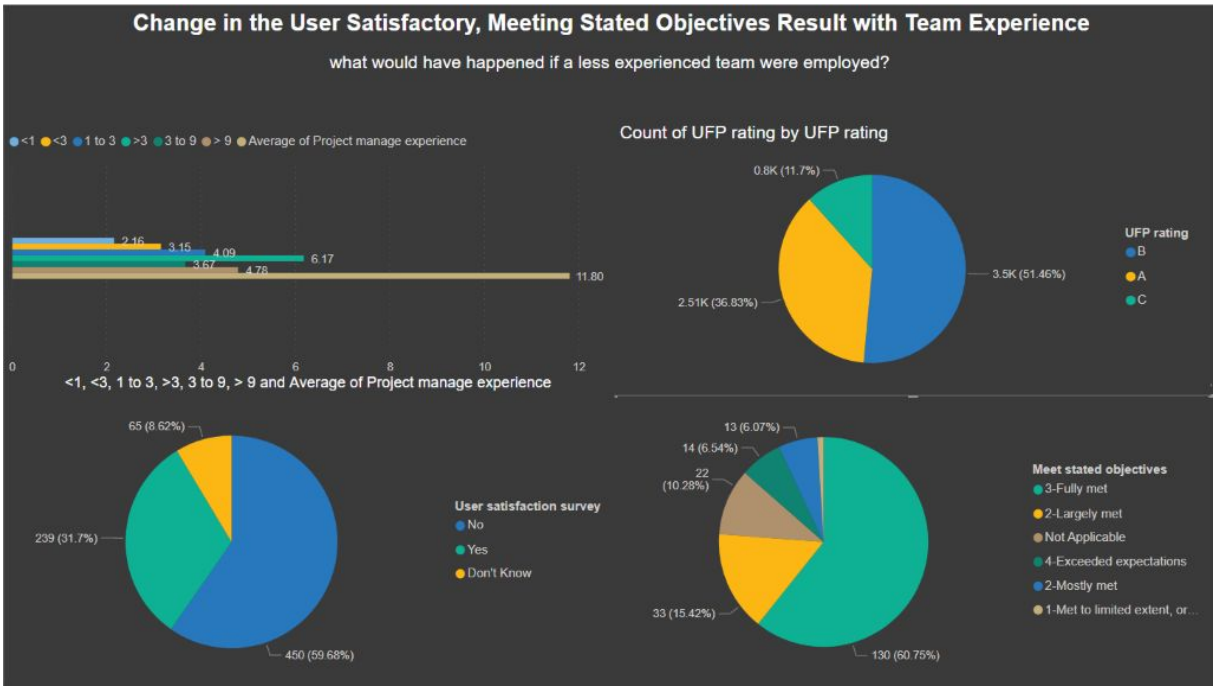
Figure 4. <BI output for answering Q-3.2.4>

3.2.3. (DSS) Q-1.6: What Would Have Happened If A Less/Most Experienced Team Were Employed?

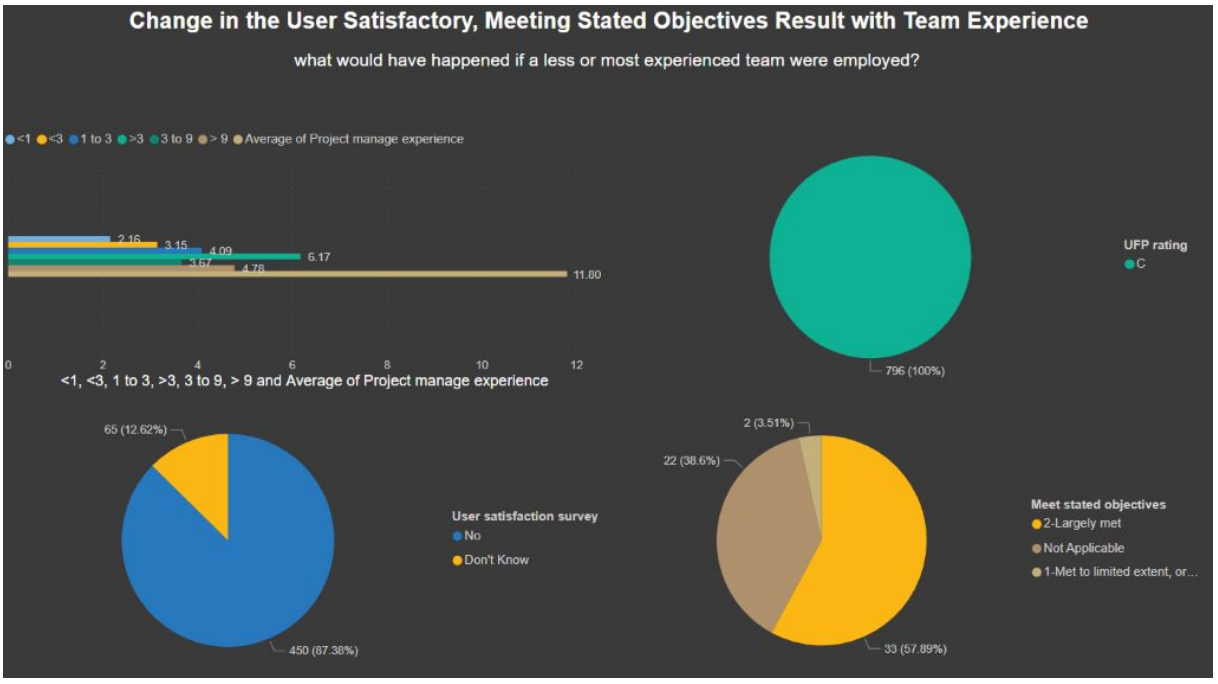
Purpose of question : Software projects are maintained by both more and less experienced developers/engineers. When the budget is a concern, managers may want to employ less experienced developers to deploy the projects. However, it is a risky option since they could blunder the project and customers may be dissatisfied. So, to experiment these two situations, teams are assumed to be less or more experienced.

BI feature used : A horizontal chart is used to show the average of experiences by year. Pie charts are used to view effects in the change of experience.

BI output description :

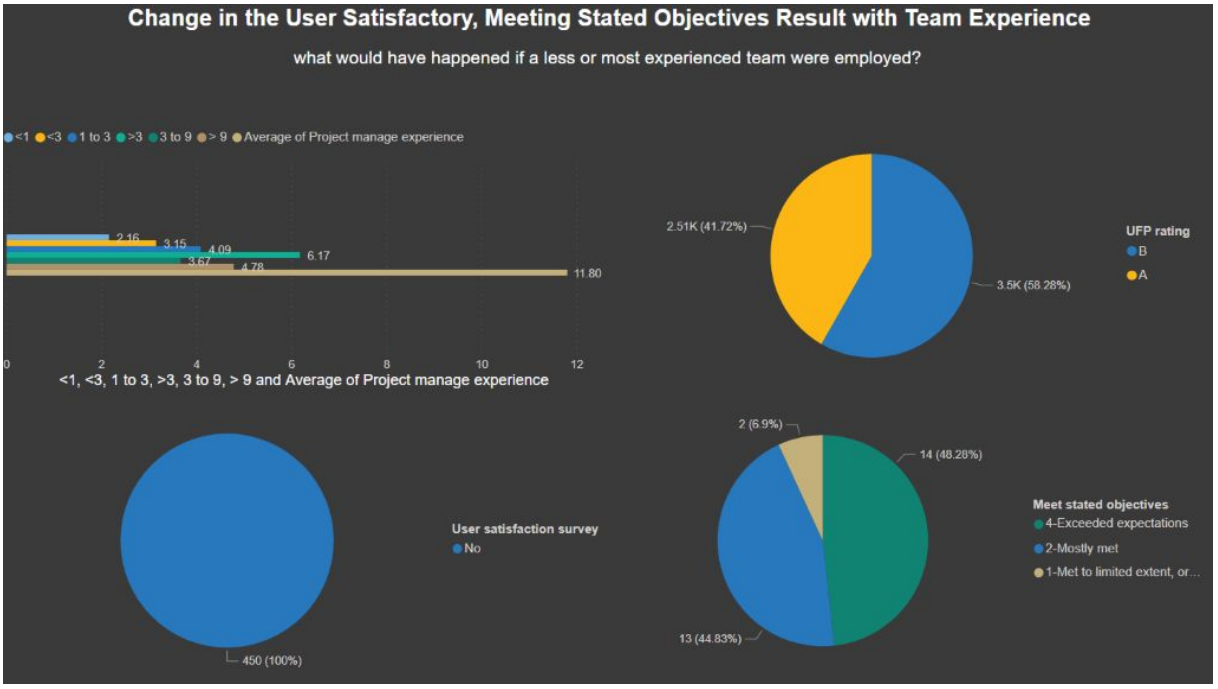


In the figure above, the setup can be seen, when data is not manipulated. The bias is towards non-satisfactory for user satisfaction survey, the UFP rating obeys the bell curve structure, and same for the stated objectives.



In the figure above, when the projects that seriously lack the most experienced developers are excluded from the projects, the distribution of each pie chart is viewed. All pie charts are designed to be independent. No data collision is possible.

The visuals are clear enough to show, in a situation where the projects lack experienced engineers, at the end the result may be not satisfactory both for software rating surveys, and customers.



When the less experienced employees are excluded, interestingly, improving results are obtained.

- User satisfaction survey got worse. However, because of the biased dataset, an anomaly is expected. Unfortunately, it does not lead to reliable results.

- UFP rating is improved by proportionally replacing Cs with Bs.

- Meet stated objectives improved spectacularly by getting almost half of the projects rated as exceeded expectations.

Implementation details :

The pattern for excluding the experienced employees followed as:

- Accepting the average of < 1 for the employees that experienced > 9 years.

- Accepting the average of < 2 for the employees that experienced 3 to 9 years.

The pattern for excluding the less experienced employees followed as:

- Accepting the average of < 1.5 for the employees that experienced < 1 years.

- Accepting the average of < 2 for the employees that experienced 1 to 3 years.

Ultimately, the charts indicate experienced employees essential for teams which aim to maximize customer satisfaction. However, the data can be very biased and requires more surveying for a better resolution.

4. OVERALL EVALUATION

Table 4. Overall evaluation of the results in BI Project

Goal	Related Question	Achieved or Not?	Reason of Evaluation
1	1.1 (MIS)	Achieved	When we used the necessary metrics and removed the unnecessary data, we saw that we had enough samples. The results were also clean and smooth enough. In this case, we got clear answers to our question.
1	1.2 (MIS)	Achieved	I think the sample size we have is sufficient for this kind of question after the processing and clearing of the dataset, . We can also see from the results that we have selected the metrics in accordance with the question. For a little more precise result, if the data of recent years (projects developed after 2015) were available, much better results could be obtained.
1	1.3 (DSS)	Partially Achieved	The results were also clean and revealing enough. In this case, we were able to get clear answers to our question. However, there were only 78 project examples similar to the scenario in the question. I think it is necessary to increase the sample size to get a best result.
1	1.4 (MIS)	Achieved	Stated Metrics are statistically partially accurate; some metrics were too few so they had to be removed , Results are both predictable and measurable. Questions are valid, important and they cover a variety of metrics that solidify the statements and arguments. The question is relatable to the goal.
1	1.5 (MIS)	Achieved	Stated Metrics are statistically accurate, Results are both predictable and measurable. Questions are valid, important and they cover a variety of

			metrics that solidify the statements and arguments. The question is relatable to the goal.
1	1.6 (DSS)	Achieved	Stated Metrics are statistically accurate, Results are partially predictable and measurable. Questions are valid, important and they cover a variety of metrics that solidify the statements and arguments. The question is relatable to the goal.

5. LESSONS LEARNED FROM THE PROJECT

Using BI tools and creating appealing reports.

Extracting knowledge out of raw data using visuals and quantitative measurements.

Preprocessing dirty data.

Grouping attributes and viewing their interdependencies.

Acquiring explanations and making action plans.

Creating Management Information Systems and Decision Support Systems questions from the perspective of a company.

Feedback: We think it was a really useful project, both for enhancing our knowledge in data science and business intelligence and for gaining some insights of business processes in companies. We think this project should be covered in several smaller-sized homeworks so that different kinds of datasets can be experienced. Seeing different models will be very helpful for proficiency.

6. REFERENCES

[1] ISBSG Development & Enhancement Repository Data, April 26, 2017

[2] R. Van Solingen, V. Basili, G. Caldiera, D.H. Rombach, Goal Question Metric (GQM)

Approach, in: Encyclopedia of Software Engineering, online vers.@Wiley Interscience, John

Wiley & Sons, 2002

[3] Management Information Systems: Managing the Digital Firm. Prentice Hall, Chapter Slides

[4] https://docs.microsoft.com/en-us/power-bi/fundamentals/