



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

TREBALL FI DE GRAU

Grau en Enginyeria Mecànica

**IMPLEMENTATION OF DMAIC AND AXIOMATIC DESIGN
TO THE CREATION OF A BASIC SMARTPHONE
APPLICATION**



Memòria i Annexos

Autor: Antoni Amat Serra
Director: Joan Martínez Sánchez
Convocatòria: Octubre 2017

ACKNOWLEDGEMENT

I would like to thank the Politecnico di Bari to give me the opportunity to study for one semester in Bari, also to Yasamin Eslami for helping me during all the process to finish this project.

I would like to specially thank Joan Martinez to accept directing my work in such a short notice giving me the opportunity to finish my bachelor.

I would like to thank my Parents to give the opportunity to study and to carry out this project abroad and also for giving me the spirit to finish the Bachelor.

ABSTRACT

This project implements two theories to the creation of a basic smartphone application: DMAIC and Axiomatic Design Theory.

DMAIC is very often used methodology that very successful enterprises had already implemented; it is used in Six Sigma, a procedure that ensures a large reduction on the errors of the enterprise processes. This method has been used by a lot of companies all over the world providing them a large increase of quality producing.

Axiomatic Design Theory takes its name from the fact that it is composed of two Axioms that have to be accomplished when implementing this theory. It is a method meant to design with a very specific order.

Both of these theories should be able to be used in any project, they are usually performed in manufacturing projects, but in this case are implemented in the creation of a software product.

This software product is a very basic application that would allow the customer to read interesting facts and try to find out whether if it is true or false.

TABLE OF CONTENTS

ACKNOWLEDGEMENT	I
ABSTRACT	II
TABLE OF CONTENTS	III
LIST OF TABLES	V
LIST OF FIGURES	VI
LIST OF EQUATIONS	VIII
1. INTRODUCTION	1
1.1 Objective of the project	1
1.2 State of the Art	2
1.3 Start-up.....	6
1.4 Approach to DMAIC.....	7
1.3.1 Define	7
1.3.2 Measure	8
1.3.3 Analyze	9
1.3.4 Improve	9
1.3.5 Control.....	10
1.5 Approach to Axiomatic Design	11
2. IMPLEMENTATION OF THE THEORIES	13
2.1 DMAIC Implementation	13
2.1.1 Define	13
2.1.2 Measure	29
2.1.3 Analyze	38
2.1.4 Improve	45
2.1.5 Control.....	47
2.1 Axiomatic Design Implementation	48
2.2.1 Domains.....	48
2.2.2 Axioms Implementation	51
2.2.2 Axiomatic Design outcomes	58
3. COMPARISON AND CONCLUSIONS	59
3.1 DMAIC implementation conclusions.....	59

3.2 Axiomatic design conclusions.....	59
3.3 Comparison between the theories	59
3.4 General conclusion.....	60
4. REFERENCES	61

LIST OF TABLES

Table 1: Pros and Cons of <i>Interesting Facts 30k+</i>	3
Table 2: Pros and Cons <i>Funny facts 8000+</i>	3
Table 3: Six Sigma Costs and Savings by Company	5
Table 4: Company structure	6
Table 5: Axioms in Axiomatic Design Theory	11
Table 6: Data collection plan.....	29
Table 7: Hypothetical measurements for the sampling.....	31
Table 8: Sampling results	31
Table 9: Continuous Measuring	32
Table 10: Statistical basic indices	33
Table 11: Quality measurement of two existing apps	34
Table 12: Proposed FRs and DPs Relation.....	52
Table 13: New FRs, DPs and PVs	53

LIST OF FIGURES

Figure 1: <i>Interesting Facts 30k+</i> screenshot #1.....	2
Figure 2: <i>Interesting Facts 30k+</i> screenshot #2.....	2
Figure 3: <i>Funny facts 8000+</i> Screenshot #1	3
Figure 4: <i>Funny facts 8000+</i> Screenshot #2	3
Figure 5: Decomposition of the surface of a rotating shaft for interfacing with a lip seal (Brown, 2011)	6
Figure 6: DMAIC procedures	7
Figure 7: Define procedures.....	7
Figure 8: Measure procedures	8
Figure 9: Analyze procedures.....	9
Figure 10: Improve phases	9
Figure 11: Control procedures	10
Figure 12: Domains, Axiomatic Design Theory for Systems, (Suh, 1999)	11
Figure 13: Project Charter	14
Figure 14: VOC diagram (Montagna)	16
Figure 15: HOQ.....	17
Figure 16: HOQ Competitive Analysis	18
Figure 17: HOQ Quality Characteristics.....	19
Figure 18: HOQ Relationship strength	19
Figure 19: HOQ Correlation between QCs	20
Figure 20: HOQ Directions of Improvement Symbols.....	20
Figure 21: HOQ Correlation Symbols	20
Figure 22: HOQ Results	21
Figure 23: SIPOC Diagram #1 Create a library of Facts	22
Figure 24: SIPOC Diagram #2 Define the App design.....	24
Figure 25: SIPOC Diagram #3 Create True/False System	25
Figure 26: Flow Chart of finding new facts.....	27
Figure 27: Histogram #1: Time for introducing a fact	33
Figure 28: Histogram #2: Time for Photo in a fact	34
Figure 29: Pareto #1 Fun Facts 8000	36
Figure 30: Pareto #2 10,5k Cool Facts.....	36
Figure 31: Fishbone Diagram Structure (isixsigma.com).....	38
Figure 32: Built Ishikawa Diagram.....	39
Figure 33: 5 Whys Diagram	41
Figure 34: Improve step implementation (Montagna)	45
Figure 35: ADT Domains (Suh, 1999).....	48
Figure 36: Domains Defined.....	50
Figure 37: ADT new Domains	54
Figure 38: Proposed new FRs and DPs Relation.....	55
Figure 39: Proposed new DPs and PVs Relation	56

LIST OF EQUATIONS

Equation 1: FRs and DPs relation (Suh, 1999)	12
Equation 2: Minimum Sample size required (Montagna)	30
Equation 3: Link between FRs and DPs (Suh, 1999)	51
Equation 4: Link between DPs and PVs (Suh, 1999)	51
Equation 5: Diagonal Matrix	51
Equation 6: Triangular Matrix	52
Equation 7: Design Matrix from the FRs and DPs proposed	52
Equation 8: New Design Matrix [A]	55
Equation 9: Design Matrix [B]	56

1. INTRODUCTION

1.1 Objective of the project

The aim of this project is to study how DMAIC and Axiomatic Design can help Start-up company to launch a simple product to the market. These theories were meant to be used in big companies and normally in manufacturing companies. However some experts about this area had proven that it can be used in any type of company.

Six Sigma has been adopted by many software development organizations to identify problems in software projects and processes, find optimal solutions for the identified problems, and quantitatively improve the development processes so as to achieve organizations' business goals (Pan, Park, Baik, & Choi, 2007).

This fragment is referring to Six Sigma but it has been referenced because Six Sigma's main tool is DMAIC.

Axiomatic design theory is applicable to many different kinds of systems, including machines, large systems, software systems, organizations, and systems consisting of a combination of hardware and software (Suh, 1999).

So this project is meant to show if these theories would be suitable for a very small software company.

1.2 State of the Art

In this project it will be tried to implement DMAIC and Axiomatic Design to the creation of an application, a very basic one that could be designed by a very small team. The type of application built is what it will be referred as a “Fun Facts” app, in which the user can learn about some historical, chemical, physical, or of any type surprising facts. There are a lot of existing applications like these but this project intends to create a new one that solves the problems that the existing ones have.

The idea of this project is to improve the existing “Fun Facts” applications by introducing new designs and making the customer interact more with it. The main reason for creating this application is the idea that the existing ones have a lack of attractiveness; the designs used are directly taken from *Android Studio*, the *Google* software for applications creation, without any photographs, a large amount of spelling mistakes and fake statements.

However some of these existing applications have more than 500.000 downloads. Some of the applications that will be compared are:

Interesting facts 30k+

Company: Meshapps

Downloads: +50.000



Figure 1: Interesting Facts 30k+ screenshot #1

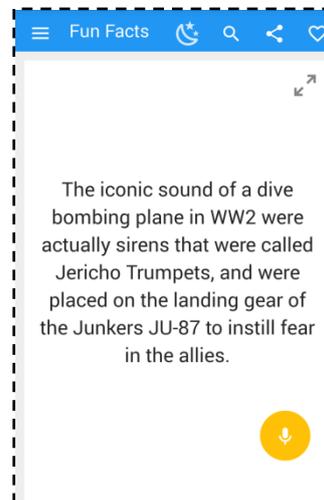


Figure 2: Interesting Facts 30k+ screenshot #2

Table 1: Pros and Cons of *Interesting Facts 30k+*

PROS	<ul style="list-style-type: none"> • Apparently great number of facts • Easy to use • Day/Night design • Search engine
CONS	<ul style="list-style-type: none"> • Not interactive • Poor design (no photos) • Spelling mistakes

Funny facts 8000+

Company: srsdev

Downloads: +1.000.000

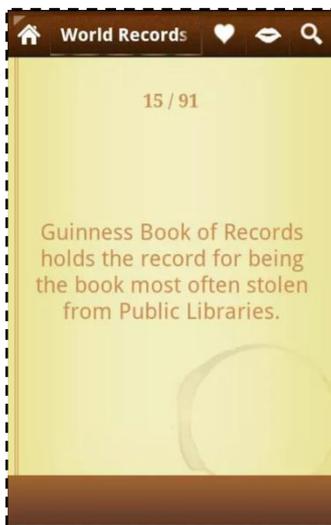


Figure 3: *Funny facts 8000+* Screenshot #1



Figure 4: *Funny facts 8000+* Screenshot #2

Table 2: Pros and Cons *Funny facts 8000+*

PROS	<ul style="list-style-type: none"> • Easy to use • Search engine
CONS	<ul style="list-style-type: none"> • Not interactive • Poor design (no photos) • Design not appealing

To improve these applications it has been chosen two methods: DMAIC and Axiomatic Design. The first one, DMAIC, was selected because is a methodology that is vastly used in big companies that implemented Six Sigma and has proven that it can be a very useful tool to improve processes. Although this theory is meant to solve problems, not creating new projects, the new application is based on existing ones so it is like improving processes or solving existing problems.

DMAIC and Axiomatic Design are methodologies that have been implemented in many companies so here there are some examples of it:

General Electric implementation is one of the best examples to explain how sig sigma and DMAIC can help a company to improve its processes. By mentoring, training and leadership, GE has accomplished one of the main levels of Six Sigma commitment. (GE, 2013)

Allied Signals was an American aerospace, automotive and engineering company that implemented Six Sigma and obtained very significant gains such as saving of 1,5 billion \$ since 1994. (isixsigma.com)

The table above shows how Six Sigma has produced enormous economic outcomes to big companies that implemented it, (isixsigma.com):

Table 3: Six Sigma Costs and Savings by Company

Year	Revenue (\$B)	Invested (\$B)	% Revenue Invested	Savings (\$B)	% Revenue Savings
Motorola					
1986-2001	356.9(e)	ND	–	16	4.5
Allied Signal					
1998	15.1	ND	–	0.5	3.3
GE					
1996	79.2	0.2	0.3	0.2	0.2
1997	90.8	0.4	0.4	1	1.1
1998	100.5	0.5	0.4	1.3	1.2
1999	111.6	0.6	0.5	2	1.8
1996-1999	382.1	1.6	0.4	4.4^g	1.2
Honeywell					
1998	23.6	ND	–	0.5	2.2
1999	23.7	ND	–	0.6	2.5
2000	25.0	ND	–	0.7	2.6
1998-2000	72.3	ND	–	1.8	2.4
Ford					
2000-2002	43.9	ND	–	1	2.3
Key:					
\$B = \$ Billions, United States					
(-) = Estimated, Yearly Revenue 1986-1992 Could Not Be Found					
ND = Not disclosed					
Note: Numbers are rounded to the nearest tenth					

In contrary to the vast history DMAIC has shown in solving these kinds of problems, Axiomatic Design Theory has not been applied that frequent. However, (Brown, 2011) explores

axiomatic design as an approach to designing for surface integrity, particularly in regard to topography for rotating lip seals. The Domains in this project are described in **figure 5**.

Domains→	Functional (FRs)	Physical (DPs)	Process (PVs)
1	<i>Resist flow</i>	<i>Sealing topography</i>	<i>Fine honing</i>
2	<i>Promote counter-face life</i>	<i>Protective topography</i>	<i>Plateau honing</i>
2.1	<i>Resist abrasive wear</i>	<i>Contact zone features</i>	<i>Fine honing</i>
2.2	<i>Provide lubrication</i>	<i>Lubricant pockets</i>	<i>Rough grinding</i>

Figure 5: Decomposition of the surface of a rotating shaft for interfacing with a lip seal (**Brown, 2011**)

As described before in the objective of the project, the implementation of DMAIC and Axiomatic Design to the creation of a basic smartphone application aim is to show if these two theories can be applied to very small software company.

The main difference with the usual DMAIC applications to companies and this work is that in this project the complaints and errors are taken from the existing applications.

The work will be structured as the theories explain; DMAIC will be applied using the tools proposed, as diagrams and tables; the same for Axiomatic Design Theory, using the domains theory and then the axioms.

1.3 Start-up

(Ries, 2011): “A Start-up is a human institution designed to create a new product or service under conditions of extreme uncertainty.”

So applying DMAIC and Axiomatic Design to a Start-up doesn’t ensure at all that it is going to make a great change and provide large gains in a short time. However this project is meant to show that DMAIC and ADT theories can help to build a solid company and to create successful products.

The Start-up structure itself will not be widely explained in this project, because it will be focused in one product, not the company, even so the basic structure of the company would be like shown in **Table 4**:

Table 4: Company structure

Antoni Amat	Gerdard Casaus	Alejandro Martín
Co-president and COO (Chief Operations Officer)	Co-president and CTO (Chief Technology Officer)	CVO (Chief Visionary Officer) and Lead Developer

1.4 Approach to DMAIC

For this project it has been chosen DMAIC because it is proven to be a very helpful tool to make a project worthwhile and with minimum failures. DMAIC basically improves existing processes and sustains improvements over the long-term. Six Sigma is the main set of techniques that uses DMAIC and it has been implemented in several important companies such as Motorola, Texas Instruments, Sony and a lot more.

DMAIC is formed by 5 phases as shown in figure 2: Define requirements, Measure performance, Analyze opportunity, Improve performance and Control performance, each and every part is vital to perform a DMAIC project. The following sections are dedicated to the detailed explanation of the steps.

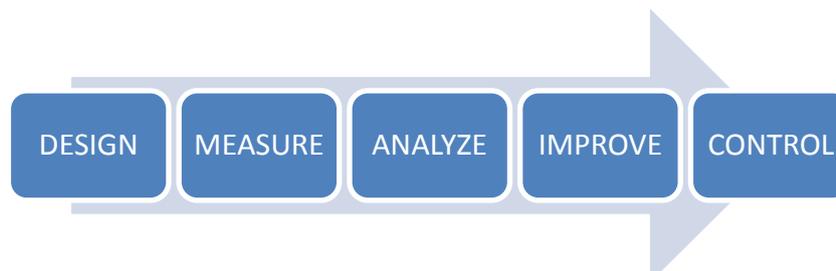


Figure 6: DMAIC procedures

1.3.1 Define

Define part first focuses on how to specify the project that will be structured: the team, budget, requirements, and etcetera. To define requirements, there are three main activities: Problem Statement, Voice of Customer (VOC) and Processing Mapping as shown in the **Figure 7**.

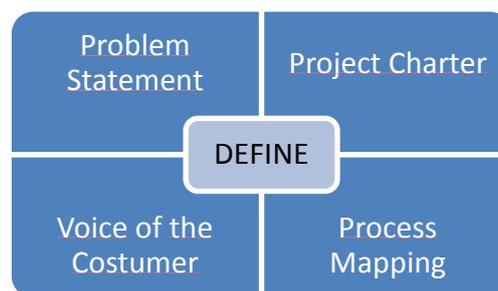


Figure 7: Define procedures

The problem statement should be a brief explanation of the issue that is intended to be improved by the project. It has to use available facts and it has to avoid questions and potential causes of the problem.

The project charter is a document where the most important issues to consider before the project are written, such as the team, the budget or the deadlines.

The Voice of the Customer (VOC) is one of the main parts of DMAIC, the aim is capturing customers' expectations, preferences and aversions. In this step the team seeks what the customers want: quality, cost, delivery, service and safety and corporate responsibility.

The last main item to complete when defining opportunities is the process mapping, using flow charts. The flow chart is a tool to diagnose activities and information flow.

1.3.2 Measure

The purpose of this step is to objectively establish current baselines as the basis for improvement. The **Figure 8** shows the work structure of the Measuring part of DMAIC.

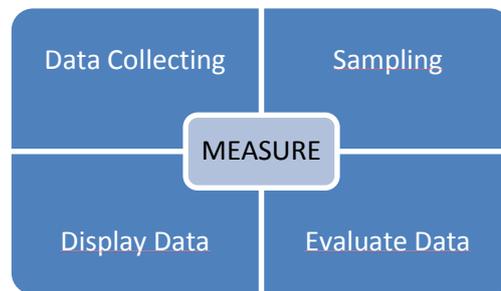


Figure 8: Measure procedures

The Data Collecting has to be coupled with the developing of a Measurement Plan to ensure the data is valid, meaningful and the relevant data is collected. Thus, it has to be done by operational definition to provide everybody with the same meaning, ensure that consistency and reliability are built in up front and describe the scope of the measure.

Sampling part is crucial because it has a big impact on the data results so it has to be chosen very carefully. The sampling is conducted by statistical parameters that allow measuring only a part of the whole population data without missing important details.

The displaying and evaluation part is completed by drawing charts that show the errors of the measuring. The most important charts used are: the histogram, the run and the Pareto.

1.3.3 Analyze

This part of DMAIC is meant to identify root causes, by the analyze part it can be identified and then eliminated. **Figure 9** shows the activities carried in the Analyze part of DMAIC:

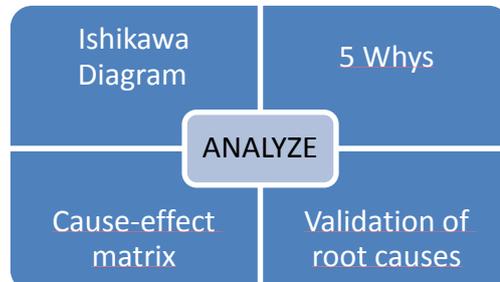


Figure 9: Analyze procedures

The Ishikawa diagram is a causal diagram to establish and clarify the relationships between an effect and its main causes. The system used to identify the root causes is the 5 Whys, that suggests that by asking five times why is a problem occurring, it is possible to find the main cause.

Cause-effect Matrix is used to relate the key inputs and outputs, and is a simplified QFD (quality function deployment) matrix. The purpose of the Cause-effect Matrix is to find what inputs have the most impact on the problems to solve.

The last part of the analysing step is the validation of the root causes, which will be done by correlation and regression analysis, hypothesis test and ANOVA (analysis of variance).

1.3.4 Improve

In this phase, the team will target their efforts to seek for solutions for the problem. In **Figure 10** can be seen the three phases of the Improve phase of DMAIC.

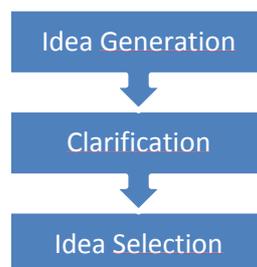


Figure 10: Improve phases

The first part of this phase will be generating many ideas about how to eliminate the root causes found in the prior step. To generate these ideas there is a great deal of techniques such as brainstorming, solution mapping or mind mapping. These techniques will be also used to the selection of ideas

1.3.5 Control

Control is last step in the DMAIC process but is a process that will endure over time; the team will be monitoring the development of the project to ensure continued and sustainable success.

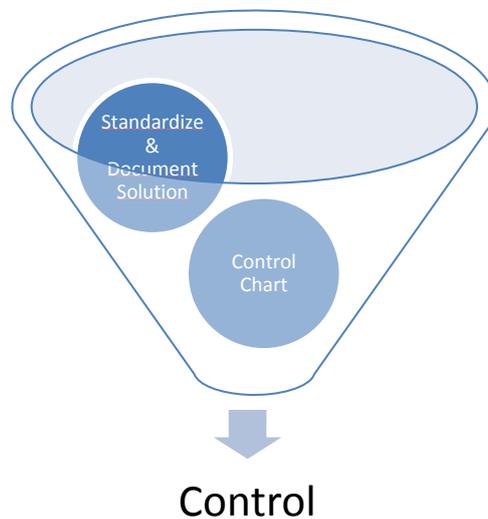


Figure 11: Control procedures

The first phase consists in standardising the solution by implementing it and monitoring it to maintain the expected results.

Then the team will work on the control charts to tell whether the process is working or it has errors. These charts will help the team to identify future root causes and variations.

1.5 Approach to Axiomatic Design

The Axiomatic Design theory is a systems design methodology applicable to any type of system that uses matrix to link customers with functional requirements, design parameters and process variables (Suh, 1999).

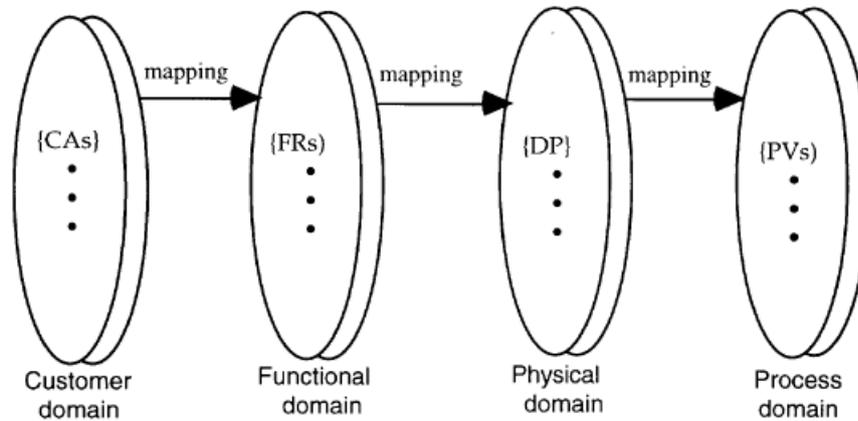


Figure 12: Domains, Axiomatic Design Theory for Systems, (Suh, 1999)

This theory takes its name (Axiomatic) from the two axioms that rule the whole process:

Table 5: Axioms in Axiomatic Design Theory

AXIOM 1: The Independence Axiom	AXIOM 2: The Information Axiom
<ul style="list-style-type: none"> • Maintain the independence of the Functional Requirements(FRS) 	<ul style="list-style-type: none"> • Minimise the information content of the design.

Moreover, this theory has also Corollaries and Theorems referring to more specific areas so are not applicable to every type of system.

The first part in this theory is the domains, the domains of the design world as defined for (Suh, 1999) in **Figure 12** shows the interaction and process from what the customer wants to the processes itself.

There are 4 domains: Customer domain, Functional Domain, Physical Domain and Process Domain.

The customer domain is where the customer attributes are placed; it basically means the characteristics the client is expecting of the product.

The functional domain is the part of the project where the capabilities of the project are taken into account, so here the team is considering what the product has to be able to do.

The physical domain is the part when the design takes part, when the team has to design the product to satisfy the functional requirements.

The process domain is the deepest part of how the team purpose to achieve the customer attributes. (Brown, 2011)

The theory itself is meant to connect these domains to finally meet what the customer wants and what are the process variables.

$$\{FRs\} = [A] \{DPs\}$$

Equation 1: FRs and DPs relation (Suh, 1999)

The relationship between Functional Requirements (FRs) and Design Parameters (DPS) can be written as in **equation 1**, [A] is a matrix defined as the design matrix that characterizes the product design (Suh, 1999).

2. IMPLEMENTATION OF THE THEORIES

2.1 DMAIC Implementation

2.1.1 Define

The parts above will be the practical implementation of the Design procedures for DMAIC, the work is structured in four activities: the problem statement, the project charter, VOC and the process mapping.

2.1.1.a Problem statement:

First and foremost when conducting a DMAIC based project, the problem that is wanted to solve must be very clearly defined, for this reason the first procedure will be writing the problem statement.

Problem Statement:

A “Fun Facts” App should be very easy to use and with a very appealing design. But not only has it to be with the look but also about how the user interacts with it, the App should test the user to increase the engaging of attention.

Plus they need something to make these fun facts a challenge for the users to keep them using the App. A system to get the user curious about the matter is to create a “True or False” system with a brief explanation after the user’s answer.

What is intended to do with this project is to design a very basic app but solving the problems the other fun facts app have, and with a team of only three people.

A “Fun Facts” App should be very easy to use and with a very appealing design. In addition to the necessity of a good look for the app, how the user interacts with it is very important, the app must be able to attract the customer’s attention and make him engage with the app But not only has it to be with the look but also about how the user interacts with it, the App should test the user to increase the engaging of attention.

The existing Apps based on “Fun facts” are far from being able to keep the user interested in using it for during a long time and their interfaces are not suitable for the users.

In addition, they need something to make these fun facts a challenge for the users to make them keep using the App. One system to make the user curious about the matter is to create a “True or False” system with a brief explanation after the user’s answer.

What is intended to do with this project is to design a very basic app but in which the existing problems are solved, and with the help of a team made of three.

2.1.1.b Project Charter:

After the problem statement, the next phase is writing the project charter so the team define the main limitations and resources that take part in the project.

Project Charter: Creation of an improved “Fun facts” App 05.23.2017

Background
This project has been undertaken to show how basic changes can make an App a lot more appealing to the user.

Goals

- Creating an appealing design.
- Improving the interface of the existing Apps.
- Introducing a challenge to the users when using the App.

Scope
The scope is to be able to implement the basic improvements based on the existing and launch the App. Marketing or sponsoring will not be included in this project.

Key Stakeholders

Project manager	Antoni Amat
Project team members	Gerard Casaus, Alejandro Martín

Project Milestones
The project start date will be the 1st July and the launching would be the 1st October.

Project Budget
The project budget is zero, all three members of the staff will be rewarded depending on the success of the App and the software used to create the App will be free.

Constraints, Assumptions, Risks and Dependencies

Constraints	The number of “Fun Facts” has to be high so the work on finding them may represent an uncertain amount of research time.
Assumptions	Because of the low level of complexity of the App there will be no problem when referring to the code.
Risks and Dependencies	The only risk of this project is that the App could be never launched.

Approval Signatures



Antoni Amat, Project Manager

Figure 13: Project Charter

Figure 13 is the document defined as the project charter and it has been built with a template from (casual.pm).

The structure of this document is not always the same but some points must be defined compulsory, in this case it has been chosen these parts:

Background: Here is where the writer explains why this project is undertaken.

Goals: In this point it has to be written the goals that the team will try to reach.

Scope: Here is where the team must define the outcome that the project is meant to have.

Key Stakeholders: This part is where all the key people involved in this project have to be defined.

Project Milestones: The team defines in here the dates of important events like the start of the project or the due date.

Project Budget: Here the team will write the money that they can use to do the project.

Constrains: Here is where the team will write the factors that will have an impact on the delivery of the project.

Assumptions: Here is where is written all the things that the team rely on in order to carry out the project.

Risks and Dependencies: In here it is written what the team thinks that could happen before the finishing of the project.

The project charter is a very helpful tool to see if the project is viable and to define the most important parts.

2.1.1.c VOC(QFD):

Voice of the Customer (VOC) is the dialog in all forms that takes place between us and our customer. (Montagna)



Figure 14: VOC diagram (Montagna)

In VOC part it is used HOQ diagram: the house of quality. This diagram is meant to identify customer requirements and have a very clear structure of the demands, company product's characteristics and also the ones of the competence, all related.

Below in **figure 15** it is a HOQ diagram drawn with the help of template from (QFDonline.com, 2011).

The Data Shown in this chart has been taken from the comments of the users of the existing app that in this project are studied.

This HOQ diagram has a lot of information in it so in the following paragraphs the parts of the HOQ chart are explained more in detail.

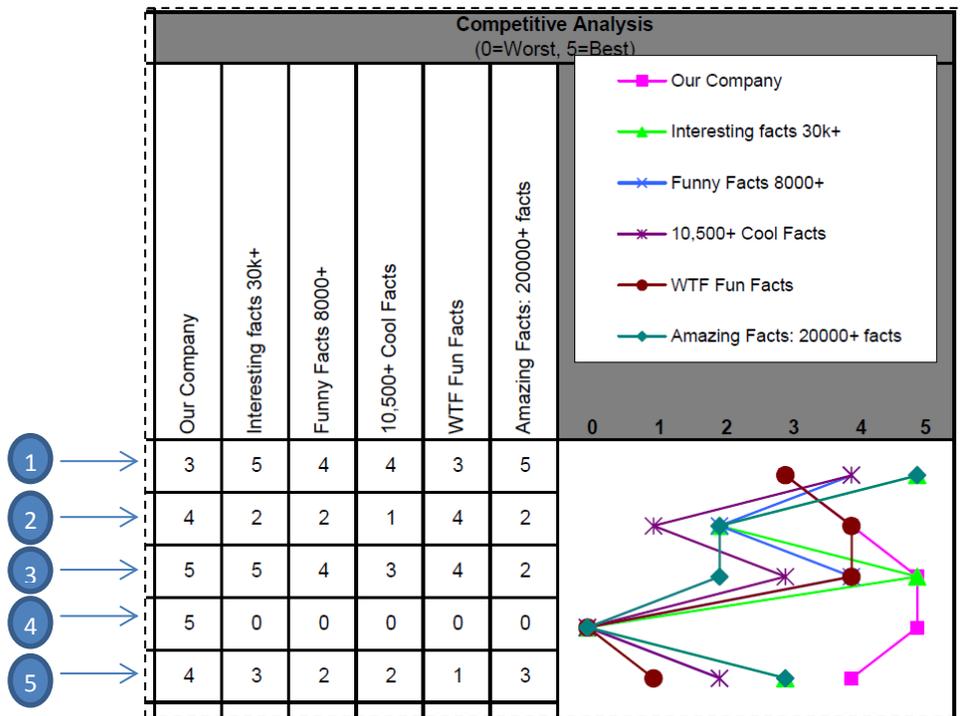


Figure 16: HOQ Competitive Analysis

In the competitive analysis part of the HOQ (**figure 16**) the team is able to see how the product quality is in comparison with the products of the company. The analysis of “our company” is the quality that the company expects of the new app that will be created.

In this HOQ the quality demanded from the customers are:

- (1) **Amount of facts:** the number of facts the app has in its library.
- (2) **Attractive Appearance:** if the app has an appealing design so the customer sight is not disturbed.
- (3) **Easy Usability:** The app must be easy to use so the customer only effort is to guess the answer of the game.
- (4) **Entertaining features:** The app has to have an easy game in it that make the customer more entertained when using the app.
- (5) **Long life frame:** This refers to the amount of time the customer will have the app on the phone and use it periodically.

The application built in this project would have only one characteristic below the other applications, the number of facts. It is because it can cause a lot of problems and delays to try to have too big library of facts, and it is unnecessary, with only 1000 facts more or less the customer will not be annoyed for the repetition of them, because of the entertaining features.

Row #	Max Relationship Value in Row	Relative Weight	Weight / Importance	Demedanded Quality (a.k.a. "Customer Requirements" or "Whats")	Column #				
					1	2	3	4	5
					Direction of Improvement: Minimize (▼), Maximize (▲), or Target (x)				
					Quality Characteristics (a.k.a. "Functional Requirements" or "Hows")				
					plus 1000 facts	Clear interface	Challenging approach	Simple structure	appealing Design
1	9	15,8	6,0	Amount of Facts	⊖		▲		
2	9	18,4	7,0	Attractive Appearance		○		▲	⊖
3	9	18,4	7,0	Easy Usability		○		⊖	
4	9	26,3	10,0	Entertaining Features			⊖		
5	9	21,1	8,0	Long lifeframe	○	▲	⊖		▲

Figure 17: HOQ Quality Characteristics

This part of the HOQ is meant to relate the quality demands with the quality characteristics. It is done with signs to show the strength of the relation between to terms:

⊖	Strong Relationship	9
○	Moderate Relationship	3
▲	Weak Relationship	1

Figure 18: HOQ Relationship strength

In this part is also defined the importance of every Customer Requirement, in this case it has been done from 0 to 10.

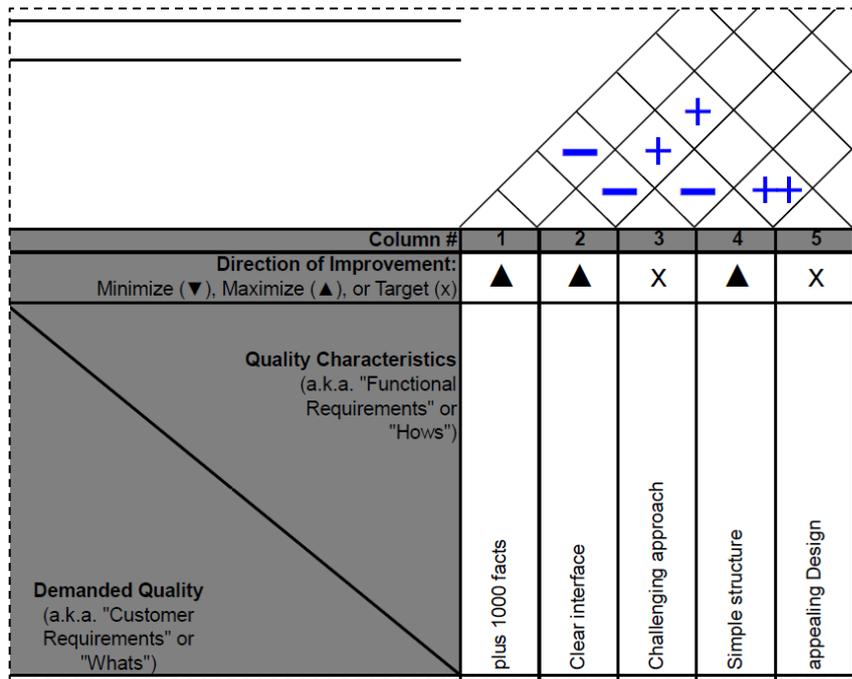


Figure 19: HOQ Correlation between QCs

Figure 19 shows the QFD detail of the correlation between the Quality Characteristic (QC) and the Direction of Improvement of each QC.

The directions of the improvement can be only three: Minimize, Maximize or Target; and they are defined with the symbols in Figure 20.

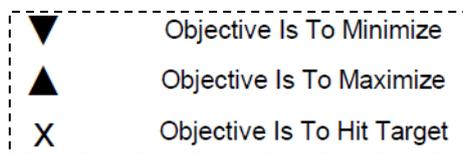


Figure 20: HOQ Directions of Improvement Symbols

The correlation between Quality Characteristic is a very useful part of the HOQ because it shows how a QC improvement can become an improvement for another QC. This is also defined with symbols such as in Figure 21.

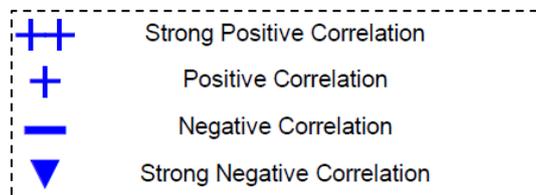


Figure 21: HOQ Correlation Symbols

The last part of the HOQ diagram is the results taken from the relations described inside the chart, as seen in **Figure 22**:

Column #	1	2	3	4	5
Direction of Improvement: Minimize (▼), Maximize (▲), or Target (x)	▲	▲	X	▲	X
Quality Characteristics (a.k.a. "Functional Requirements" or "Hows")					
Demanded Quality (a.k.a. "Customer Requirements" or "Whats")	plus 1000 facts	Clear interface	Challenging approach	Simple structure	appealing Design
Amount of Facts	⊖		▲		
Attractive Appearance		○		▲	⊖
Easy Usability		○		⊖	
Entertaining Features			⊖		
Long lifeframe	○	▲	⊖		▲
Target or Limit Value	3000				
Difficulty (0=Easy to Accomplish, 10=Extremely Difficult)	2	3	6	2	4
Max Relationship Value in Column	9	3	9	9	9
Weight / Importance	205,3	131,6	442,1	184,2	186,8
Relative Weight	17,8	11,4	38,4	16,0	16,2

Figure 22: HOQ Results

In the HOQ of this project it has been seen that the Quality Characteristic that has more weight is the Challenging Approach that refers to the creation of a True/False game.

2.1.1.d Process mapping:

Process mapping is the last part of the Define phase of this project, and it has been carried out using two types of charts: SIPOC diagram and Flow Chart. The pages below explain these two charts including some examples.

SIPOC Diagram

A SIPOC (Suppliers-Inputs-Process-Outputs-Customers) diagram is a tool used in the part of process mapping to help defining the project by determining every part of the processes (sixsigma.com). SIPOC takes its names from the parts of any process defined in the diagram: Suppliers, inputs, process, outputs and customers. In some cases, requirements of the customers can be appended to the end of the SIPOC for further detail.

The SIPOC is used to clarify who supplies inputs to the process, what specifications are placed on the inputs, which the true customers of the process are or what the requirements of the customers are.

The SIPOC diagrams below are describing the main processes this project is focused on:

1. Process name: *Create a library of facts*

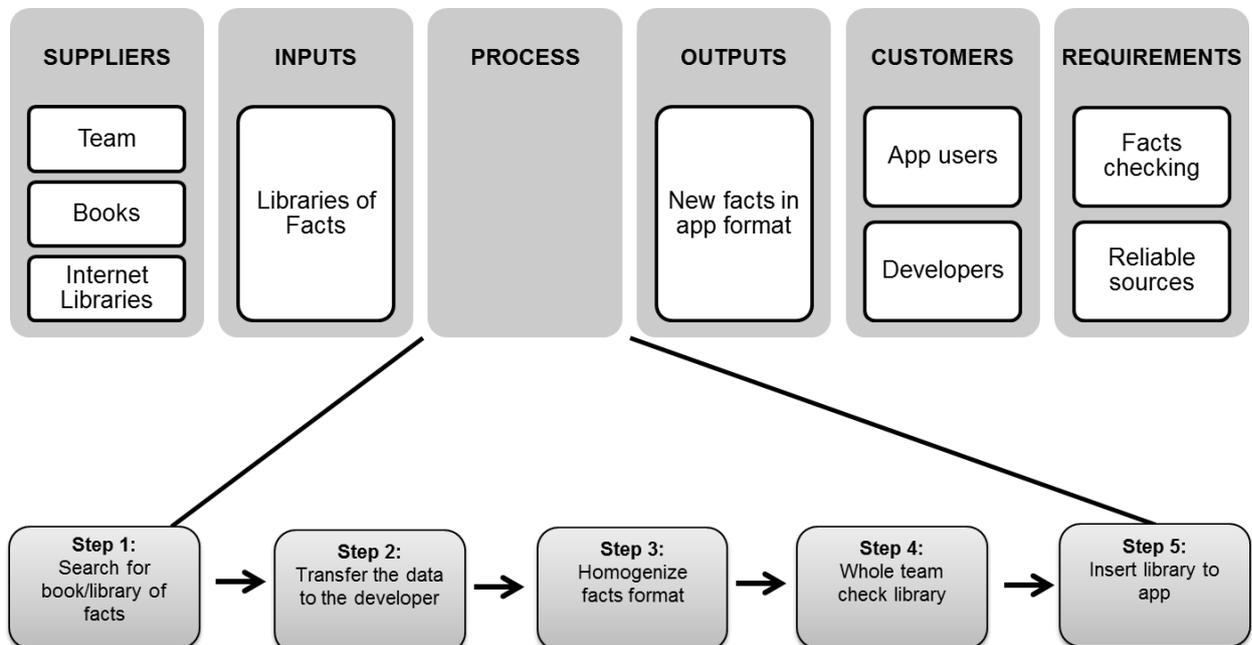


Figure 23: SIPOC Diagram #1 Create a library of Facts

Figure 23 is the SIPOC diagram of the creation of a library fact process. The library of facts refers to all the facts that are inside the app.

The **suppliers** of this process are the sources where the team will find the facts, such as books, internet libraries or even the team members' knowledge.

The **inputs** in this process are the libraries of facts without the app format and not in the app library yet.

The **process** itself has five steps:

Step 1 - Search for Book/Library of facts: in this phase the team will search for facts of any type and in any format.

Step 2 - Transfer the data to the Developer: Here the facts that are found will be transferred to the person in charge of the changing of format.

Step 3 - Homogenize facts format: This is step in which the person in charge of the format will change every fact to the desired format

Step 4 - Whole team check library: when the facts are already in the same format the whole team will check if the new facts are reliable.

Step5 - Insert library to the app: when facts are already in the good format and checked the developer will import the facts to the app.

The **Outputs** in this diagram are the new facts in the desired format so it can be used in the app.

The **Customers** here are the app users because in the end are who are going to see the outcomes but also the developer that receive the new library to import inside the app.

The **Requirements** in this process are the checking of the reliability of the fact and also finding reliable sources to not lose all the time checking the veracity of the fact.

2. Process name: **Define the Application Design**

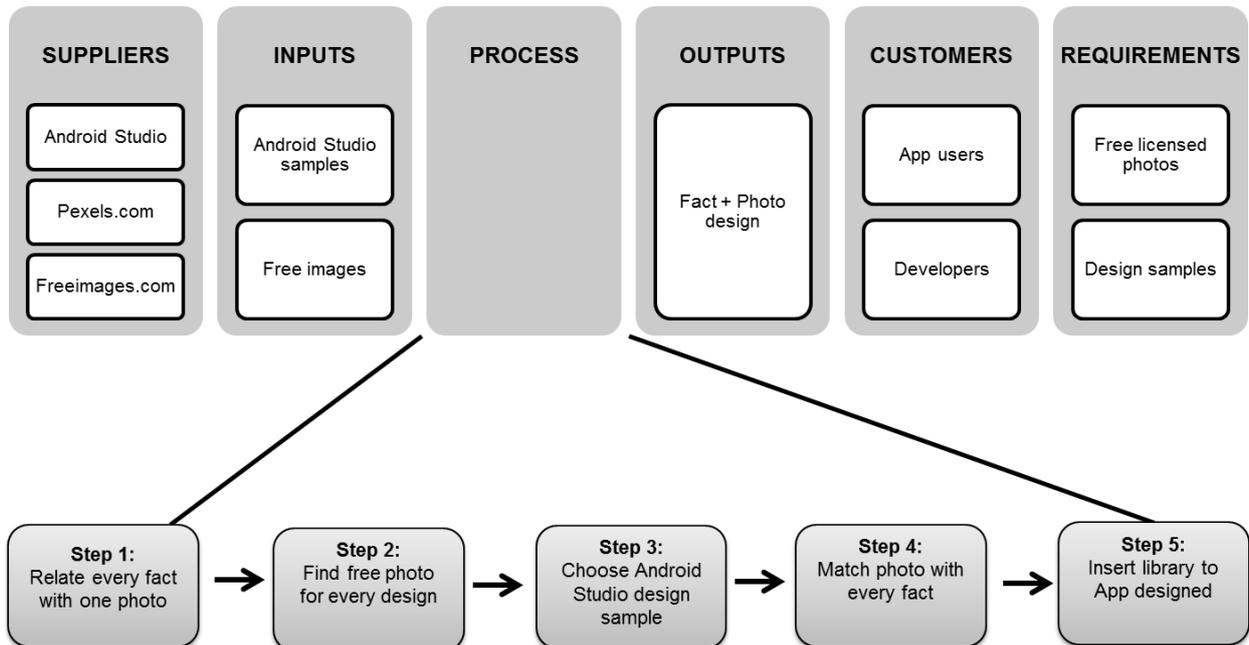


Figure 24: SIPOC Diagram #2 Define the App design

Figure 24 is the SIPOC diagram of the Definition of the app design. The app design is here what the user will see and how the app is organise, its structure and how facts will be shown to the user.

The **suppliers** of this process are the platforms that will help the developer and the rest of the team to build the app design. The app will be made by Android Studio that provides the developer with templates to shorten the coding time. To download the photos for every fact will be used two websites: (Pexels.com) and (Freeimages.com) because they provide images without copyright and free use.

The **inputs** in this process are the Android Studio Templates and the free images that can be downloaded from the two websites written in the inputs.

The **process** itself has five steps:

Step 1 - Relate every fact with one photo: In this phase the team will think of what kind of photo relate with every fact.

Step 2 - Find free photo for every design: In this phase the team will do a research work in the websites providing free photos to find one photo for every fact.

Step 3 - Choose Android Studio Design sample: In this steps the team will decide the main design of the app, with the help of the Android Studio templates.

Step 4 - Match photo with every fact: When the images are already downloaded they have to be linked with every fact.

Step5 - Insert library to the app designed: Here when every fact is linked with a photograph it has to be all imported to the app.

The **Outputs** in this diagram are the facts with the photographs and the design of the app.

The **Customers** here are the same as the first diagram: app users because in the end are who are going to see the outcomes but also the developer that receive the new library to import inside the app.

The **Requirements** in this process are the free licensed photographs to link with the facts and the Android Studio Templates.

3. Process name: **Create True/False System**

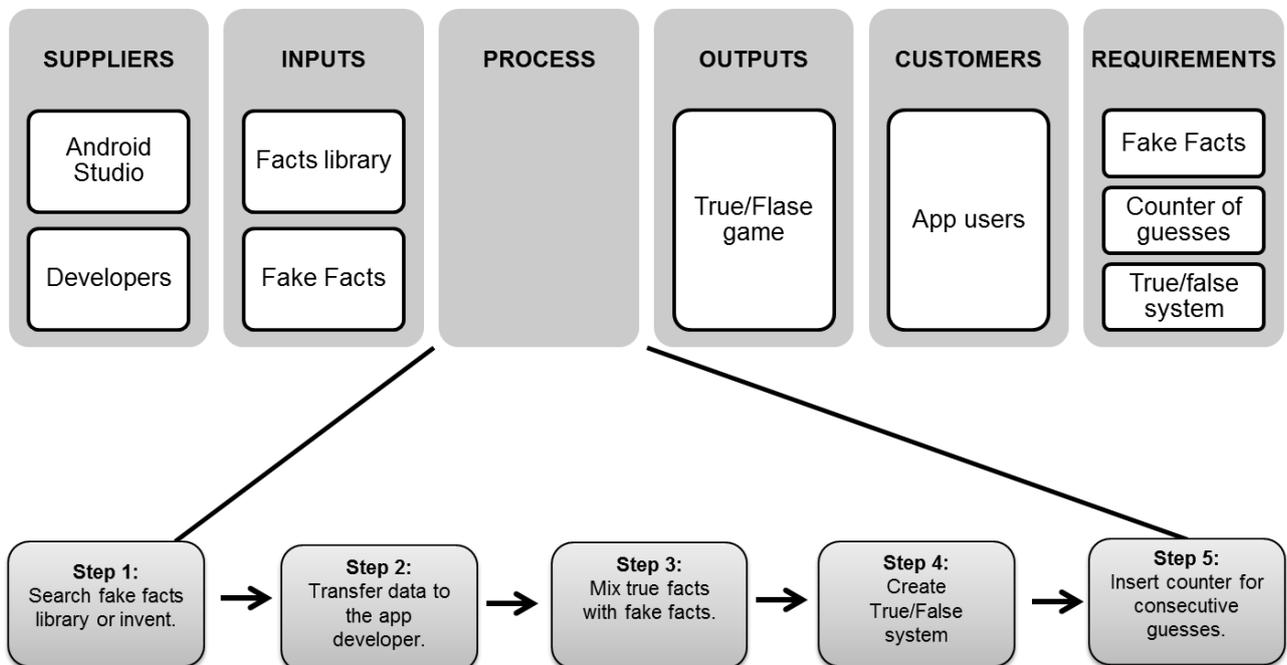


Figure 25: SIPOC Diagram #3 Create True/False System

Figure 25 is the SIPOC diagram of the creation of the True/False system. This means how the team will manage to introduce the new feature to make the users more interested: a True/False game in every fact.

The **suppliers** of this process are the Android Studio templates, because it can also be useful when creating a game; and also the developers because they will design the game and built it.

The **inputs** in this process are the facts library and the fake facts that will also be introduced in the app for the game.

The **process** itself has five steps:

Step 1 - Search fake facts library or invent: In this phase the team will search for facts that are fake or simply invent it. Another strategy is to use true facts and change them a little to make them fake.

Step 2 - Transfer data to the app developer: In this phase the team will send the data to the developer to have the fake facts in the app library.

Step 3 - Mix true facts with fake facts: Here the developer will mix the true and fake facts in the library but in some format that the fake facts can be detected by the game.

Step 4 - Create True/false system: The developer will create the game with the help of the Android Studio template.

Step 5 - Insert counter for consecutive guesses: For getting more attention from the user the game has to be competitive so the developer will include a counter of good answers.

The **Output** in this diagram is the game itself.

The **Customers** here are only the app users that will play the game.

The **Requirements** in this process are the Android Studio Templates to create the game, the fake facts to complete the library and the consecutive guesses counter.

Flow Chart

This type of chart illustrates the sequence of operations performed to reach a solution. It is a very helpful solution to improve the communication between programmers and the rest of the team, since is a language between the code and business documents writing.

The chart below, in **Figure 26**, is the representation of the process of finding Facts to introduce it to the app library:

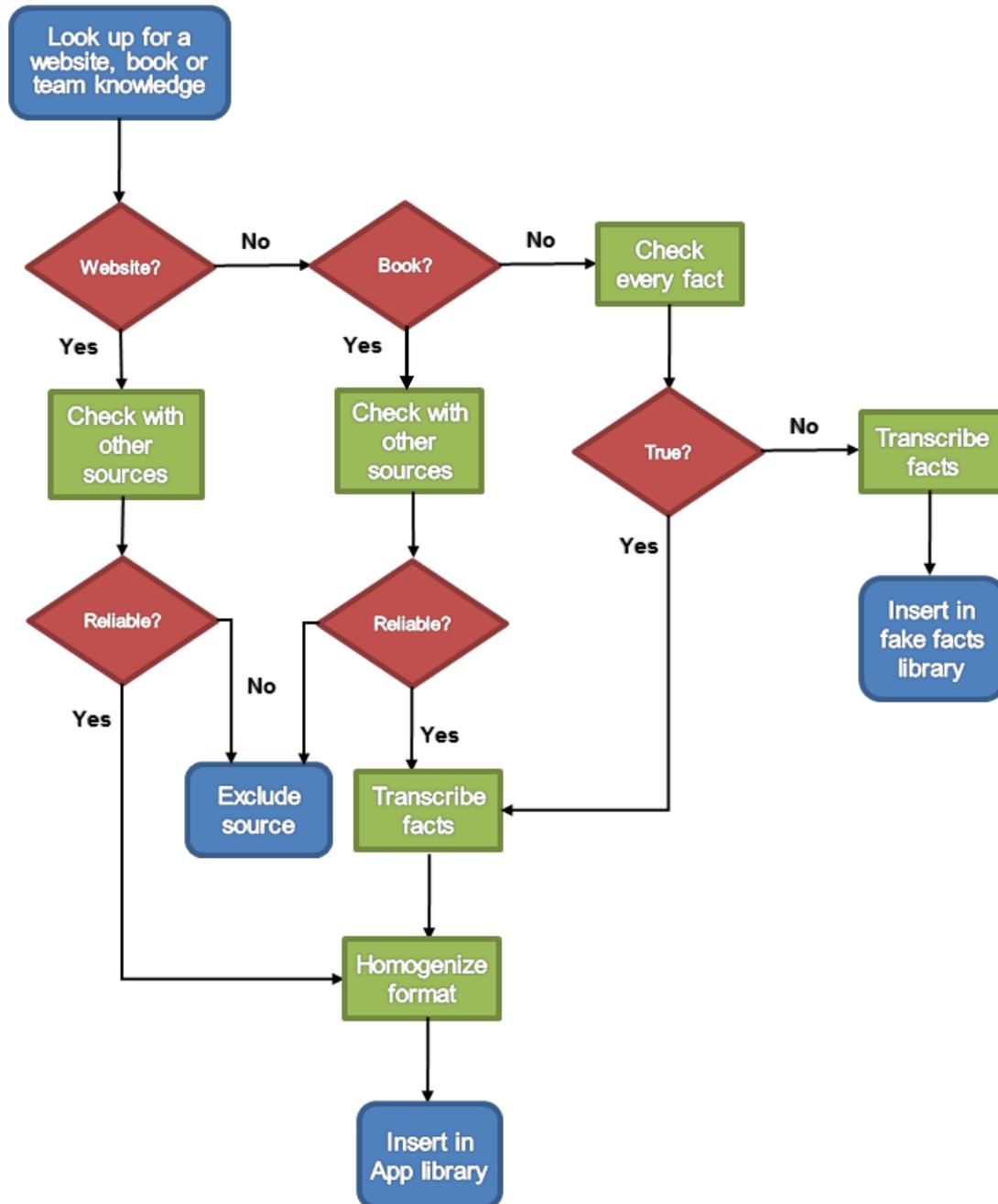


Figure 26: Flow Chart of finding new facts

2.1.1.e Define phase Outcomes

The first step of DMAIC, Define phase, is it very useful for make clear the purpose of the project and how the team is going to face every part of it.

In the case of this project it has been very helpful to know what can be the problems that the process of creating the app can have. But also it has provided a lot of clearance about how the work can be organized to avoid these possible problems.

By using the Define sources of DMAIC it is a lot easier to choose the data to the Measure part, in the following pages.

2.1.2 Measure

The measurement in this project will be focused on the opinions the users of the competitor’s applications, because the app has not been yet created so it is not possible to measure the errors that it has. However some of the measurement will be hypothetically made because it can be also helpful to find root causes afterwards

The measure phase of DMAIC has three main parts: the measurement plan, the sampling part and finally the displaying and evaluation of the data collected.

2.1.2. a Data collection plan:

When developing a project with DMAIC procedures, it is very important that the data collected for the measuring is answering the right questions that the team members must ask themselves (isixsigma.com).

The table above, **Table 6** explains the data collection plan:

Table 6: Data collection plan

DATA COLLECTION PLAN					
Measure	Data Type	Operational Definition	Stratification Factors	Sampling Notes	Who and How
Number of downloads	Discrete	Number of downloads that an App has.	By App By launching date		Project leader looking for the statistics in the Google play store.
Number of Facts	Discrete	Number of facts in the App library.	By App		Project leader by looking in the app information.
Satisfaction	Discrete	Rating of the costumer about the App.	By App		Project leader looking for the statistics in the Google play store.
Time for inserting one fact to the library	Continuous	Cycle time from the moment the team member find a new fact until the fact is in the app library.	By type of fact (true/false)	Sample every tenth fact	Developer with a timer.
Time for getting one photo for every fact	Continuous	Cycle Time from the moment the Fact is in the library until it is matched with a photo.	None	Sample every tenth fact	Developer with a timer.

The **Table 6** defines 4 characteristics of every measurement:

Data type: the data has to be classified in two types, continuous or discrete. Continuous data is when it is measured on the continuum and it is subjective; Discrete measuring is when it is a count or a qualitative measurement.

Operational Definition: a precise description of the specific criteria used for the measures (the what), the methodology to collect the data (the how), the amount of data to collect (how much), and who has responsibility to collect the data (the who), (Montagna).

Stratification factors: if the population of the measurement is going to be divided, by what factors.

Sampling notes: the sampling is how the team is going to choose the data to collect.

Who and how: this is referring to who will be the person to measure the data and by what methods will it be done.

2.1.2.b Sampling:

Using a sample of data you draw conclusions about the entire population of data. This is known as “statistical inference.” Sampling saves costs and time. Sampling provides a good alternative to collecting all the data. Identifying a specific confidence level allows us to make reasonable business decisions (Montagna).

The sampling part in this project will be made to sample the number of facts that will have to be measured in a total of 3000 facts to analyze the time used to import this facts in the app library and the time used to link them to a photograph.

The formula used to know how many of the total of facts must be measured is in **Equation 2:**

$$n = \left(\frac{1.96s}{\Delta} \right)^2$$

Equation 2: Minimum Sample size required (Montagna)

Variables in **Equation 2:**

n: it refers to minimum sample size required, how many test will have to be done.

s: standard deviation of the process data, approximate.

Δ: level of precision desired in the same units as the “s” measurement.

1.96: constant representing a 95% confidence interval.

To approximate the standard deviation it has been done an approximation on what the measures could be in the two cases: Time for introducing fact and Time for photo in fact, as seen in **Table 7**.

Table 7: Hypothetical measurements for the sampling

Time for introducing fact (min)	Time for photo in fact (min)
2	5
2	10
10	15
0,5	40
1	15
2	10
8	5
4	6
2	4
0,5	45
0,5	10
0,5	5
2	4
1	31
1	22
9	21
6	25
0,5	16
3	17
0,5	2

Table 8: Sampling results

Δ	s	n
1	3,0	35,1
3	12,2	34,6

In the **Table 7** are shown the sampling results it will have to be measured 35 facts for every measurement, approximately every tenth fact in a library of 3000 facts.

The next after knowing how many facts have to be measured is the measuring displaying and evaluation of the data, shown in the next point.

2.1.2.c Displaying and evaluation of the Data

The data measured in this project is how much time it takes to realise two tasks: introducing a fact to the library and the time for linking it to a photo; it has also been measured the quality of the existing apps of the same type to see how they can be improved.

Table 9 shows the measuring of the times for the two tasks; it has been measured 35 times, as the results of the sampling.

Table 9: Continuous Measuring

Time for introducing fact (min)	Time for photo in fact (min)
2	5
2	10
10	15
0,5	40
1	15
2	10
8	5
4	6
2	4
0,5	45
0,5	10
0,5	5
2	4
1	31
1	22
9	21
6	25
0,5	16
3	17
0,5	2
3	12
2	15
0,5	45
1	41
1	20
1	23
1	15
5	17
8	26
2	15
0,5	18
3	19
1	12
2	8
2	10

Table 10: Statistical basic indices

mean	2,5	17,3
median	2	15
mode	2	15

Table 10 shows the main three indices to measure the centre of the data that can help to see what the most common results are.

For the displaying of the data measurements it has been chosen first the histogram so it is a very clear graph to see how high the levels of the data obtained are and how different are between each other.

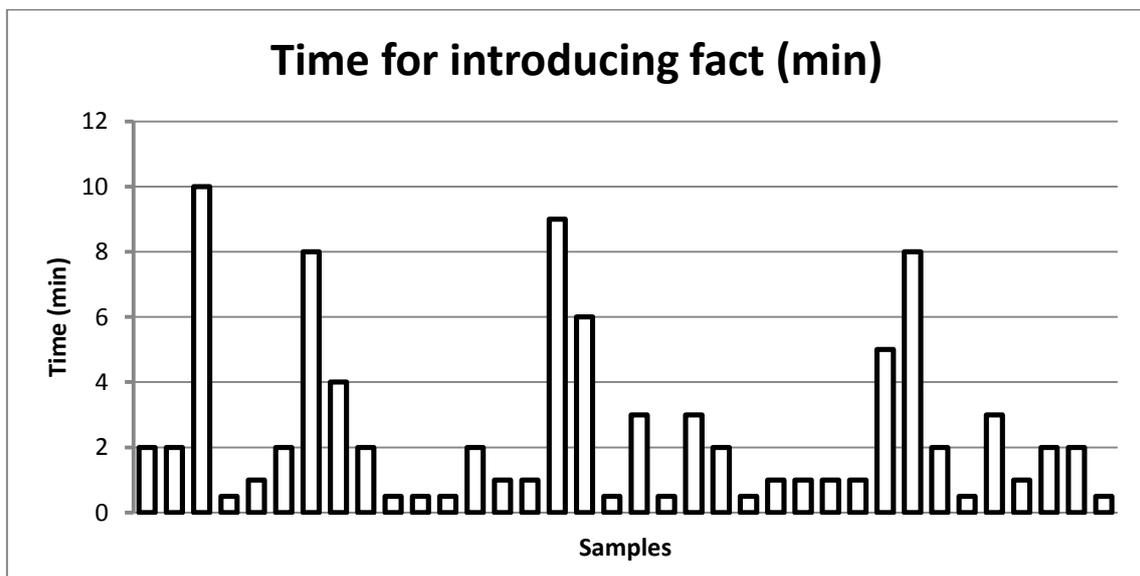


Figure 27: Histogram #1: Time for introducing a fact

Figure 27 is the histogram that shows the data measured about the amount of time the team spends when introducing a new fact in the library of the app. It can be seen that 5 samples are out of the normal numbers.

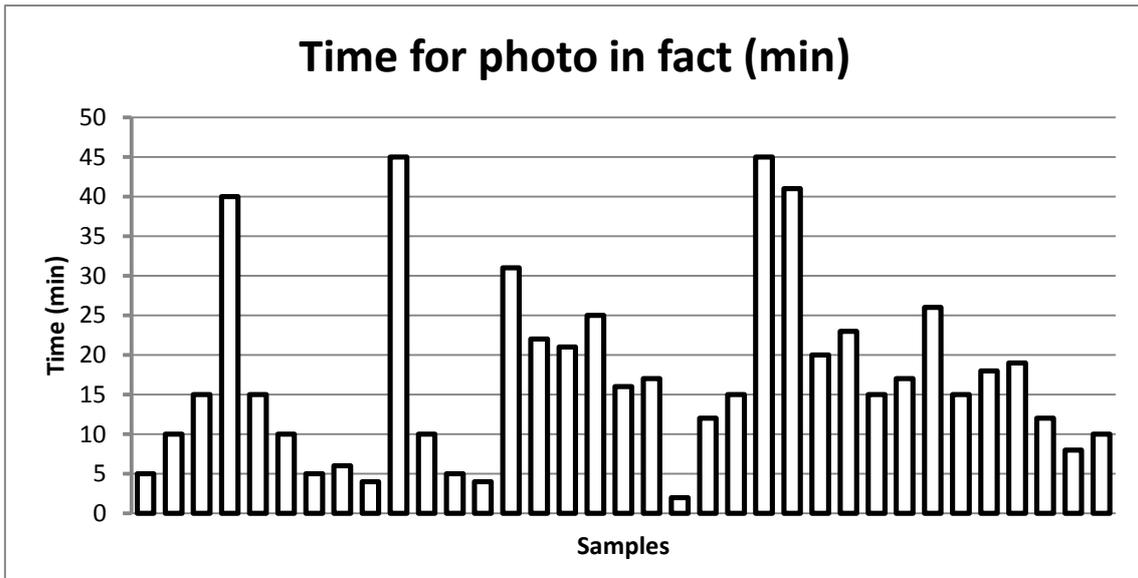


Figure 28: Histogram #2: Time for Photo in a fact

Figure 28 is the histogram that shows the amount of time the team spends when for finding a photo to a fact, this measurement is irregular in comparison to the first histogram (Figure 27).

As said before it has been also measured the quality of the existing same type application that are currently available. The information has been obtained by reading the comments the app has in the Google Play Store.

Table 11: Quality measurement of two existing apps

	<i>Fun Facts 8k</i>	<i>10,5k Cool Facts</i>
Facts quality	20	10
Interface	5	0
Design	1	3
Fake facts	8	17
Incomp.(device)	3	0
Usability	11	7
Lack info	1	3
Categories	1	0
N of facts	0	10

Table 11 shows how the complaints of two existing applications are classified. It has been taken into account 50 opinions of both of applications. The complaints about the quality of the application have been divided in 9 types:

Facts quality: it refers to the way some facts are not interesting, that don't bring knowledge to the user or are even inappropriate.

Interface: some users complained about the interaction between the user and the application.

Design: sometimes the apps have an unclear design or some features that can disturb the user.

Fake facts: most of “Fun Facts” applications have large libraries with more than 5.000 facts and their veracity is not always revised.

Incompatibility: if the application is not prepared to be run in every portable device it can cause complaint by the customers.

Usability: this refers to the fact that some applications have are not easy to use.

Lack of information: these applications are meant show very brief facts but users complain about the fact that there are facts with incomplete information.

Categories: most of these applications divide the facts between categories, and there are user’s opinion complaining about a lack of categories.

Number of facts: these applications are supposed to have large libraries of facts, indeed they have, but some of the facts are repeated.

This measurement is displayed with following Pareto charts, in **figures 29 and 30**.

Pareto analysis is used to organize data to show what major factors make up the subject being analysed. Frequently it is referred to as “the search for significance.” The basis for building a Pareto is the 80/20 rule. Typically, approximately 80% of the problem(s) result from approximately 20% of the causes (Montagna).

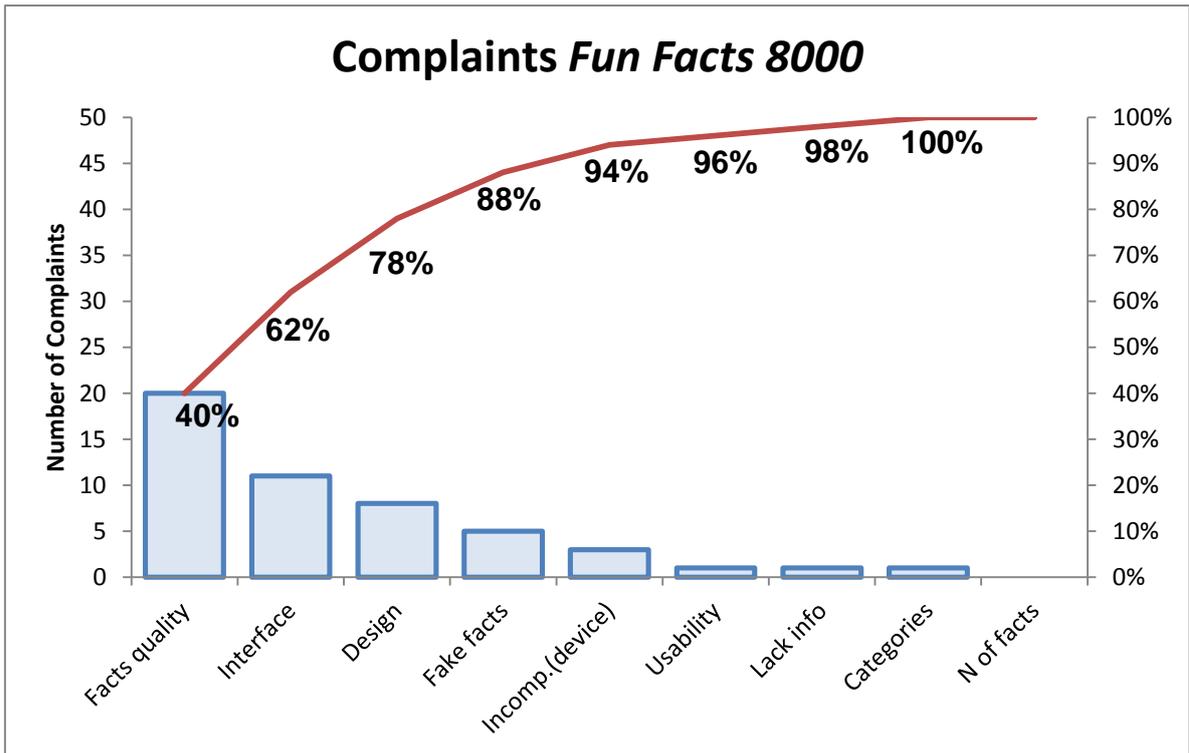


Figure 29: Pareto #1 Fun Facts 8000

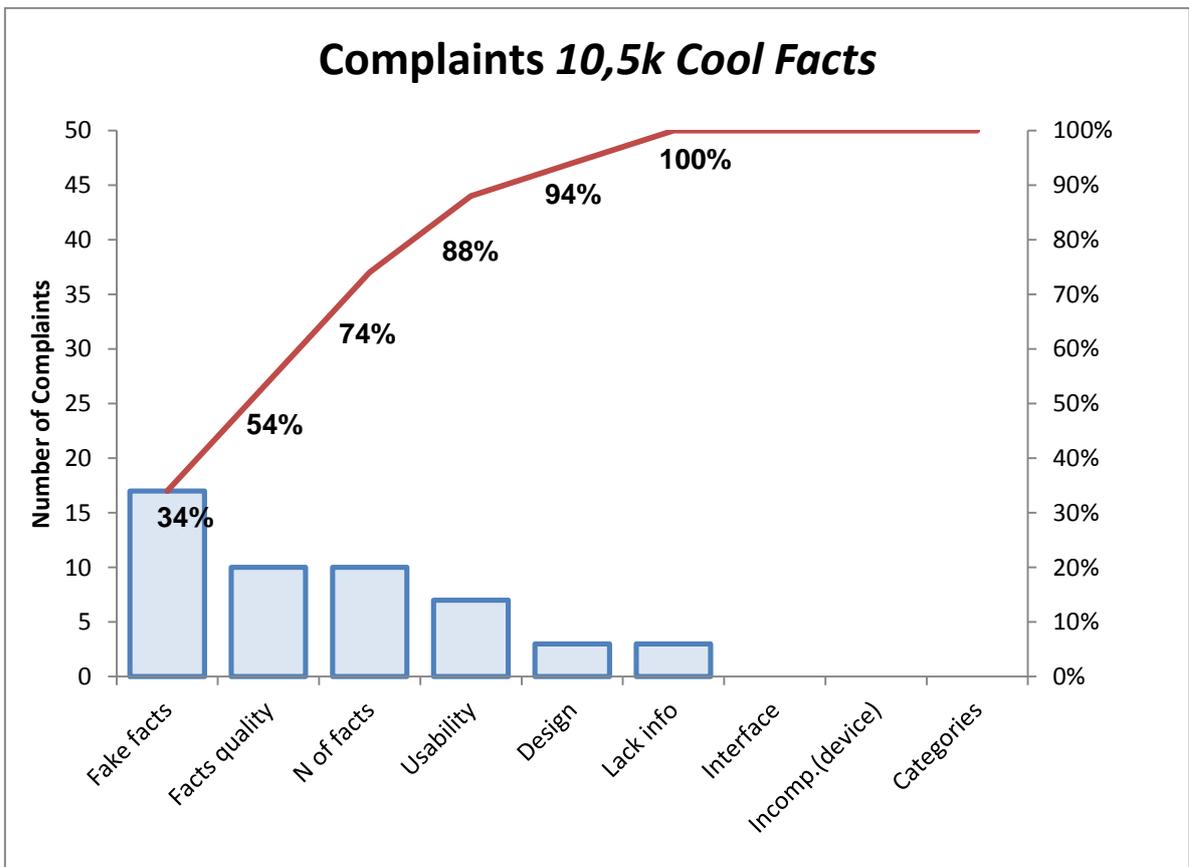


Figure 30: Pareto #2 10,5k Cool Facts

By using the Pareto chart it has been resulted that the main problems about the existing applications are the Fake facts, the facts quality, the number of facts and the Interface.

2.1.2.d Measure phase Outcomes

The measure part in DMAIC, implemented to this project has shown the most important problems the app could have and will be studied in the next part: Analyze.

2.1.3 Analyze

In the analyze step, the objective is to use the data from the measure step to begin to determine the cause-and-effect relationships in the process and to understand the different sources of variability. In other words, in the analyze step we want to determine the potential causes of the defects, quality problems, customer issues, cycle time and throughput problems, or waste and inefficiency that motivated the project. It is important to separate the sources of variability into common causes and assignable causes (Montgomery, 2009).

This part will be carried in 4 parts: the Ishikawa Diagram, the 5 whys, the cause-effect matrix and the DOE.

2.1.3.a Ishikawa Diagram

An Ishikawa diagram or cause-and-effect diagram (sometimes also termed fishbone diagram) is a useful tool to identify the influence parameters, i.e. the sources of uncertainty, of a whole test procedure or of a single working step (Meyer, 2007).

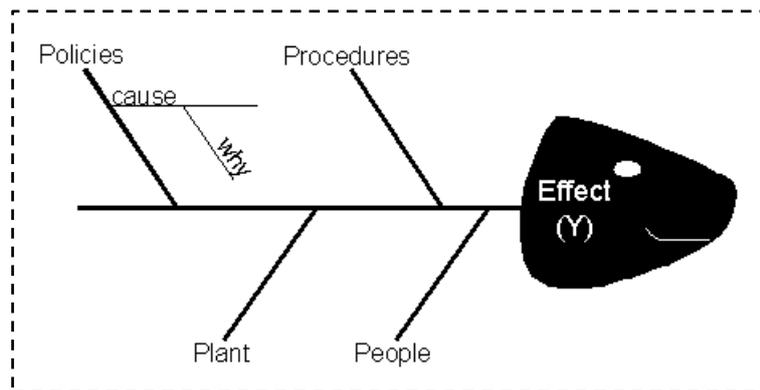


Figure 31: Fishbone Diagram Structure (isixsigma.com)

Figure 31 shows the structure that a Ishikawa or Fishbone Diagram has to have, were all of the causes form the fishbone and lead to the Effect.

In this project the effect will be the lack of interest that users have in the “Fun Facts” applications and the categories will be the most common types of complaints that the users express. The Ishikawa diagram Built for this project is shown in Figure 32. This diagram has been created with the help of a Template from (isixsigma.com).

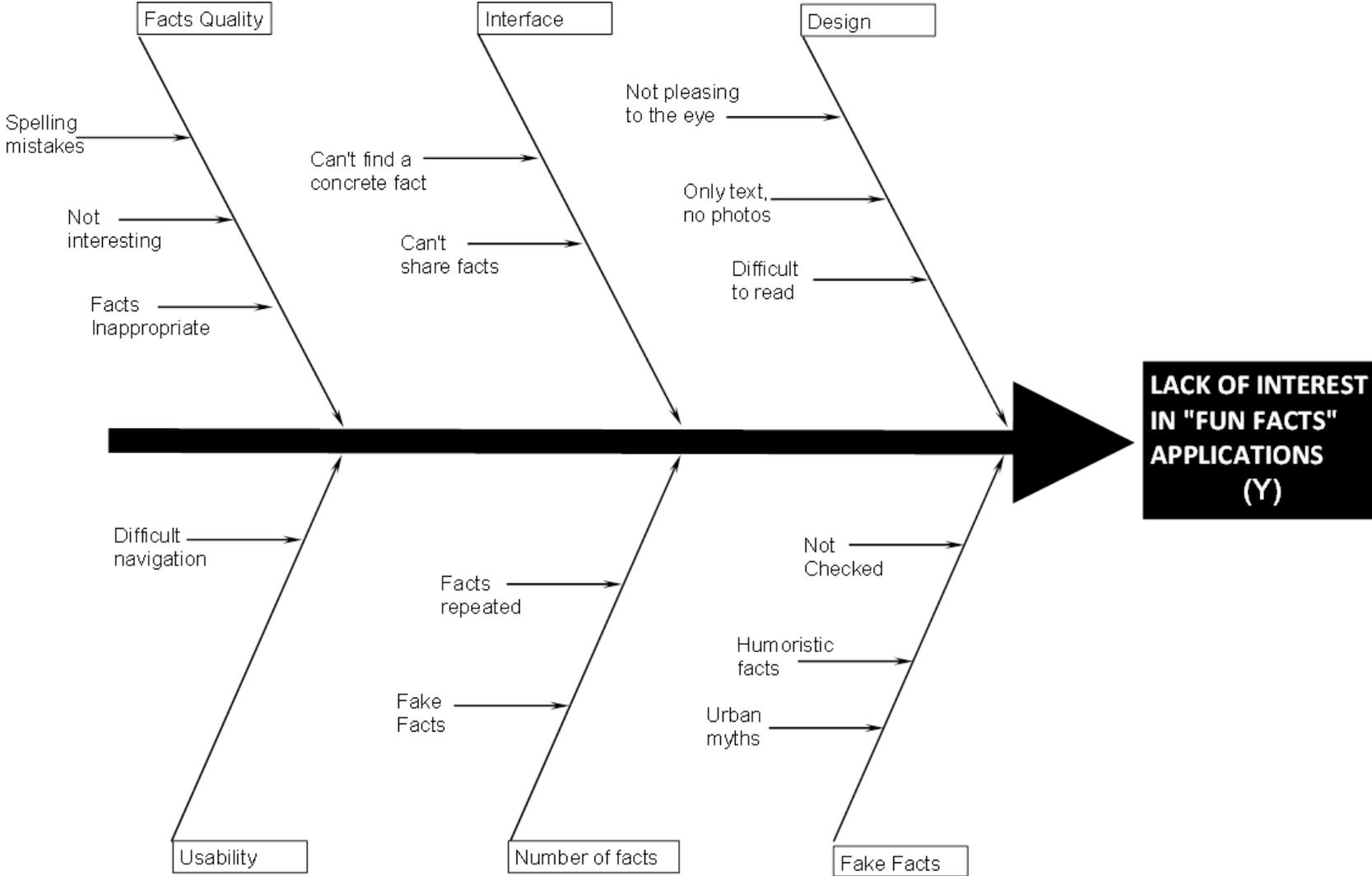


Figure 32: Built Ishikawa Diagram

2.1.3.b The 5 Whys

When looking to solve a problem, it helps to begin at the end result, reflect on what caused that, and question the answer five times. This elementary and often effective approach to problem solving promotes deep thinking through questioning, and can be adapted quickly and applied to most problems (Serrat, 2012).

To conduct the 5 Whys technique it has been used the earlier explained fishbone diagram and then it has been tried to find every root cause. The Diagram is shown in **Figure 33** in the following page.

The text in the diagram is not complete because of space reasons, so it is explained below:

- **Spelling mistakes:** some of the facts have either orthographic or grammatical mistakes.
- **Not Interesting:** for some users the facts are described as no interesting.
- **Inapp. Facts:** some apps have apps that are not appropriate for every user while the app is recommended for the use of people of all ages.
- **Finding Facts:** users have problems to find concrete facts that are in the app library.
- **Sharing Facts:** the user should be able to share the facts that are more interesting to him.
- **Not please eye:** some of the users complained about the fact that the app is not pleasing to the eye when reading the facts and can it is uncomfortable to read them for a long time.
- **Only Text:** the apps studied presented the facts without pictures.
- **Difficult to read:** some facts were written in fonts that are not easy to read.
- **Diff. Navigation:** the apps don't meet the point where the user can find the facts that they are looking for.
- **Repeated facts:** A lot of complaints were about the repetition of facts.
- **Fake facts:** it has been said by the users that some facts are not truthful.
- **No Checking:** it refers to the fact that the team have not checked the app enough before launching it.
- **Humour facts:** there were complaints about the facts not being serious, only jokes.
- **Urban myths:** some of the facts aren't true because are only things believed by many and often repeated.
- **Not VOC:** the apps studied lack of meeting what the customer really wants.
- **Not for all:** some facts showed are not suitable for children.
- **No search engine:** the user is unable to search for a specific fact.
- **No search button:** while some apps allow the user to easily share the facts they wants, some others don't have this feature.

- **Colours choice:** some facts can be difficult to read because of the text colour and the background colour.

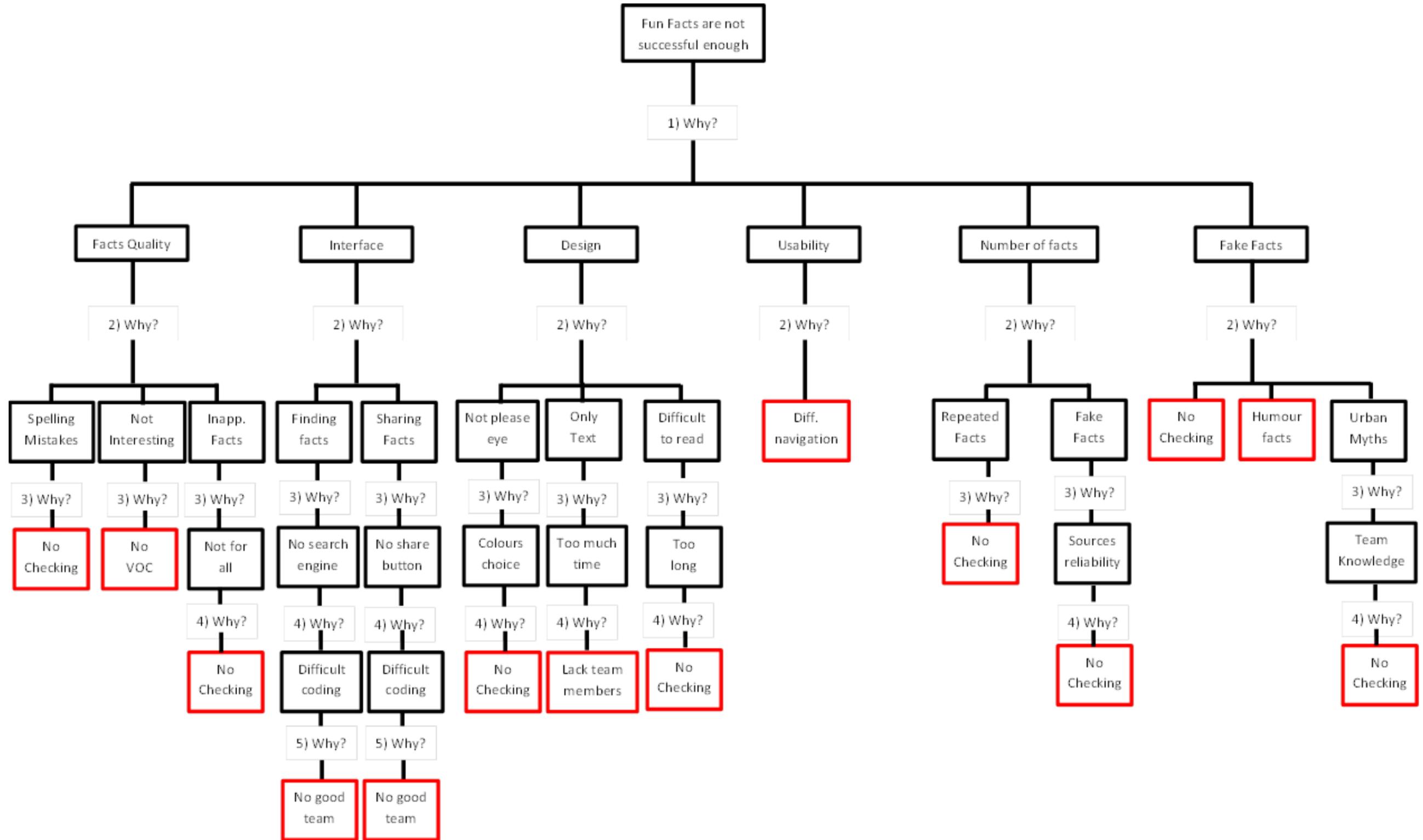


Figure 33: 5 Whys Diagram

- **Too much time:** one of the reasons of the mistakes made can be a lack of time when creating the app.
- **Too long:** Some of the facts have too much text, they are meant to be short and concise.
- **Sources reliability:** some of the sources where the team finds the facts are not checked about its reliability.
- **Team Knowledge:** some of the facts are just proposed by the team members.
- **Difficult coding:** features as the search engine or the share button are a bit more difficult to code than the usual facts app.
- **Lack of team members:** one of the reasons for the lack of time can be that the team is not big enough.
- **No good team:** if the team is not able to overcome the coding difficulties it can be caused because the members of the team are no well-chosen.

This diagram has shown that the root causes are the following:

- No Checking
- No VOC
- Lack of team members
- Difficult navigation
- Humour facts
- No good team

So the conclusion that this diagram leads to is that the main problem are the lack of checking the app before the launching and the poor choosing of the team members.

2.1.3.c Cause-effect Matrix

The objective of the Cause- effect Matrix is to identify what the most important root causes are by a correlation study between the Inputs and the Outputs.

In this project the Inputs and Outputs chosen are the customers complains and their causes, as shown in the **Table 12**:

Table 12: Cause-Effect Matrix

		1	2	3	4	5			
Scale:									
0	None								
2	Low								
4	Med-Low								
6	Med								
8	Med-High								
10	High								
		Output Variables (Ys, Effects)	Bad quality facts	Poor interface	Poor Design	Number of facts	Fake facts		
		Output Rating	10	8	7	6	9		
Input Variables (Xs, Causes)		Association Table					Rank		
1	Spelling mistakes	8	0	0	0	0		80	
2	Not Interesting facts	7	0	0	0	0		70	
3	Inappropriate facts	5	0	0	0	2		68	
4	No search engine	0	10	3	1	0		107	
5	No share button	0	7	3	0	0		77	
6	Not eye pleasing	0	2	10	0	0		86	
7	Only text, no photos	0	2	10	0	0		86	
8	Difficult to read	2	2	9	0	0		99	
9	Repeated facts	2	0	0	9	0		74	
10	No checking	9	2	7	7	10		287	
11	Humour facts	2	0	0	0	7		83	
12	Urban myths	3	0	0	0	10		120	

The clear result of this study is that the main cause is the **lack of checking**, as already seen in the 5 why's chart.

2.1.3.e DOE

DOE stands for Design Of Experiments, The manipulation of controllable factors (independent variables) at different levels to see their effect on some response (dependent variable) (Montagna).

A designed experiment is a test in which some purposeful changes are made to the input variables of a process or system so that we may observe and identify the reasons for changes in the output response. ... Experimental design methods play an important role in process development and process improvement (Montgomery D. C., 1991).

In this project the app has not yet been created so the Design Of Experiments cannot be carried out with the actual conditions, since it is not possible to obtain a response by changing an input if the app yet created or launched.

2.1.1.e Analyze phase Outcomes

In the page above, it has been clarified by the use of DMAIC's Analyze phase the root causes of the complaints the users have about the existing applications and after this it is simpler to come up with ideas to solve the problems.

2.1.4 Improve

As explained in the Introduction, the Improve part has 3 parts: Idea Generation, Idea Clarification and Idea Selection.

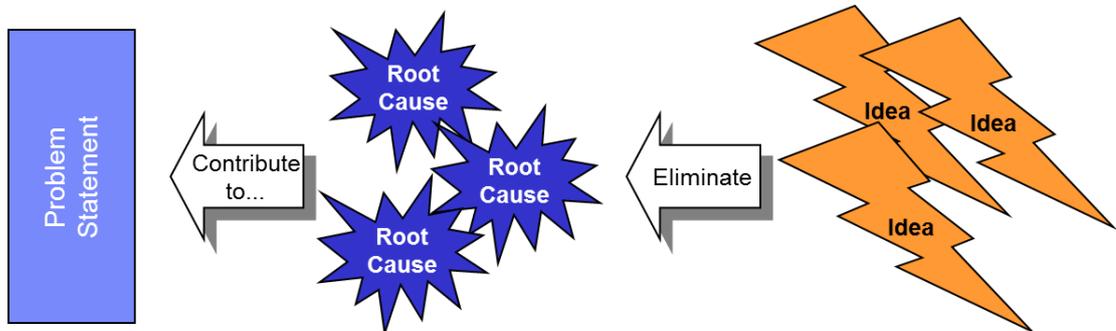


Figure 34: Improve step implementation (Montagna)

2.1.4.a Idea Generation

After having identified the Root causes in the Analyze step, the next thing the team has to carry is the Idea Generation, it means that with the help of the Brainstorming technique the team will come out with ideas to solve the root causes.

Brainstorming combines a relaxed, informal approach to problem solving with lateral thinking. It encourages people to come up with thoughts and ideas that can, at first, seem a bit crazy. Some of these ideas can be crafted into original, creative solutions to a problem, while others can spark even more ideas. This helps to get people unstuck by "jolting" them out of their normal ways of thinking (mindtools.com).

Since the Start-up to create the app is not yet created, the brainstorming team step will not be carried in this project.

However the ideas for this project are the following: A lot of checking of the application before the launching, implementing the "true/false game", implementing the "favourites" and "share" button and built the app with a clear design.

2.1.4.a Idea Clarification

In this phase the team will work to make the ideas that every member has proposed clearer select them afterwards.

2.1.4.a Idea Selection

To select between the Ideas generated in the Brainstorming it is often used a method called Multi-Voting:

Multi-Voting is a voting/brainstorming technique that prioritizes ideas. Its primary goal is to reduce the range of options, thereby preventing “information overload”.

Also known as N/3 voting, in Multi-Voting, N refers to the total number of ideas. Every team member is then given N/3 votes and instructed to vote on the most important ideas; the team member can only assign one vote per idea. E.g., if there are 30 ideas, each team member gets 30/3 or 10 votes each. Since there are less votes than there are ideas, the less important ideas will naturally be “weeded out”, thereby reducing the number of ideas the team must contend with (Swan).

By the end of this phase the team will have decided the ideas to implement in the project.

2.1.5 Control

This step can only be carried if the ideas selected were already implemented so in this project it will not be carried.

However in this part the team will do two main tasks: Standardize and Document the Solution and Control Chart.

2.1 Axiomatic Design Implementation

ADT will be applied in this project following the structure proposed by (Suh, 1999), first the definition of the four domains, then the Axioms with the help of the design matrixes.

2.2.1 Domains

The first step when applying the Axiomatic Design Theory is to define the domains of the design.

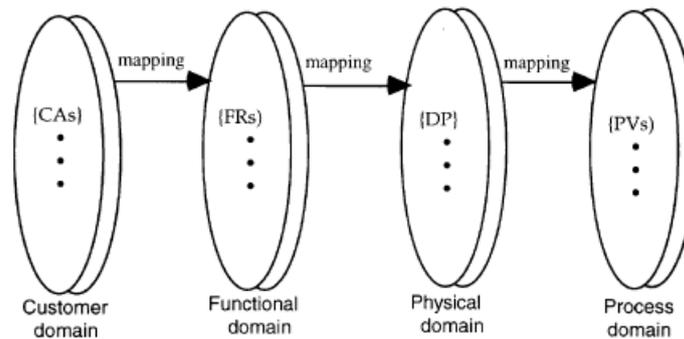


Figure 35: ADT Domains (Suh, 1999)

The world of design is made up of four domains: the customer domain, the functional domain, the physical domain and the process domain. The domain structure is illustrated schematically in **Figure 31**. The domain on the left relative to the domain in the right represents 'what we want to achieve' whereas the domain on the right represents the design solution for 'how we propose to satisfy the requirements specified in the left domain' (Suh, 1999).

In the following page, in **Figure 32** it has been applied the Domains theory to the creation of the application.

The CNs (Customer Needs), in the Customer Domain defined are:

- **CN1:** Big library of facts
- **CN2:** Entertaining features
- **CN3:** Easy usability
- **CN4:** Attractive appearance

The FRs (Functional Requirements), in the Functional Domain:

- **FR1:** More than 1000 facts
- **FR2:** Clear interface
- **FR3:** Easy usability
- **FR4:** Attractive appearance
- **FR5:** Appealing Design

The DPs (Design Parameters), in the Physical Domain:

- **DP1:** Use of Google free Templates
- **DP2:** Introduce True/False question
- **DP3:** Structure by categories and Saved facts
- **DP4:** Simple design: Photo + fact

And the PVs (Design Parameters), in the Process Domain:

- **PV1:** Build the application with Android Studio
- **PV2:** Assign to every fact a True/false question and more info
- **PV3:** Generate a library of fake facts
- **PV4:** Classify every fact with its category and create a favourite folder
- **PV5:** Find a photo for every fact

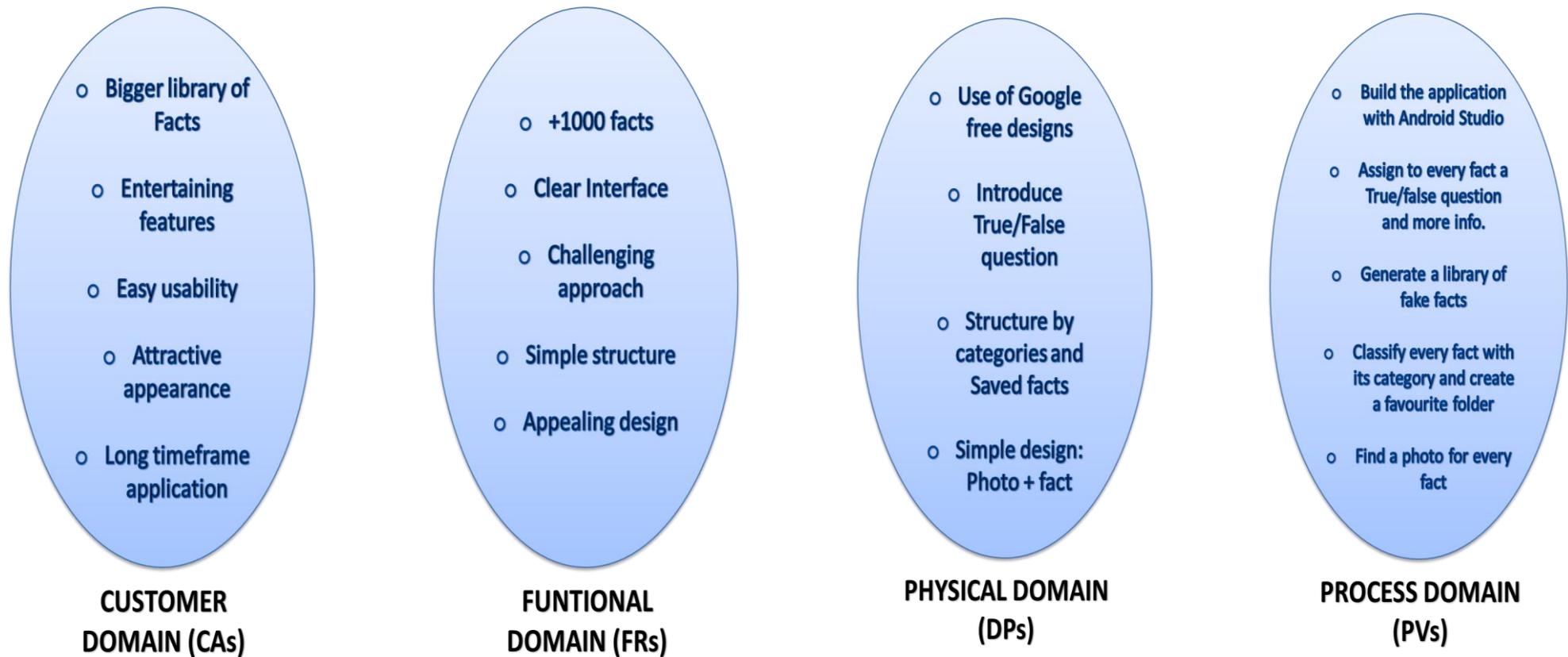


Figure 36: Domains Defined

2.2.2 Axioms Implementation

After having defined the 4 Domains of the Design the following step is to check if the design model follows the Axioms proposed in the Axiomatic Design theory, which are the following:

As defined by (Suh, 1999) an Axiom is a self-evident truth or fundamental truth for which there are no counter examples or exceptions. Axioms may not be derived from other laws of nature or principles. The two Axioms that rule this theory are:

Axiom 1: The Independence Axiom

Maintain the independence of the Functional Requirements (FRs). The Independence Axiom requires that the functions of the design be independent from each other, not the physical parts.

Axiom 2: The Information Axiom

Minimize the information content of the design. The Information Axiom states that the design option with the highest probability of success is the best design (Suh, 1999).

2.2.2.a Axiom 1 Implementation

For the Axiom 1 implementation and a better comprehension of it, the theory proposes the use of the Design Matrixes which are the links between the Domains.

$$\{FRs\} = [A] \{DPs\}$$

Equation 3: Link between FRs and DPs (Suh, 1999)

$$\{DPs\} = [B] \{PVs\}$$

Equation 4: Link between DPs and PVs (Suh, 1999)

Equations 3 and 4 show how the domains can be linked using the design matrixes. In **Equation 3**, $[A]$ is the Design Matrix that relates Functional Requirements with Design Parameters, and in **Equation 4**, $[B]$ is the Design Matrix that relates Design Parameters with Process Variables.

For assuring the first Axiom, the Independence Axiom, both of the design matrixes defined before have to be either diagonal or triangular, as **Equations 5 and 6** show:

$$[A] = \begin{bmatrix} X & 0 & 0 \\ 0 & X & 0 \\ 0 & 0 & X \end{bmatrix}$$

Equation 5: Diagonal Matrix

$$[A] = \begin{bmatrix} X & X & X \\ 0 & X & X \\ 0 & 0 & X \end{bmatrix}$$

Equation 6: Triangular Matrix

When the design matrix $[A]$ is diagonal, each of the FRs can be satisfied independently by means of one DP. Such a design is called an uncoupled design. When the matrix is triangular, the independence of FRs can be guaranteed if and only if the DPs are changed in a proper sequence (Suh, 1999).

Definition of matrix $[A]$, the design matrix that relates Functional Requirements with Design Parameters:

- **FR1:** More than 1000 facts
- **FR2:** Clear interface
- **FR3:** Easy usability
- **FR4:** Attractive appearance
- **FR5:** Appealing Design

- **DP1:** Use of Google free Templates
- **DP2:** Introduce True/False question
- **DP3:** Structure by categories and Saved facts
- **DP4:** Simple design: Photo + fact

Table 13: Proposed FRs and DPs Relation

	DP1	DP2	DP3	DP4
FR1				
FR2			X	
FR3	X	X	X	
FR4	X			X
FR5	X			X

Table 12 shows which of the Functional Requirements are related to Design Parameters, to build the Design Matrix. The Design Matrix resulting from this would be:

$$[A] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & X & 0 \\ X & X & X & 0 \\ X & 0 & 0 & X \\ X & 0 & 0 & X \end{bmatrix}$$

Equation 7: Design Matrix from the FRs and DPs proposed

Equation 7 concludes that how the domains have been defined the system would not meet the requirements of the First Axiom, yet the resulting Design matrix is neither diagonal nor triangular.

Then the next step for the correct implementation of the Axiomatic Design Theory should be the reformulation of the four domains of the design, finding the correct FRs, DPs and PVs to meet the requirements of the Axioms.

The new FRs, DPs and PVs have been defined in the **Table 13** and in **Figure 33**, in the next page of this document.

Table 14: New FRs, DPs and PVs

	Functional Requirements (FRs)	Design Parameters (DPs)	Process Variables (PVs)
1	+1000 Facts	Facts Resources	Checking Reliability
2	True/false game	Question in every fact	Coding the Game
3	Clear Interface	Categories	Classifying every fact
4	Appealing Design	Android Studio Templates	Choosing the Templates

The FRs (Functional Requirements), in the Functional Domain:

- **FR1:** More than 1000 facts in the App library
- **FR2:** True/False Game
- **FR3:** Clear Interface
- **FR4:** Appealing Design

The DPs (Design Parameters), in the Physical Domain:

- **DP1:** Facts Resources, where to find the facts
- **DP2:** True/False question in every fact
- **DP3:** Categories
- **DP4:** Android Studio Templates

And the PVs (Process Variables), in the Process Domain:

- **PV1:** Checking Reliability of the Resource
- **PV2:** Coding game
- **PV3:** Classifying every fact
- **PV4:** Choosing the Android Studio Templates

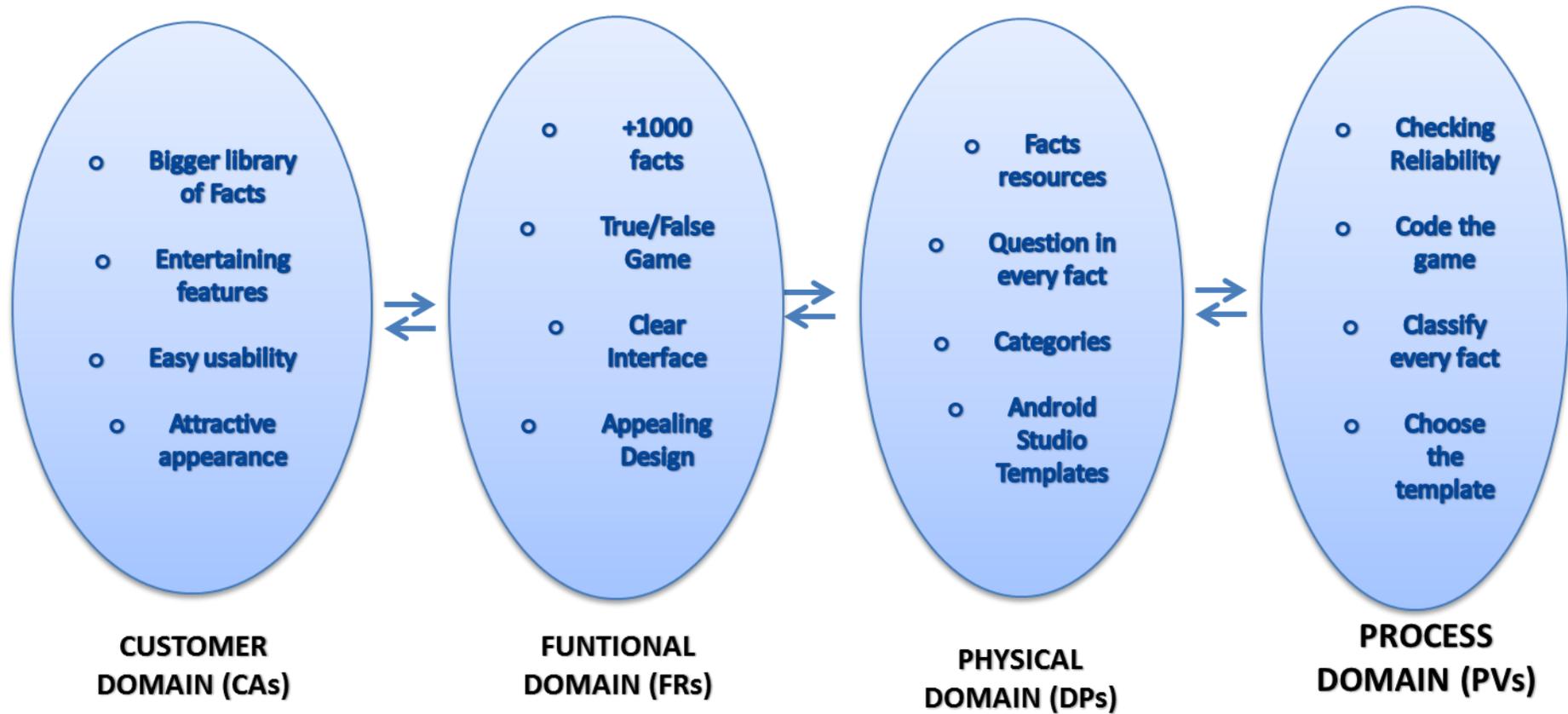


Figure 37: ADT new Domains

Next step is Check if the new formulation of the Domains of the Design meets the Axiom 1, the Independence Axiom. Therefore the design matrixes have to be built.

Definition of matrix $[A]$, the design matrix that relates Functional Requirements with Design Parameters:

- **FR1:** More than 1000 facts in the App library
- **FR2:** True/False Game
- **FR3:** Clear Interface
- **FR4:** Appealing Design

- **DP1:** Facts Resources, where to find the facts
- **DP2:** True/False question in every fact
- **DP3:** Categories
- **DP4:** Android Studio Templates

	DP1	DP2	DP2	DP3
FR1	X			
FR2		X		
FR3			X	
FR4				X

Figure 38: Proposed new FRs and DPs Relation

$$[A] = \begin{bmatrix} X & 0 & 0 & 0 \\ 0 & X & 0 & 0 \\ 0 & 0 & X & 0 \\ 0 & 0 & 0 & X \end{bmatrix}$$

Equation 8: New Design Matrix $[A]$

As **Figure 34** and **Equation 8** show, at least the first part of the model now meets the requirements from the Independence Axiom of Axiomatic Design Theory. The reason because the relation between FRs and DPs fulfil the requirements of the First Axiom is because the Design Matrix resulting from its relation is a diagonal Matrix.

To ensure that the Independence Axiom is respected in this design, next step is to check if the relation between Design Parameters and Process Variables also generate a Diagonal or Triangular Matrix.

Definition of matrix $[A]$, the design matrix that relates Functional Requirements with Design Parameters:

- **DP1:** Facts Resources, where to find the facts
- **DP2:** True/False question in every fact
- **DP3:** Categories
- **DP4:** Android Studio Templates

- **PV1:** Checking Reliability of the Resource
- **PV2:** Coding game
- **PV3:** Classifying every fact
- **PV4:** Choosing the Android Studio Templates

	PV1	PV2	PV2	PV3
DP1	X			
DP2		X		
DP3			X	
DP4				X

Figure 39: Proposed new DPs and PVs Relation

$$[B] = \begin{bmatrix} X & 0 & 0 & 0 \\ 0 & X & 0 & 0 \\ 0 & 0 & X & 0 \\ 0 & 0 & 0 & X \end{bmatrix}$$

Equation 9: Design Matrix $[B]$

As **Figure 35** and **Equation 9** show, the new proposition of the four domains now meets the requirements of the First Axiom of the ADT, the Independence Axiom, maintains the independence of the Functional Requirements.

2.2.2.b Axiom 2 Implementation

There can be many designs which are equally acceptable from the functional point of view. However, one of these designs may be superior to others in terms of probability of success in achieving the design goals as expressed by the functional requirements. The Information Axiom states that the one with the highest probability of success is the best design (Suh, 1999).

Rigorous applications of the axioms to select the best design options, and development of complete design solutions, can both be facilitated by developing quantitative relations between the elements in the domains. Nevertheless axiomatic design can be used advantageously qualitatively and the attempt to develop the quantitative relations can be valuable (Brown, 2011).

In other words, as it has been understood, this Axiom proposes that the Design that needs less measurement or indexes (Information) will be the best. In the case of this project the team will evaluate qualitatively every Functional Requirement, Design Parameter and Process Variable to see what would be the most important requirements to solve.

In the following paragraph they are written all the Functional Requirements (FRs), Design Parameters (DPs) and Process Variables (PVs) studied in this project:

- **FR1:** More than 1000 facts in the App library
- **FR2:** True/False Game
- **FR3:** Clear Interface
- **FR4:** Appealing Design
- **DP1:** Facts Resources, where to find the facts
- **DP2:** True/False question in every fact
- **DP3:** Categories
- **DP4:** Android Studio Templates
- **PV1:** Checking Reliability of the Resource
- **PV2:** Coding game
- **PV3:** Classifying every fact
- **PV4:** Choosing the Android Studio Templates

According to the second Axiom of the Axiomatic Design Theory the most important Functional Requirement to meet is the one that states that the app must have at least 1000 facts, since the information required to measure if the team meet the goal is only the counting of them.

2.2.2 Axiomatic Design outcomes

The implementation of Axiomatic Design Theories it has been very helpful to try to meet the Customer's Need, and by applying the Axioms, it can be very helpful to organise the work much better and not carry out unnecessary parts.

3. COMPARISON AND CONCLUSIONS

3.1 DMAIC implementation conclusions

What the Implementation of DMAIC in this project has shown is that it is a methodology clearly meant to be applied on existing products or processes, some of its parts cannot be applied without the product itself already created.

The conclusion is that the first three parts: Define, Measure and Analyze are very helpful when creating an application. These steps can help the team to structure the work and optimise the efforts. In this project it could be implemented another similar theory: DFSS (Design For Six Sigma), that focuses on the creation of a product.

3.2 Axiomatic design conclusions

Axiomatic Design Theory is a method that can help the team to do a very clear work, focusing only in the main problems.

By using these methods the team can solve the problems by only focusing in the main parts of them, not losing time in the least important things.

3.3 Comparison between the theories

Both of these theories are a really good way to manage a project, but are really different.

The main difference detected is that DMAIC focuses on every detail and every step of the project, while ADT focuses on the important aspects only and it is meant to be used only in the first's parts of the project.

The main similarity would be that both of the theories emphasise on the Voice Of the Customer, it is very important to meet the exact need that the customer has.

3.4 General conclusion

The end of this project it is the general conclusion that is that to take 100% advantage of both theories the work can be done in other ways: DMAIC should be implemented after creating the app; and ADT should be implemented more in a quantitative way, not qualitative.

However these theories have cleared a lot the idea of how a project can be managed and clearly they can help to make a project successful.

4. REFERENCES

- Brown, C. (2011). Axiomatic design applied to a practical example of the integrity of shaft surfaces for rotating lip seals. *Procedia Engineering*, 19, 53-59.
- casual.pm*. (n.d.). Retrieved from <https://project-charter-template.casual.pm/>
- Freeimages.com*. (n.d.). Retrieved from www.freeimages.com
- GE. (2013). *What is six sigma: the roadmap to customer impact*.
- isixsigma.com*. (n.d.). Retrieved from <https://www.isixsigma.com/implementation/financial-analysis/six-sigma-costs-and-savings/>
- Meyer, V. R. (2007). Measure Uncertainty. *Journal of Chromatography A*, pp. 15-24.
- Montagna, P. M. (n.d.). DMAIC Approach. Politecnico di Milano.
- Montgomery, D. C. (1991). *Design and Analysis of Experiments*. New York: John Wiley and Sons.
- Montgomery, D. C. (2009). *Introduction to statistical quality control*. John Wiley & Sons Inc.
- Pan, Z., Park, H., Baik, J., & Choi, H. (2007). A Six Sigma Framework for Software Process Improvements and its Improvements. *14th Asia-Pacific Software Engineering Conference (APSEC'07)*, (pp. 446-453). Republic of Korea.
- Pexels.com*. (n.d.). Retrieved from www.pexels.com
- QFDonline.com*. (2011, December 11). Retrieved from <http://www.QFDOnline.com>
- Ries, E. (2011). *The Lean Startup*. New York: Crown Business.
- Serrat, O. (2012). The five whys technique. *Asian Development Bank*, 16-18.
- Suh, N. P. (1999). Axiomatic Design Theory For Systems. *Research in Engineering Design*, 189-209.

