Computational Representative Techniques of Software Engineering for Social Sciences

Basit Habib
Bahauddin Zakariya University, Multan

Javaid Akhter, PhD
Bahauddin Zakariya University, Multan

Abstract:
A project when developed can have the following problems of the statement not being clarified or its context being not clear. It depends on the user how he understands the statement of the problem and decides to solve the problem. But first of all it is essential for him to have a clear vision of the problem from its development to its working. In this paper it will be seen how to create a idea of the statement of the problem and how to proceed it by using some essential tools of software engineering for the practical approach of social scientists.

Keywords: Understand ability, Judgment, Problem Description and Problem Presentation

I. Introduction

Today’s problem defines us the needs of a social scientist in all aspects. The first problem is that how can he understand a problem. So in other words his main problem is a “Problem”. If any project is given to a social scientist how can he present his data or the main theme of what is desired by him. So in this paper there are some easy approaches which he can adopt to represent his work in terms of a report, a manuscript or even in his thesis.

II. Background

There are some very obvious techniques for the solving of different issues so that a naïve user can easily understand them and can make a perception on what he is looking at. The main thing which is important to understand is that has the problem been understood by us or not?. If the problem statement is understood then can come the part of its feasibility and its development. The development of the problem is not the issue at this stage of work the main thing is can the work be understood and how can t be easily represented.

III. History and methodology

As the concern of the idea is concerned, people who are working in the field of social sciences have more of a qualitative work which they do. But due to the daily enhancement of scientific tools they basically quell with them and at the end they define that they have solved the problem but without knowing its basic methodology and its background working (Manually).
The methodology which is going to be used in this paper is the description of The “System Development Life Cycle” (SDLC). And along with this the three main diagrammatic ways to represent the flow of the problem, that how it can be solved which are:

i. Use Case Diagram
ii. ERD (Entity Relation Diagram)
iii. DFD (Data Flow Diagram)

By using these Easy, Understandable and adoptable techniques A problem will no longer be a problem for the social scientists and the understanding and representation of a problem will be easy.

IV. System Development Life Cycle (SDLC)
The System development life cycle (SDLC) is a framework defining different tasks performed at steps in the system development process. SDLC is a structure followed by an analyst or team of analyst for the problem statement. It consists of a detailed plan development, its output and working. The life cycle defines steps for improving the quality of system and the overall development process. They also have the ability show reclusiveness if there is any problem during the research. This term is also known as the system development process.

SDLC consists of following activities:

**Preliminary investigation:**
The most important parts of system development, requirement gathering or requirement analysis are usually done by seeing the statement of the problem. After the requirements are gathered for research, a document is created in which the scope of the project is determined and defined.

**Requirement specification:**
The user in this phase of SDLC defines the tools which will be required by him for the research work which he will do. These tools can depend of Qualitative and Quantitative, both the approaches

**Development / Implementation:**
For a social scientist, development phases will act on different scenarios as compared for scientists. In this phases he will decide how to manipulate his data, define a hierarchy (where to begin with) of data, how to then interpret his data and at the end how to generate desired output from his data (keeping on the track).

V. Testing
This is the process of finding defects in the work done. There are three main types of testing which are suitable for all researchers defined as under:

**White-box testing**
It is a method that tests internal structures or workings of a component or modules produced by the researcher from his data. This testing is done by the user himself so that he can identify what are the major problems in his work before releasing it into the market.
White-box test design techniques include:
   Control flow testing: How the data was described and arranged in a flow
Data flow testing: Is the integration and synchronization being developed between the defined themes.

   Branch testing: Do themes or attributes show a relationship between each other
Path testing: Has he been doing work on the right path w.r.t the desired output of the work.

**Black-box testing**
It is a method that tests the functionality of a component in the viewer’s eye. In this type if any objection is made on the done work by any reader then it will be known as black box testing because the researcher does not know that by which approach his work has been questioned mark.

Typical black-box test design techniques include:
   Decision table testing: Table is created in which different reasoning are given for the failure of the work.

   All-pairs testing: Things do or do not pair with one another and is sense being developed?
Equivalence partitioning: Hierarchy of themes or attributes is seen that are they equally justified.

   Boundary value analysis: Has the work been done in its meaningful wall or things are irrelevant.

**Grey-box testing**
It is blend of two opponent test methods namely white-box and black-box testing. The aim of this testing method is to look for the possible defects due to improper structure or improper usage of applications. It is also termed as translucent testing

**VI. Use Case Diagram**
A use case diagram is used for the representation of An Entity. An entity can be any person, Organization or depending on the statement of the problem. A use case diagram explains all the characters which play a role in the problem statement and the work which is going to be performed by them. In other or scientifically words we can describe the use case diagram as Attributes and its sub-Attributes.

   The purpose of use case diagram is to show the versatile aspect of a system. But there are many other diagrams which show the same purpose, so we will let’s see some specific purpose which will distinguishes it from the other diagrams.

   Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified.
So it concerns with the following:
   i. Gathering requirements of a system.
ii. Achieve an outside view of a system.
iii. Identify external and internal factors influencing the system.
iv. Show the interacting among the requirements are actors.

**How to draw Use Case Diagram?**
Use case diagrams are considered for requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So we can say that uses cases are nothing but the system functionalities the actors can be human user, some internal applications or may be some external applications.

**Relationships among the use cases and actors.**
The following guidelines are used to draw an efficient use case diagram. The name of a use case is very important. So the name should be strong enough to show its functionalities
Give a suitable name for actors.
Show relationships and dependencies clearly in the diagram.

**VII. ERD (Entity Relation Diagram)**
An ERD is such kind of an explanatory representative diagram which shows the integrity of the attributes of a statement. In this the main entities are co related with the sub entities or attributes of the entities and a relationship is developed so that bounding of elaborating and solving the problem can be made.

It is a graphical representation of entities and their relationships to each other. An entity is a basic of data which can be an object or its storage. ERD’s can be explained in many different aspects, they can be simple or they can be complexed. More the ERD is complex more it will be understandable.

**One-to-One Relationship:**
If an entity (A) is associated with another entity (B) directly. For example, in a database of Account holders, each Account holder (A) is associated with a specific Bank(B).

**One-to-Many**
If an entity (A) is associated more than one instances of another entity (B), but entity (B) exists more than one time and every entity (B) is directly connected to Entity (A). For example, One person (A) having affiliation with more than one Bank (B).
Many-to-Many
If an entity (A) is associated with one, zero or many instances of another entity (B), and one instance of entity (B) is associated with one, zero or many instances of entity (A). For example, Many People having accounts in different Banks.

Represent that information with symbols. Generally E-R Diagrams require the use of the following symbols:

- **Entity**
- **Relationship**
- **Attribute**

ERD’s are basically the bones which develop a skeleton like structure of the understanding of a problem statement. In ERD’s it is important for the user to define such kind of sub attributes which show co-relation of the entities and a better integrity rate can be achieved, because more the integrity more the strongest will increase of the skeleton. By carrying on the following above example let’s see the part of ERD of a person and the remaining will be discussed below.

**VIII. DFD (Data Flow Diagram)**
A DFD shows the working and hierarchy of the entities and their procedural in such a manner that a step wise process is generated and it helps easily to understand the working of statements. The *Data Flow Diagram* (DFD) is a graphical representation of the flow of data systematically. It enables to represent the processes in an informative manner. The DFD visualizes how the system operates, what its accomplishment’s are and how it will be implemented.

Data flow diagrams are used by analysts to design information-processing systems but also as a model of whole the system. DFD at the very beginning are modelling processes which model the functions of any problem which carry out and interact between functions together with focusing on data exchanges between processes. This is a conceptual, logical, and physical data models and object-oriented models.
In this diagram the entities and their attributes again show the working but this time in an elaborative manner with directions. A DFD is such a diagram in other words, which shows the functionality of the organisms of the skeleton defined earlier as the structure of the problem statement. Whereas the organisms are the attributes which run the skeleton to perform particular functions the below mentioned DFD know shows all the themes or Entities of the problem statement, and is showing a hierarchy relationship between all the themes abstracted from using the above representative tools of software engineering.

IX. Explanatory Diagram:
This diagram shows the basic connectivity of the attributes with the sub attributes in a direct or a indirect manner. This diagram connects in such a way that the integration level of the attributes can be seen by which we can access that how much the work done is authentic and different attributes or themes are showing dependency on each other.

X. Problem Statement:
By the coordination of USAID and Aurat foundation a project was given to Gender studies Department, B.Z. University, Multan. The problem statement is to gather data on Gender Base Violence and by doing its analysis, to abstract themes and see what are the main issues of violence in our society of South Punjab.

System Development Life Cycle (SDLC):

Preliminary Investigation:
By applying this phase on our above mentioned problem it is for us to see that How much data is to be collected for the analysis on GBV? Which kind of area is to be observed which includes house wives, working women, Students and other kind of Gender at different places which cover the requirement? What will be the results which can be achieved by this data? Is this Data collecting technique the only technique (Questioner) which can be applied on this statement or can we use another approach?

User Requirement:
In this phase it will be seen that what are the tools on the user end which will be required for the gathering of data if we go with the data collecting technique as mentioned above:

i. Preparation of data gathering through a Questioner or set of Questioners for the different categories.
ii. The commonality in the questions should be kept for the further step of analysis.
iii. Public Meetings and interactions
iv. To have the record of gathered information for statistical analysis of different attributes asked in the questioner.

Development:
In this phase if we go with the approach of preparing a questioner, then the user will start to first of all develop a questioner and see what are the important aspects which can be kept in common in the questioner?
In the above mentioned case study a single questioner was developed on the aspects of Gender Base Violence. The precedence level and importance of the topic is to be kept in track. Along with this some public meetings were arranged and recorded voices of different people existing in the above mentioned categories were done. After that different options of the questioner were put into the computer so that comparisons on different themes of the questioners could be seen.

Testing:
As the phase of testing is concerned, in this scenario white box testing was done because it was up to the data developer to decide whether the data given is correct or not and by seeing the statistical analysis of the data the results proved that the answers given at different aspects were how much integral to each other and did they show the desired results as regarding the user perceptive or not.

As far as data gathering is concerned and making any perceptions through it, it can be said that mostly the “White Box” testing technique is more effectively used.

This is how some phases of SDLC can be applied on a problem statement of a Social Scientist. Know lets observe the step wise process which can indicate us how the process could have occurred

Use Case Diagram:

Fig.1. Use Case Diagram
Entity Relation Diagram (Simple):

Fig. 2.1. Entity Relation Diagram (Simple)

Entity Relation Diagram (Complex):

Fig. 2.2. Entity Relation Diagram (Complex)
XI. Conclusion

The approach of using computational techniques in a problem statement helps out how to categorically solve a problem by parts. These techniques also produce ease for the understanding of the process and defining the hierarchy of the process. The main issue in social science is representation and a specified module to solve problems qualitatively and quantitatively. If the above techniques of “Computational techniques for
social scientists” could be adopted then a streamed line process can be defined, as mentioned in the Problem statement discussed.

References:


Software Methodologies Advantages & disadvantages of various SDLC models.mht www.smartdraw.com/resources/tutorial/entity Relation-diagrams