

# Adaptive Q-Sort Matrix Generation: A Simplified Approach

Bruno Oliveira

bruno.f.silva.de.oliveira@tecnico.ulisboa.pt

Francisco Maria Calisto

francisco.calisto@tecnico.ulisboa.pt

Lilian Gomes

lilian.gomes@tecnico.ulisboa.pt

José Borbinha

jlb@tecnico.ulisboa.pt

**Abstract**—This paper aims to use the previous work related to the DELPHI method, and, in particular, the Q-Sort method for information retrieval of a panel of experts, to provide a new and simple algorithm to generate Q-Sort matrices that adjust to the size of a given survey to have more questions whose weight is null for the outcome of the round, giving experts the need to prioritise some questions above others in order to reach a consensus in a more direct way.

**Index Terms**—Qualitative Methods; Quantitative Methods; User Studies; Q Methodology; Human Computer Interaction; Online Questionnaires; User Interface; Q-Sort; Delphi Methodology.

## I. INTRODUCTION

This document reports the findings of an INESC-ID summer internship of Bruno Oliveira and Francisco Maria Calisto, which work was motivated by the MSc Research Project of Lilian Gomes and advised by Professor José Borbinha.

Most of the surveys analysed with the help of the DELPHI method [8] make use of a particular subset of tools (the Q-sort method) popularised by the aforementioned method, that rely on a matrix-like structure known as the Q-sort matrix, that allows a panel of experts to reach a consensus about a specific topic by ranking the available options in an importance scale. In Q-Sort, the idea is that this ranking scheme works in rounds, and by making the results of the  $N - 1$ th round known, the  $N$ th round will be affected, having a convergence effect until a consensus is reached. This method is quite effective for a moderate number of questions, as previously discussed in [1], with a consensus being reached after 3 to 4 rounds.

It is believed that, however, this method has some limitations when the number of available options grows too large, both in terms of feasibility, since the time taken to answer a full survey increases, as well as in terms of the size of the available experts pool, as they might lose interest in subsequent rounds making the statistical analysis of obtained results difficult and of low relevance. Our algorithm proposes a possible solution to address these issues, by giving the experts a specific Q-Sort matrix structure based on the number of questions on the survey, which lets the experts focus on the

more important convergence points by giving a null weight to a larger number of questions. It is believed that such approach allows for a more adaptive survey generation method and, as a consequence, experts will tend to stay active and engaged in answering the survey for more rounds. That will make the ordering of the obtained data more reliable and easy to handle, and, more importantly, it will make the end results more reliable and accurate. This paper briefly introduces the reader to the DELPHI Methodology [8], using the subset that is the Q-Sort method [9] as a motivation. The authors reference the current research on DELPHI Methodology to identify the variables used to understand Information Technology, Information Seeking and General Information through the usage of the Q-Sort method and explain what can be improved in existing solutions. We present an approach of the DELPHI Method [8] that maximises its potential on the User Interface, specifically when the Q-Sort subset of methodologies is used as an analysis tool. Our main goal is to develop a new user interface that builds upon existing solutions to increase both the completion rate of surveys "deployed" under our new interface and, as a direct consequence of that, also increase the reliability of our results [3]. In our approach, called Adaptive Q-Sort Matrix Generation, study owners configure surveys whose options are presented to experts as different answers to the question that is being the subject of the survey. Then, experts will sort their answers according to personal significance.

## II. ONLINE DELPHI METHOD PROTOTYPE TOOL

This section will briefly introduce the Online DELPHI Method Prototype Tool [10], designed to support a study based on the DELPHI methodology using Q-Sort as an approach. Its main function is to support the gathering of web-based data, which reduces time between rounds of the study. Supporting the Q-sort method in our tool was of great importance, as the usage of more traditional systems (e.g. registering all answers by post mail) in very large panels [2], (as is the case-study on this document), would normally fail, but instead, when using this tool, those limitations are overcome, and at the same time, it is assured that all panel members follow the standard the Q-Sort procedures [3], essential for proper application of

the methodology. The high response rate confirms that the choices made regarding the support tool have been appropriate and contributed to the success and accuracy of relevant data gathering.

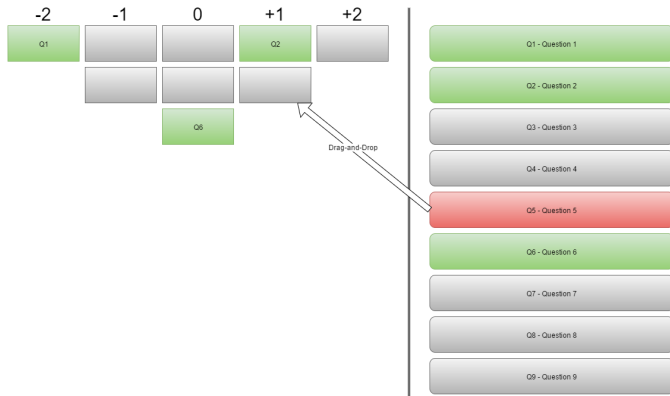


Fig. 1: Q-Sort Drag-and-Drop Questions into Matrix example

After authentication, the expert is presented with a pre-configured (by the study owner(s)) list of questions to be drag-and-dropped to the Q-Sort Matrix (Figure 1). This action can be undone and redone, so that the expert can correct unexpected mistakes, or even to choose other matrix position, may he change its previous opinion. Using colors to give the experts a clear perception of the heterogeneous distribution of values in the matrix can be useful, as it improves the productivity and efficiency of the answers in a survey. It also provides a not only simple, but reliable technique for experts to understand where a question should or should not be placed. In the example (Figure 2), the list had 9 questions, so, the first step was to split the questions into four different groups, distinguishable by different colours.

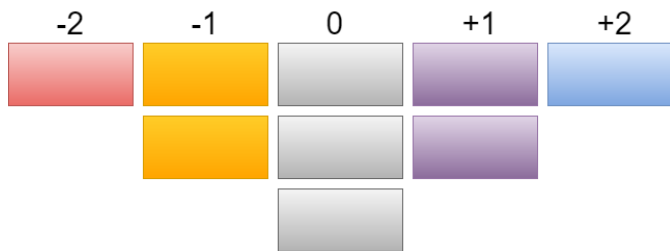


Fig. 2: Colors in the Q-Sort matrix depicting the weight of questions

For this number of questions, each panel member must choose a group made up of a single most important question (for this example), with a positive weight (in what concerns the final statistical analysis) of 2 marked in blue, a group made up of 2 violet questions with a positive score of 1, a group made up of 3 neutral questions with a null score, ambivalent or with a regular importance, greyed out. The other questions follow the same score distribution, but, carrying negative weight towards the final statistical analysis. This division in three groups results from an almost normal distribution, pre set according to Figure 3. The distribution may vary from study

to study according to the number of questions to be used. On the matrix, we already have the questions dragged from the right side of the tables as we can see on Figure 1. For this, the experts analyse the list of questions that would like to rearrange between questions already dragged to the matrix. This behaviour is represented in Figure 3.

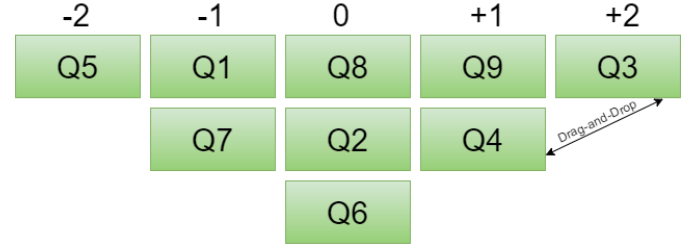


Fig. 3: A symmetrical Q-Sort Matrix

### III. CHANGING PERSPECTIVES

In this paper we present a different user interface for the Q-Sort matrix as a major component to a DELPHI Method based analysis [8,10]. The vast majority of existent web-based solutions put a heavy workload on the expert, which is not desired and ends up being counterproductive, as the focus on providing carefully thought out answers is lost [4]. Additionally, this is an issue to study owner(s) as well, since a long time will be wasted in configuring an important survey, just to see it being badly used due to very poor and confusing UI designs. We aim to change these issues by presenting a more modern and interactive tool for both configuration and execution of surveys, possibly executed in a multiple round model, that will allow the study owner(s) and experts to make the most out of the methodology. As discussed and analysed in [1], survey analysis performed with the method in discussion will take up to 4 rounds (typically 3), depending on the number of questions in the survey, which makes having an intuitive and clean UI the cornerstone for a successful survey. The main idea behind the presented work is to eliminate some steps in both the configuration and actual execution stages that the authors see as being redundant. For example, there should be no intermediate step between deciding which questions to place in the matrix and actually placing them in the matrix itself, and, instead, it should be a unified and straightforward step, which is achieved by allowing the expert to drag the questions to the specified position in the matrix, giving him more flexibility in what concerns both the matrix completion and choice of which questions are first placed in the aforementioned matrix. This is one of the most significant features of this work.

### IV. THE PROTOTYPE AND TECHNOLOGY STACK USED

It was chosen to develop this prototype as a web-based solution instead of a stand-alone offline one, because, in the recent years, both industry and research have seen a huge increase in demand of deploying what were originally stand-alone, offline applications into the online model. For making this task manageable for developers and researchers,

the use of what is known as a "Technology Stack" is generally considered good practice. A technology stack can be described, as being a set of technologies that are linked together and comprise the different components of a web application. A stack is generally identified by the initials of the technologies that comprise it, and, while these are (generally) interchangeable, the role of each component remains the same. The most popular stack is known by the acronym LAMP:

- L - Linux, as the operating system;
- A - Apache, as the server;
- M - MySQL, as the persistency (database) layer;
- P - PHP/Python, as the server-side scripting language;

While the above stack was very popular and allowed people to develop complex applications easily, it had the issue of being a *cross-language* stack, which meant that usually a team was needed to produce applications developed with the aid of all these components. Usually, a part of the team would develop the "visual and interactive" components, which are the building blocks in the "Front-end" of the application, while the remaining elements of the team would configure the database and servers, handling what is commonly known in the programming jargon as the "Back-end" of the application, i.e., the components that "support" all the core functionality. A more modern and unified approach appeared in the recent scene of web applications development, unified around the JavaScript language, known by the acronym of MEAN. Like its predecessor, the MEAN stack works under the same principle of unifying a different set of technologies to create a complex application, however, unlike LAMP, the MEAN stack has a major advantage: everything is written in the same language. MEAN stands for:

- M - MongoDB, as the database;
- E - ExpressJS, as the server framework;
- A - AngularJS, as the *front-end* language;
- N - NodeJS, as an event-driven I/O server-side JavaScript environment;

It is important to refer that the concept of connecting distinct components to create larger and more complex applications is not new and it is not connected to the uprising of "Technology Stacks", instead, these stacks appeared thanks to a software architectural pattern labelled MVC (Model-View-Controller).

The MEAN stack promotes the development of applications with the MVC software architectural pattern as the main driving philosophy behind all the application logic from the layouts until server configurations. In its essence, the prototype we present has a logical structure that allows for such separation of concerns to be naturally embedded in our application. From the view point of the study owner, the goal of our interface is to allow for a dynamic matrix, i.e., one where the study owner can reorder cells and manipulate its configuration at will. The questions are automatically adjusted to the current number of cells that compose the matrix,

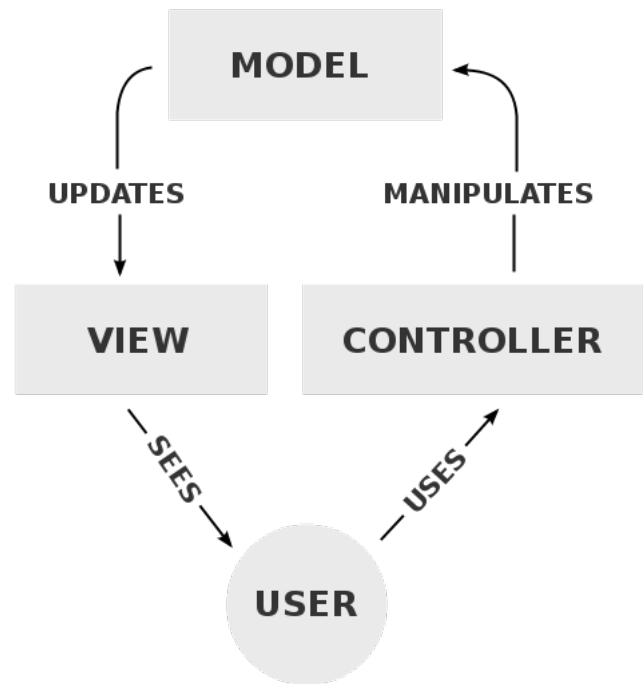


Fig. 4: MVC flow and separation of concerns

to maintain consistency. In the most recent version of the prototype (see Work To Be Done) the matrix is generated automatically, according to a simple adaptive algorithm that is used to ensure the symmetric structure of the matrix. The idea for this algorithm raised from the structure of a perfect, fully symmetric Q-Sort matrix, as the one presented below:

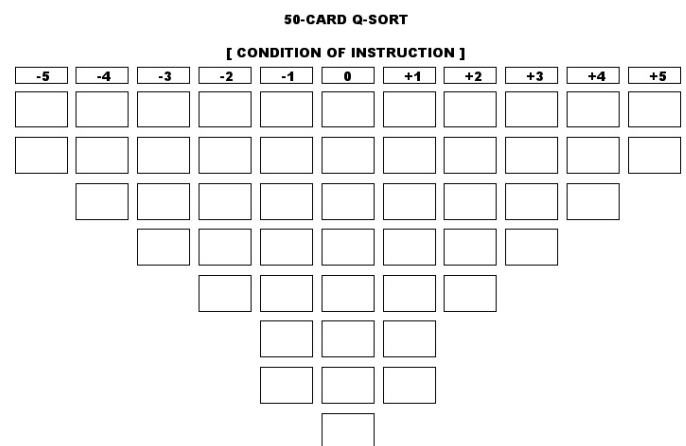


Fig. 5: A symmetrical Q-Sort Matrix

The basis for the adaptive algorithm was devised as follows:

It is known that, for an arithmetic progression that yields the sum of the first  $K$  odd numbers, we have the closed sum:

$$\sum_{i=1}^K (2i - 1) = K^2 \quad (1)$$

Let us look at the example where the number of questions  $N$  equals 9. It is easy to reason that the generated matrix will have 3 rows, with the first one (numbering the rows in a top-down fashion) comprising 5 cells, the second one, comprising 3 cells, and the last one comprising a single cell. These numbers are exactly the numbers that are derived from the order of the odd numbers:

$$1, 3 \dots 2N - 1 \quad (2)$$

Using the above ordering, it is easy to iteratively build a matrix when the number of questions in the survey is a perfect square or a perfect square plus a multiple of  $2\sqrt{(N_{questions})} - 1$  which corresponds to repeating the wider column in the matrix as needed. This approach caused some matrices to appear very unbiased and, as such, it was decided to "extend" the basic Q-Sort matrix by allowing two central columns instead of one, which gave the matrices a better and more "compact" look n' feel. We state that this matrix configuration shall keep the results unbiased (as desired) if an according statistical analysis is performed.

## V. WORK TO BE DONE

For future work, improvements to the existent DELPHI Method Online Tool will be made, which will result in the development and subsequent conclusion of the Master's thesis of colleague Lilian Gomes. She will continue to implement this prototype and later develop an interactive tool allowing the automation of all procedures related to the implementation of the Q-sort method. The user interface will be enhanced taking into account aesthetic and minimalist aspects of design producing a visually appealing result for the end users. For example, different levels of colors will be used, as mentioned in section II, giving the experts a clear perception of the heterogeneous distribution of values in the matrix. Another example of the user interface development is the creation of interactive boxes using a flexbox layout. This layout will allow for a visual and interactive feedback on already filled positions, consisting in visual information about the question and options to lock and remove the question.

## VI. CONCLUSION

We showed the utility and value of our adaptive Q-Sort Matrix Generation method based on the number of questions in a survey [5]. This approach has improved usability and interaction over the interfaces implemented and discussed in [1] since it is less demanding to the user in what concerns the number of clicks and information visualisation, and, it is expected that it will allow faster iterations of the method, making the results gathering stage less troublesome for the study owner(s). The experts have a cleaner view of the Q-sort matrix, and by dragging and moving the questions around in a visual fashion, they can focus on giving precise and more accurate answers instead of struggling with poor UI designs.

It is our goal to allow for institutions and people using this methodology that they can make the best use out of it. For that matter, it was hosted on the web, a prototype of our interface, as seen on section [1], where preliminary studies [7] and use cases of the interface can be conducted and evaluated. It is our hope that our Adaptive Q-Sort Matrix Generation Prototype can become a useful interface for both research and development in the fields [6] of Human-Computer Interaction and Information Systems [1,2].

## VII. ACKNOWLEDGMENTS

Several people gave important contributions that allowed this paper to be written. Our sincere thanks to: Professor Daniel Gonçalves, Bruno Dias, Rodrigo Lourenço, Bruno Cardoso, João Miranda and Inês França Martins for all the insights and comments.

## REFERENCES

- [1] L. D. Santos. Factores Determinantes do Sucesso de Serviços de Informação Online em Sistemas de Gestão de Ciência e Tecnologia. *EEUM*, Guimarães, Portugal, 2004.
- [2] J. A. Meloche. Q Methodology as a research methodology for human computer interaction. *OzCHI '99*, Springfield, Illinois, USA, pp. 149–152, 1999.
- [3] K. O'Leary, J. O. Wobbrock, and E. A. Riskin. Q-methodology as a research and design tool for HCI. *CHI '13*, pp. 1941–1950, 2013.
- [4] M. Ugur, D. Shastri, P. Tsiamyrtzis, M. Dcosta, A. Kalpakci, C. Sharp, and I. Pavlidis. Evaluating smartphone-based user interface designs for a 2D psychological questionnaire. *UbiComp '15*, 2015.
- [5] K. Gosha, W. Moses, J. Gilbert, and A. Wachs. Creation of a voice user interface survey for dynamic service feedback. *ACM SE '10*, 2010.
- [6] J. Block. The Q-Sort Method in Personality Assessment and Psychiatric Research. *Charles C Thomas Publisher*, Charles Sturt University: Charles Sturt University, 1961.
- [7] S. D. Gosling, S. Vazire, S. Srivastava, and O. P. John. Should we trust web-based studies? A comparative analysis of six preconceptions about internet questionnaires. *American Psychologist* 59, 2 (2004), 93.
- [8] Delphi method, Wikipedia. [https://en.wikipedia.org/wiki/Delphi\\_method](https://en.wikipedia.org/wiki/Delphi_method)
- [9] Q methodology, Wikipedia. [https://en.wikipedia.org/wiki/Q\\_methodology](https://en.wikipedia.org/wiki/Q_methodology)
- [10] C. Okoli, and S. Pawlowski. The Delphi method as a research tool: an example, design considerations and applications. *Information & Management*, Volume 42, Issue 1, December 2004, Pages 1529.