Grant Agreement no: 690770

Ship Lifecycle Software Solutions (SHIPLYS)

Project Deliverable Report

D2.2 A report on templates for business case and ROI analyses

Version:	1.0
Author:	Arijana Milat (as2con)
Contributors:	Nataša Golik Klanac (as2con), Tanja Pavlović Flegar (as2con)
Internal reviewers:	Darko Frank (as2con), Pantelis Anaxagorou (NTUA), Thomas Koch (AES), José Ignacio Zanón Millán (SOERMAR), Elif Oguz (SU), Jane Allwright (TWI), Ujjwal Bharadwaj (TWI)
Deliverable due date:	2017-02-01
Actual submission date:	2017-02-01
Work package:	WP2
Task:	T2.3
Dissemination level:	Public
Lead beneficiary:	TWI (WP leader), as2con (task leader)
Status:	Final

VERSION AND CONTROLS

Version	Date	Reason	Editor
0.0	2017-01-20	Draft	Arijana Milat
1.0	2017-01-26	- expanding chapter on ROI analysis; - minor text modification; - further elaborated functional characteristics; - added Appendices A & B; - updated QFD table and results (section 2.3)	Arijana Milat

Acknowledgement:

The research leading to these results has received funding from the European Union's Horizon 2020 research programme under grant agreement No. 690770.

EXECUTIVE SUMMARY

Introduction

This deliverable presents the work progress in Task 2.3 *Setup templates for the business case and ROI analyses.* The business case presented here for the development of SHIPLYS functionality in response to end-users' requirement is determined by a formal procedure using Quality Function Deployment (QFD). The QFD process is used to consolidate and prioritise the technical characteristics to be developed in the SHIPLYS suite of software.

Aims and Objectives

In accordance with the project's DoW, the objectives of Task 2.3 are to outline the needs and constraints of the SHIPLYS end-users, the ROI expectations and the stage gate processes for the demonstration of commercial targets. The overarching aim of this deliverable is to identify priorities in the development of the technical features of the SHIPLYS software so that its value, from the perspective of the end-users, is maximised.

Summary of the results

The methodology used is Quality Function Deployment process. The results show which users' requirement are relevant for the users and what their level of importance is. Thereafter, 22 technical characteristics are defined so they can satisfy users' requirements. Through a preliminary analysis of the complexity of each characteristic's development, the functional characteristics have been prioritised considering the importance level of the corresponding requirement.

A template for Return on Investment (ROI) has also been presented as part of this Deliverable with a view to conducting analysis in the future when relevant data is available.

Work done in this Deliverable lays the background to other tasks within the project, particularly T9.3 on business plan and exploitation.

CONTENTS

1 Introduction

This deliverable represents the work progress in Task 2.3 *Setup templates for the business case and ROI analyses* scheduled to run from the beginning of the project until the 5th month. The objectives of Task 2.3 are to outline the needs and constraints of the users, the ROI expectations and the stage gate processes for the demonstration of commercial targets. Within the task, a formalised process such as Quality Function Deployment has been used to consolidate and prioritise the SHIPLYS software functional characteristics.

The goal of the SHIPLYS project is to develop virtual modelling tools to reduce time spent, capital investment and potential environmental and risk impacts during early ship design, and particularly aims at SME shipyards and design offices as the end-users of the software. In order to make a more competitive software, the end-users' requirements and needs have been thoroughly investigated and after defining the most relevant requirements, the software functional characteristics have been determined in a way to respond these requirements.

Defined and prioritised functional characteristics will be assigned to the work packages. Also, the outcomes from this task will be used in the development of business plan and exploitation in WP9.

2 Quality function deployment process

Quality function deployment (QFD) is "an overall concept that provides a means of translating customer requirements into the appropriate technical requirements for each stage of product development and production" (Sullivan, 1986). In SHIPLYS project, this technique is used to consolidate and prioritise the software functional characteristics which will be then assigned to all work packages. These functional characteristics have to respond to the customer/user requirements where the potential users of the SHIPLYS software are shipyards and design offices.

The steps of QFD process are:

1. Collect and analyse users' needs/requirements
2. Prioritise users' requirements
3. Define the software functional characteristics
4. Define the functional characteristics interdependencies
5. Define the functional characteristics difficulties
6. Define the relationship between user requirements and functional characteristics

According to (Elboushi & Sherif, 1997), the QFD process enables one to capture requirements, and to produce specifications and designs that are faster, more robust, and more consistent.

QFD uses a planning matrix referred to as the House of Quality (HOQ) to discover interrelationships between users' requirements and product functional characteristics, as shown in Figure 1. Relating the users' requirements to the functional characteristics will show the strength of the relationship between them; whether the requirements are addressed fully and properly or whether the final product will have difficulty in meeting users' needs.

The HOQ matrix comprises of six key blocks:

- Users' requirements – It includes a set of users' desires, expectations and requirements from the product.
- Functional characteristics – It consists of a list of technical descriptors that explain about how the users' needs can be fulfilled.
- Relationship matrix – It is the interior of the house, and explicates the relationship between users' requirements and software characteristics using symbols or numbers.
- Technical correlation matrix – It is also termed as roof of the matrix and demonstrates how each of the functional characteristics influences each other.
- Importance weighting – It encompasses the quantified users' needs and positions them in order of their importance.
- Difficulty weighting – It describes how difficult and costly it is to deploy each functional characteristic.

As a result of a HOQ matrix, the functional characteristics are evaluated to decide their relative significance and are ranked accordingly.

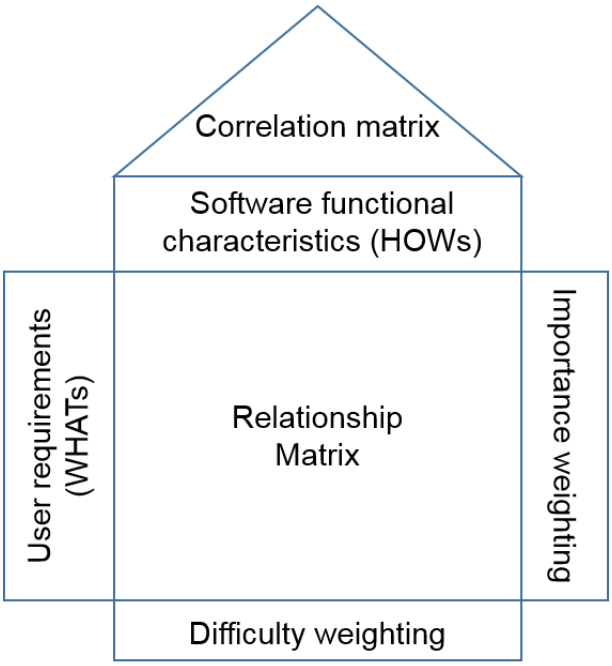


Figure 1: The house of quality (Elboushi & Sherif, 1997)

2.1 Users' requirements

The first step of QFD process was to gather and organize users' requirements. The goal was to gain feedback from a variety of users with emphasis on shipyards and design offices as they represent the target users according to project Description of Work (DoW). The users were asked to consider their expectations

of a software described as a virtual modelling tool to reduce time spent during the early ship design and to enable the optimal design with life cycle cost assessment, environmental and risk assessment.

The process was:

1. To define the list of users' requirements
2. To prioritise the defined users' requirements based on a questionnaire intended to find out project partners and Stakeholders' Committee opinions

Sources in obtaining the users' requirements were:

- project partners (consortium) including three shipyards and two design offices
- literature on user requirements in shipbuilding and software development ((Prasad & Chakraborty, 2016), (Hadjina, Matulja, & Rubeša, 2015))
- Stakeholders' Advisory Committee (SAC) - an external industrial advisory group comprising a number of managers from major stakeholders interested in the objectives and results of the SHIPLYS project

2.1.1 Defining list of users' requirements

At the second month of the project, an open-ended question was sent to the project consortium to gather all possible requirements that are necessary and important for the development of a software for an early ship design. Also, the sources mentioned above have been used in defining the initial list with all possible requirements. The collected requirements were very detailed and numerous and needed to be organized. Therefore, the requirements were analysed with an iterative approach, using deduction and inductive reasoning, and the list has been finalised. The list was checked internally and through two external experts so that it is understandable and meaningful. The results of the process are presented in Table 1.

Table 1: List of users' requirements

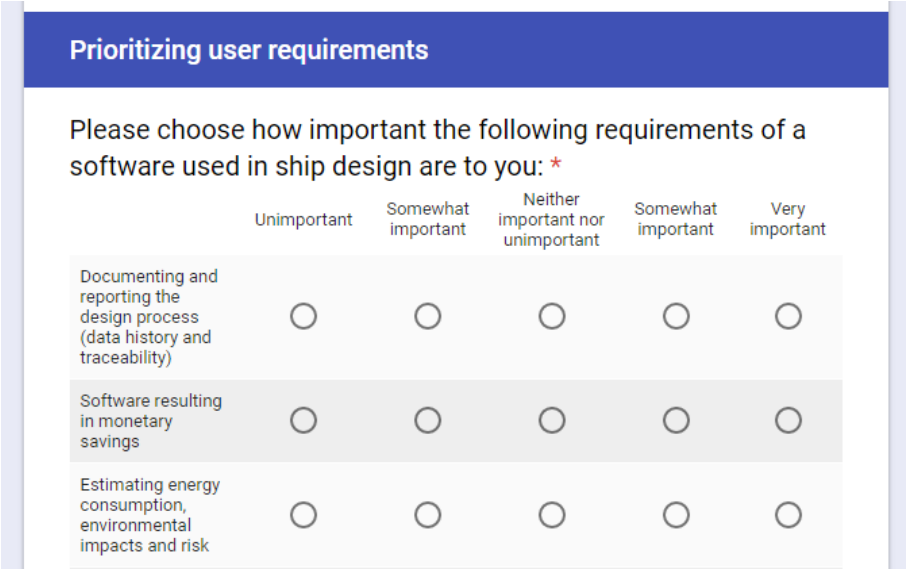
Main need / requirement category	Category definition	Need/requirements sub-categories
Reliability	Being free of doubts and uncertainties	Being confident in calculations and data
		Being confident in the software technical performance
		Supporting ship design in accordance with rules and regulations
		Documenting and reporting the design process (data history and traceability)
Convenience	Doing work with little or no effort, stress or limits	Having an intuitive and clear user interface
		Requiring no special IT skills
		Having simple installation process

Competence	Making informed design decisions, performing the work well, maintaining high level of work quality	Estimating energy consumption, environmental impacts and risk
		Making ship behaviour predictions (seakeeping, flooding)
		Being able to compare different ship designs and identify optimal solutions
		Estimating design work activities and volumes
		Providing input for production process
Efficiency	Doing work fast, reducing time	Having flexibility in ship design modification
		Making quick estimations
		Automating design processes
		Enabling a variety of information handling and processing options
Profit	Increasing income or reducing costs	Gaining value for money
		Software resulting in monetary savings

2.1.2 Questionnaire to prioritise the users' requirements

The second step was to prioritise the users' requirements. Therefore, a questionnaire was prepared and circulated to project partners and stakeholders with the aim to prioritise the requirements on a 5-level scale ranging from unimportant to very important. The method used in prioritising is based on Likert Scale that requires a respondent to indicate a degree of agreement or disagreement with the variety of statements related to the attitude or object. It is also called summated scales, because the scores on the individual items are summed to produce a total score of the respondent (Kumar, Aaker, & Day, 2002). According to (Allen & Seaman, Jul 2007), Likert scales are a common ratings format for surveys.

The questionnaire was prepared in a tool Google Forms. In order to better understand the importance of each requirement, the questionnaire was divided in two parts. In the first part, the users have been requested to assign a level of importance to each requirement from Table 1 (sub-category). The example is shown in Figure 2. In the second part, the users have been requested to sort the main categories of the user requirements. The complete questionnaire is attached in Appendix A.



Prioritizing user requirements

Please choose how important the following requirements of a software used in ship design are to you: *

	Unimportant	Somewhat important	Neither important nor unimportant	Somewhat important	Very important
Documenting and reporting the design process (data history and traceability)	○	○	○	○	○
Software resulting in monetary savings	○	○	○	○	○
Estimating energy consumption, environmental impacts and risk	○	○	○	○	○

Figure 2: Questionnaire example – Part 1

2.1.3 Respondents

In order to reach more responses, the link to the questionnaire was sent via e-mail to:

- Project partners
- Stakeholders' Advisory Committee
- Target group of companies in wider network (relevant shipyards and design offices)

The questionnaire was completed by 38 persons from nine different countries, five user groups and different departments, providing a variety of views.

Table 2: Description of respondents of the questionnaire

Countries	Type of companies	Positions
Croatia Spain United Kingdom Greece Republic of Korea Bulgaria United States Italy Portugal	University Shipyard Shipyard Technical Advisory Design office Classification society Research and Test Center Shipping company Ship operator Supplier Ship Management Institute	Researcher, Senior Designer General Manager Design Engineer, Project Engineer Executive Advisor to the Management Board Senior Project Manager Head of R&D Planning Business development Naval Architect Software Development Manager Production Manager Plan approval specialist

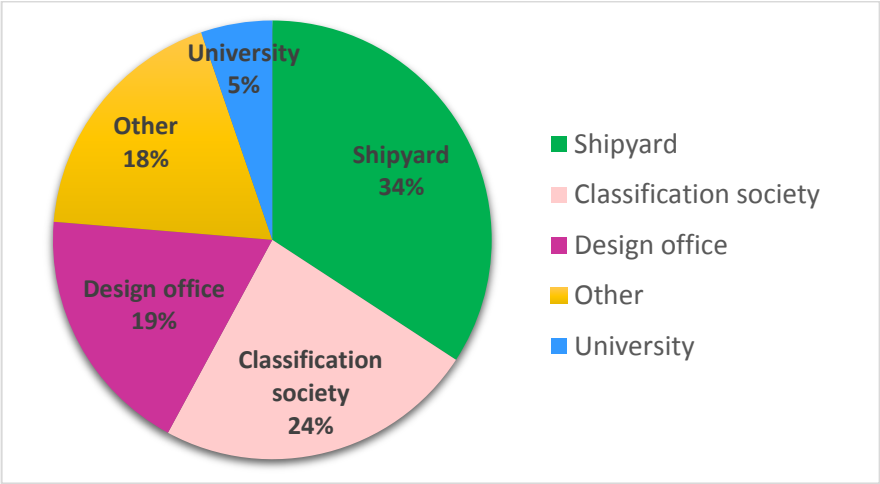


Figure 3: Division of respondents by type of company (Others are Supplier, Research and Test Center, Ship operator, Institute, Ship management)

2.1.4 Results

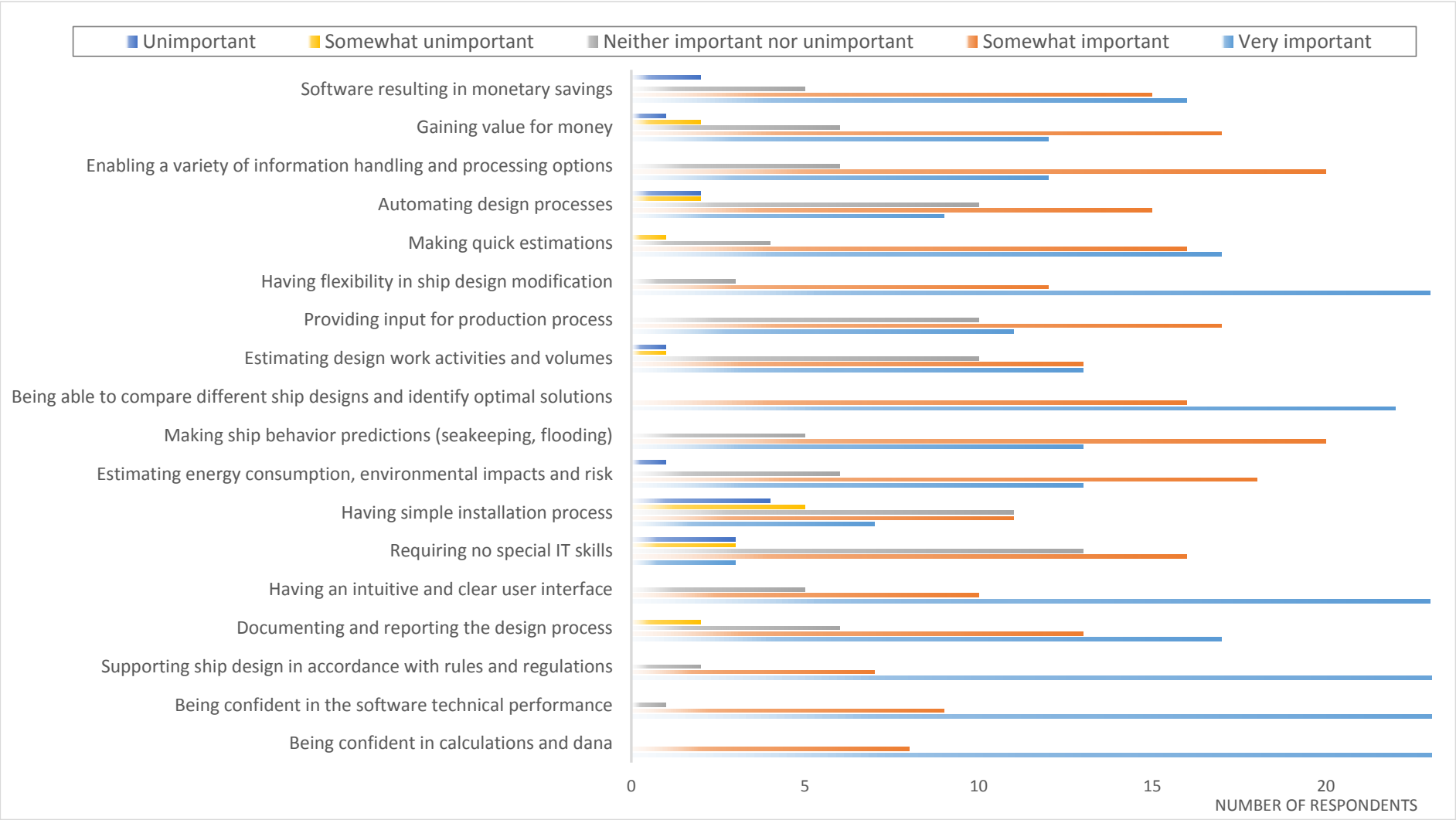


Figure 4: Questionnaire results on Part 1

Based on the responses to the questionnaire, the main categories of the users' requirements are ranked as follows:

1. Reliability – being free of doubts and uncertainties
2. Competence – making informed decisions, performing the work well, maintaining high level of work quality
3. Efficiency – doing work fast, reducing time
4. Convenience – doing work with little or no effort, stress or limits
5. Profit – increasing income or reducing costs

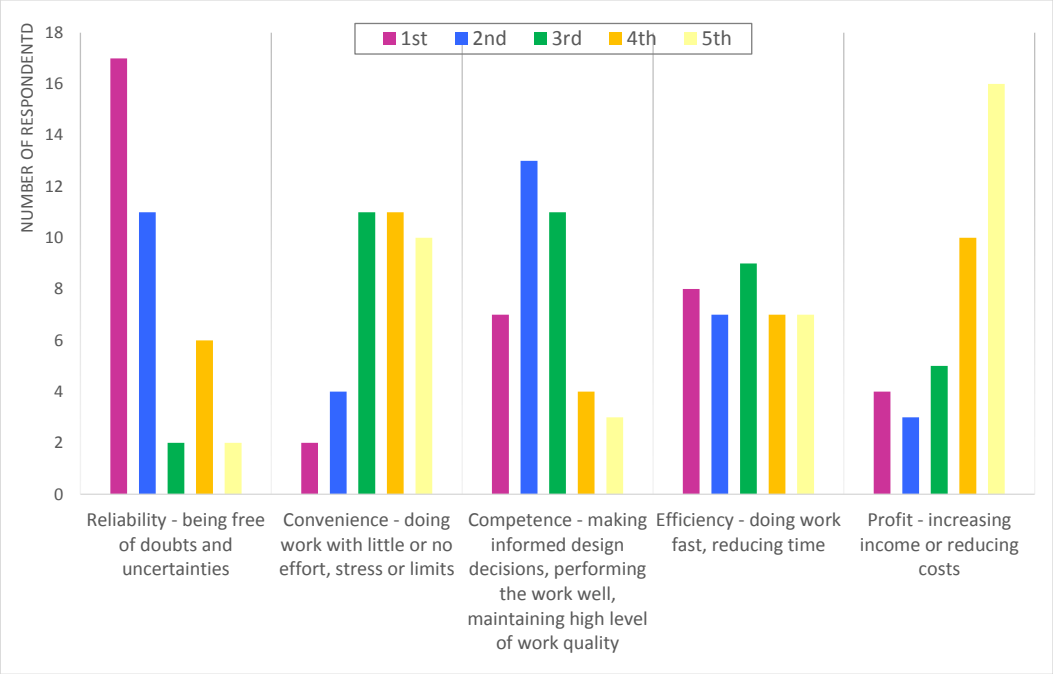


Figure 5: Questionnaire results on Part 2 (1st is most important and 5th is least important)

The results on both Part 1 and 2 are also expressed in a form of pie charts which are given in Appendices B and C.

The importance rating of each requirement is measured in a way in which the percentage of assigned importance level is multiplied with the corresponding weighting factors from Table 3. The ranking from 1 to 9 has been used as all the requirements are important and close to each other and the wider scale makes finer distinction between each requirement.

Table 3: Importance weighting factor

Very important	9
Somewhat important	7
Neither important nor unimportant	5
Somewhat unimportant	3
Unimportant	1

After processing the questionnaire results, the users' requirements have been prioritised as presented in Table 4.

Table 4: Prioritised users' requirements with assigned level of importance

Need/requirements sub-categories	Importance level (mean value)	Spread
Being confident in calculations and data	9	7-9
Being confident in the software technical performance	8	5-9
Supporting ship design in accordance with rules and regulations	8	5-9
Documenting and reporting the design process (data history and traceability)	7	3-9
Having an intuitive and clear user interface	8	5-9
Requiring no special IT skills	6	1-9
Having simple installation process	6	1-9
Estimating energy consumptions, environmental impacts and risk	7	1-9
Making ship behaviour predictions (seakeeping, flooding)	7	5-9
Being able to compare different ship designs and identify optimal solutions	8	7-9
Estimating design work activities and volumes	7	1-9
Providing input for production process	7	5-9
Having flexibility in ship design modification	8	5-9
Making quick estimations	8	3-9
Automating design processes	6	1-9
Enabling a variety of information handling and processing options	7	5-9
Gaining value for money	7	1-9
Software resulting in monetary savings	7	1-9

Considering the weighting factor from 1-9, Table 5 shows the processed results for the second part of questionnaire that dealt with general categories of user requirements.

Table 5: Prioritised main categories of the users' requirements

Need/requirements main categories	Mean value	Spread	Rank
Reliability	7	1-9	1st
Competence	6	1-9	2nd
Efficiency	5	1-9	3rd
Convenience	4	1-9	4th

Profit	3	1-9	5th

Observing the results separately among different groups, the rankings were very similar. A slight difference was found in ranking the profit where shipyards and design offices perceived profit as more important than classification societies.

2.2 Software functional characteristics

The second step of the QFD process was to establish functional characteristics of a SHIPLYS software. The as2con team together with the partners involved in the Task 2.3 identified all the measurable functional characteristics of the SHIPLYS software which could meet the specified users' requirements.

The SHIPLYS project description of work gives an insight of what the SHIPLYS software should offer to its customers. The main focus of the software should be to enable ship design considering life cycle cost assessment, environmental and risk assessment. Within the WP2, project partners have selected and developed appropriate scenarios for the application of the SHIPLYS software. The proposed scenarios are:

- Optimise the design of a short-route ferry using a novel hybrid propulsion system
- Re-design of a SME shipyard production system for a new design of a multipurpose ship
- Optimizing / assessing different types of retrofitting, Scrubber and Water ballast management plant

Software functional characteristics were defined through steps:

1. Definition of functional software characteristics that included the following sources:
    a. project description of work (Grant agreement)
    b. partners' discussion on proposed scenarios
    c. minutes of meetings
    d. literature on ship design software ((Elboushi & Sherif, 1997)
2. Based on a-d, a list of initial software characteristics was composed and sent to partners in Task 2.3 for commenting
3. List of software characteristics was sent to software developers (Atlantec Enterprise Solutions) to comment and to determine the level of technical difficulty also assessed by as2con
4. Feedback was analysed

The project is still in its early phase and the software functional characteristics presented in Table 6 are preliminary and subject to change as the project progresses. In this phase, precise ranking of the technical difficulty of each characteristic is not feasible because the methodologies of the tools incorporated within the SHIPLYS software are still in the process of definition. Therefore, an average level of difficulty has been assigned to each characteristics based on assumptions and experience from software development.

Table 6: List of software functional characteristics and corresponding level of technical difficulty (1 meaning not so difficult to implement and 5 meaning very difficult)*

Functional characteristic**	Level of technical difficulty
Extensive database with information on ship main particulars, ship construction and operational characteristics	3

Life cycle cost assessment tool (cost of production, operation, maintenance, repair and refurbishment)	2-4
Life cycle environmental assessment tool	2-5
Life cycle risk assessment tool (performance time profiles, degradation profiles)	2-5
Multi-criteria decision support tool	2-4
Ship operation profile tool - *functions to quickly capture and collate the operational profile for a new type of ship (transport capacity, service speed, operating cost, type of propulsion)*	2-4
Ship configuration tool – *functions to quickly define the major configuration properties for a new design*	2-5
Rapid prototyping tools – *functions to automate the model generation activities for design in the areas of hull form definition, compartmentation, general arrangement, structural configuration, work breakdown, equipment and systems definition*	4
Production analysis and planning tools – *functions to perform early design production verification*	2-3
Integration of developed tools – *system integration to ensure seamless interaction between all system components including connection to external systems in terms of data synchronisation and module communication*	1-5
Ship Design Workflow controller – *management function to control and assess the database and the stages of design and analysis*	2-4
Information on software data reliability	2-5
Use of and compatibility with existing early design tools	1-5
Data history, variety and traceability	1-3
Interactive usage of component database (Libraries)	2-4
Theoretical and user manual and after-sale support	3
Concise and clear automated reports	2-3
Well defined structure & simple user input procedure	2-4
Easy installation procedure	1-2
Purchase and maintenance price of software	1-5
Software verification by 3rd party / registry	1-5
Hardware requirements – *Use of common hardware; providing economic access to computationally intense systems*	1-5

* The difficulty depends on: the actual tool, the connectivity of developed tools, level of detail / usage of existing technology

** Each characteristics will be thoroughly defined as the project progresses.

2.2.1 Software functional characteristics dependencies (Correlation matrix)

The dependencies between the functional characteristics are placed at the roof of the house of quality. The characteristics have one or more relationships to each other. These relationship describe how the deployment of one functional characteristic affects another.

The relationship between characteristics can be described as strong positive, positive, negative and strong negative. Placing these relationships in the correlation matrix can highlight the characteristics that might be in conflict with each other. Improving one characteristic can positively affect a related characteristic but also it can affect others negatively. One of the principal benefits of the roof of the HoQ is that it flags these negative relationship so they can be resolved (Tapke, Muller, Johnson, & Sieck, n.d.).

In this phase of the project, there is no need to formulate the correlation matrix because the list of software functional characteristics is only a preliminary and needs to be updated with the progress of the project.

2.3 Quality matrix

The most important step of the QFD process is to determine the strength of relationship between the users' requirements and the software functional characteristics. The strengths of relationship are indicated as:

S – strong relationship (weight factor = 5)

M – medium relationship (weight factor = 3)

W – weak relationship (weight factor = 1)

Numerous strong relationships indicate that the corresponding requirement is highly covered and on the other hand, absence or only weak relationships indicate that the corresponding relationship is not likely to be covered by the proposed functional characteristics.

Calculation of weight factors based on users' requirements is performed using the following equation (Hadjina, Matulja, & Rubeša, 2015):

$$WF = \sum_{i=1}^{n} N_i \times W_i,$$

where, N_i is i –th column and W_i is i –th weight factor based on importance level.

After multiplying each cell's value by the weight of the requirements' importance and totalling the column for each functional characteristic, the software functional characteristics have been prioritised.

The preliminary QFD table is presented in Figure 3. The relationships between software functional characteristics and users' requirements are based on the assumption that each tool can be treated as an autonomous software module.

Whats - User-client values / needs				Hows - Software characteritics																					
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Main need / requirement category	Category definition	Need/requirements sub-categories	Importance rating	Extensive database	Life cycle cost assessment tool	LC environmental assessment tool	Life cycle risk assessment tool	Multi-criteria decision support tool	Ship operation profile tool	Ship configuration tool	Rapid prototyping tools	Production analysis and planning tools	Integration of developed tools	Ship Design Workflow controller	Information on software data reliability	Use of and compatibility with existing early design tools	Data history, variety and traceability	Interactive usage of component database	Theoretical and user manual and after-sale support	Concise and clear automated reports	Well defined structure & simple user input procedure	Easy Installation procedure	Purchase and maintenance price of software	Software verification by 3rd party / registry	Hardware requirements
Reliability	Being free of doubts and uncertainties	Being confident in calculations and data	9	M	M	M	M	M	M	M	M	M		M	M	M	W	W	S	W				S	
		Being confident in the software technical performance	8		M	M	M	M	M	M	M	M		M		M			S		M	W		S	S
		Supporting ship design in accordance with rules and regulations	8	M		S	S		M		W				W	M									
		Documenting and reporting the design process	7	M			M				M			S			S	M		S	W				
Convenience	Doing work with little or no effort, stress or limits	Having an intuitive and clear user interface	8		M	M	M	M	M	M	M	M				M		M			S				
		Requiring no special IT skills	6	M	M	M	M	M	M	M	M	M	S	S		M		M	W	W	S	S			M
		Having simple installation process	6		M	M	M	M	M	M	M	M				M			W			S			M
Competence	Making informed design decisions, performing the work well, maintaining high level of task / work quality	Estimating energy consumptions, environmental impacts and risk	7	M	S	S	S	M	S		S	W	S			M	W	M			M				M
		Making ship behavior predictions	7	M			M	S	M	M	M		S			M		W							M
		Being able to compare different ship designs and identify optimal solutions	8	S	M	M	M	M	M	M	M	M	S	M		M	M			W	S				M
		Estimating design work activities and volumes	7	M	M					M	M	M	S	S		M					S				M
		Providing input for production process	7	S	S	M	M	M	S	S	M	S	M			M		S		M					W
Efficiency	Doing the work fast, reducing time	Having flexibility in ship design modification	8	M							M	M	S	S		M		S			M				M
		Making quick estimations	8	M	S	S	S	S	S	S	S	S	S	M		M		M			M				S
		Automating of design processes	6	M						M			S			M		M			M				M
		Enabling a variety of information handling and processing options	7	M	S	S	S	S	S	S	S	S	M	S		M	M	S			M				M
Profit	Increasing income	Gaining value for money	7		S											M			W		M		S		W
		Software resulting in monetary savings	7		S			M		W		S	W			M					M		S		W
			Score	315	371	306	348	308	325	312	361	332	334	274	35	372	96	252	104	79	326	68	70	85	287
			Rating	9	2	12	4	11	8	10	3	6	5	14	22	1	17	15	16	19	7	21	20	18	13

Figure 6: Quality Function Deployment process (strong relationship = red; medium = yellow; weak = green)

SHIPLYS – Grant Agreement number: 690770

GA Ref: Ares (2016) 2353538

Table 7: Preliminary list of prioritised functional characteristics*

Rank	Functional characteristic
1	Use of and compatibility with existing early design tools
2	Life cycle cost assessment tool (cost of production, operation, maintenance, repair and refurbishment)
3	Rapid prototyping tools
4	Life cycle risk assessment tool (performance time profiles, degradation profiles)
5	Integration of developed tools (data synchronisation and module communication)
6	Production analysis and planning tools
7	Well defined structure & simple user input procedure
8	Ship operation profile tool (transport capacity, service speed, operating cost, type of propulsion)
9	Extensive database with information on ship main particulars, ship construction and operational characteristics
10	Ship configuration tool
11	Multi-criteria decision support tool
12	Life cycle environmental assessment tool
13	Hardware requirements
14	Ship Design Workflow controller
15	Interactive usage of component database (Libraries)
16	Theoretical and user manual and after-sale support
17	Data history, variety and traceability
18	Software verification by 3rd party / registry
19	Concise and clear automated reports
20	Purchase and maintenance price of software
21	Easy installation procedure
22	Information on software data reliability

* Calculated based on the importance of a requirement and the level of impact of the respective technical characteristic on the requirement.

3 ROI analyses

Return on investment (ROI) is a performance measure used to evaluate the efficiency of an investment. It measures the amount of return on an investment relative to the investment's cost. To calculate ROI, the benefit (or return) of an investment is divided by the cost of the investment, and the result is expressed as a percentage (Return on investment, n.d.).

Regarding the investment in SHIPLYS software, two aspects can be considered:

A. Investment of software developer (owner of the software) – how the price of the software affects the owner's income
B. Investment of software buyer – how the software saves money through its usage

Calculation of ROI in case A requires the following data:

- Cost of manufacturing finalized and functional software (R&D + debugging)
- Sale price of the software
- Sale related costs (sales network)
- Market size (number of licences to be sold)
- Cost of acquiring IP protection
- Cost of additional staff for providing software training and customer support

On the other side, calculation of ROI in case B requires:

- Sale price of the software
- Savings that software can achieve in the form of time savings, the use of human resources and as well, the use of other software and replacing other more expensive software
- Profit that can be achieved because, through better design, the user can sell his service more expensively

Templates for the ROI calculation are presented in Table 8 and Table 9 which will be later used in the development of the business plan within Task 9.3 *Business plan and exploitation*. Within project, only a perspective for a software developer can be observed because ROI for a software buyer requires financial data and the expectations of each specific software user.

Table 8: Software developer / seller ROI*

	Years				
	1	2	3	4	…n
A) **REVENUES = A1 + A2**					
A1) Licence sales income (price of a licence times number of licence sold)					
A2) Maintenance income (price of maintenance times number of clients)					
B) **COST = B1 + B2 + B3 + B4**					
B1) R&D costs / Immaterial asset depreciation					
B2) Software testing costs + IP + training					
B3) Sales and promotion cost					
B4) Customer service costs					
PROFIT = A - B	P1	P2	P3	P4	…
CUMULATIVE PROFIT	C1= P1	C2= P2+ C1	C3= P3+ C2	C4= P4+ C3	…

*ROI is achieved when cumulative profits are greater than the initial investment (R&D).

Table 9: Software buyer ROI*

	Years				
	1	2	3	4	…n
A) **REVENUES = A1 + A2**					
A1) Using service income charged extra due to a better design service					
A2) Saving (in terms of time, employee costs, other software expenses)					
B) **COST = B1 + B2 + B3 + B4**					
B1) Licence costs					
B2) Maintenance & support costs					
B3) Staff training costs					
PROFIT = A - B	P1	P2	P3	P4	…
CUMULATIVE PROFIT	C1= P1	C2= P2+ C1	C3= P3+ C2	C4= P4+ C3	…

*ROI is achieved when cumulative profits are greater than the initial investment (1st year licence + maintenance + training).

4 Conclusion

The deliverable specified the list of user requirements relevant for the further development of the software and software characteristics. In order to understand users' expectations, a questionnaire was prepared and its results gave the level of importance of each user requirement. Furthermore, the software functional characteristics were defined in a way that they respond to users' requirements.

The initial list of software characteristics was also defined with the intention to be updated through the progress of the project. In particular, the interdependence of each characteristic and its level of difficulty need to be assessed with the project progress.

The information on input type for ROI analysis is defined and one can be conducted once the software is specified in more detail.

5 Bibliography

Allen, I. E., & Seaman, C. A. (Jul 2007). Likert Scales and Data Analyses. *Quality Progress, Milwaukee 40.7*, 64-65.

Elboushi, M. I., & Sherif, J. S. (1997). Object-oriented software design utilizing quality function deployment. *Journal of Systems and Software 38 (2)* , 133-143.

Hadjina, M., Matulja, T., & Rubeša, R. (2015). Methodology for the ship exploitation feedback inclusion for improving the ship design and production process based on adjusted QFD method. *Scientific Journal of Maritime Research 29*, 107-114.

Kumar, V., Aaker, D. A., & Day, G. S. (2002). *Essentials of marketing research, 2nd ed.* USA: John Wiley & Sons.

Prasad, K., & Chakraborty, S. (2016). A QFD-based decision making model of computer-aided design software selection. *Management Science Letters 6*, 213-224.

Return on investment. (n.d.). Retrieved from Investopedia: http://www.investopedia.com/terms/r/returnoninvestment.asp

Sullivan, L. P. (1986). Quality function deployment. *Quality Progress 19 (6)*, 39-50.

Tapke, J., Muller, A., Johnson, G., & Sieck, J. (n.d.). *Steps in Understanding the House of Quality*. Retrieved from http://www.public.iastate.edu/~vardeman/IE361/f01mini/johnson.pdf

Appendix A Questionnaire to prioritize users' requirements

SHIPLYS - Ship Life Cycle Software Solutions

SHIPLYS project aims to develop simulation and modelling tools that would minimize the cycle time and costs of ship design and production. The idea is to develop and integrate rapid virtual prototyping tools with life cycle tools that will be compatible with existing early design software.

In order to define the most important features of SHIPLYS software, we would appreciate your assistance in prioritizing user requirements.

* Required

SHIPLYS

1. **Country** *

_____

2. **Type of company** *
Mark only one oval.

- ◯ Shipyard
- ◯ Design office
- ◯ Classification society
- ◯ University
- ◯ Other: _____

3. **Your position in the company** *

4. Please choose how important the following requirements of a software used in ship design are to you: *

Mark only one oval per row.

	Unimportant	Somewhat unimportant	Neither important nor unimportant	Somewhat important	Very important
Supporting ship design in accordance with rules and regulations					
Having simple installation process					
Documenting and reporting the design process (data history and traceability)					
Automating design processes					
Gaining value for money					
Providing input for production process					
Being confident in the software technical performance					
Requiring no special IT skills					
Being able to compare different ship designs and identify optimal solutions					
Making quick estimations					
Estimating design work activities and volumes					
Being confident in calculations and data					
Estimating energy consumption, environmental impacts and risk					
Having flexibility in ship design modification					
Having an intuitive and clear user interface					
Software resulting in monetary savings					
Making ship behavior predictions (seakeeping, flooding)					
Enabling a variety of information handling and processing options					

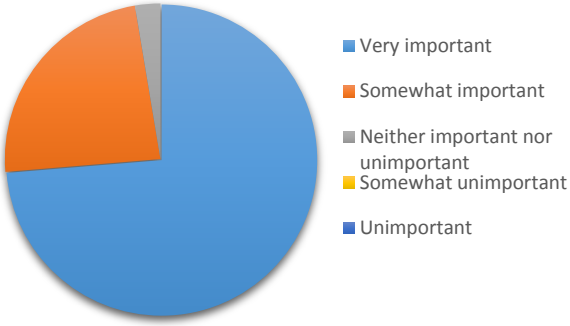
5. **Please sort the following categories of user requirements according to their level of importance (where 1 stands for the most important and 5 for the least important):** *

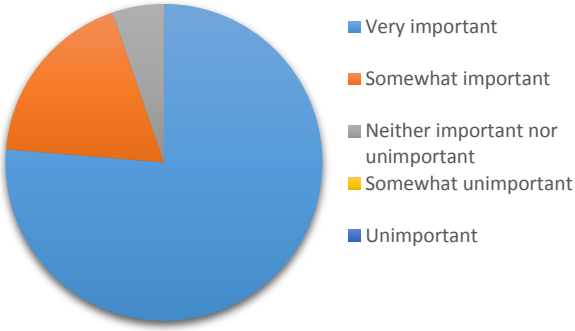
Mark only one oval per row.

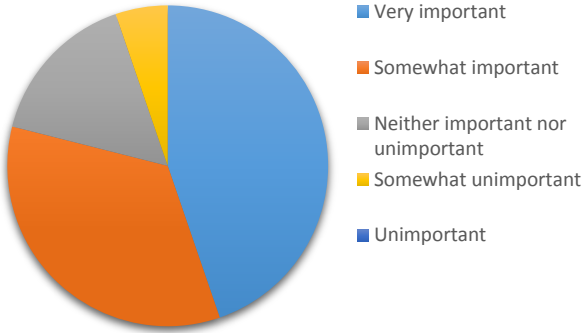
	1	2	3	4	5
Reliability - being free of doubts and uncertainties	◯	◯	◯	◯	◯
Convenience - doing work with little or no effort, stress or limits	◯	◯	◯	◯	◯
Competence - making informed design decisions, performing the work well, maintaining high level of work quality	◯	◯	◯	◯	◯
Efficiency - doing work fast, reducing time	◯	◯	◯	◯	◯
Profit - increasing income or reducing costs	◯	◯	◯	◯	◯

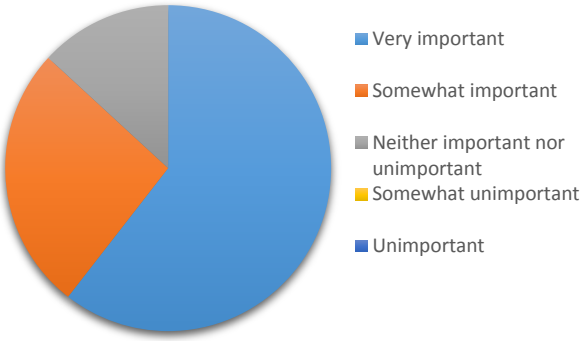
6. **Comments**

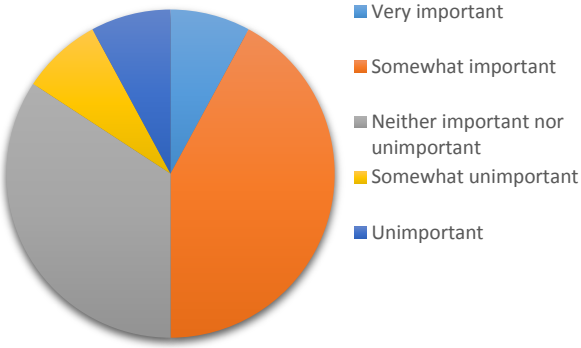
Appendix B Questionnaire results on Part 1 (pie charts)

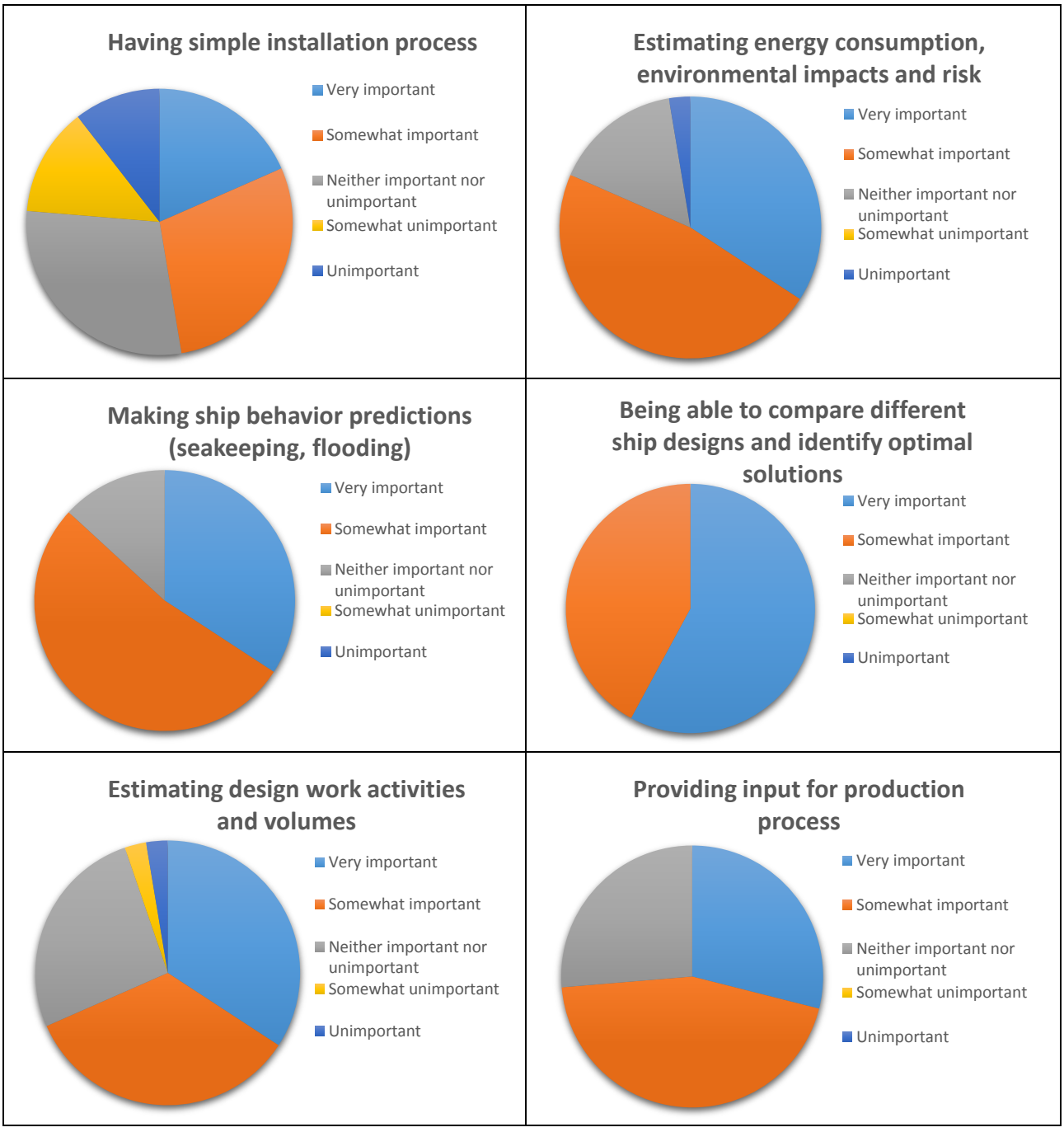


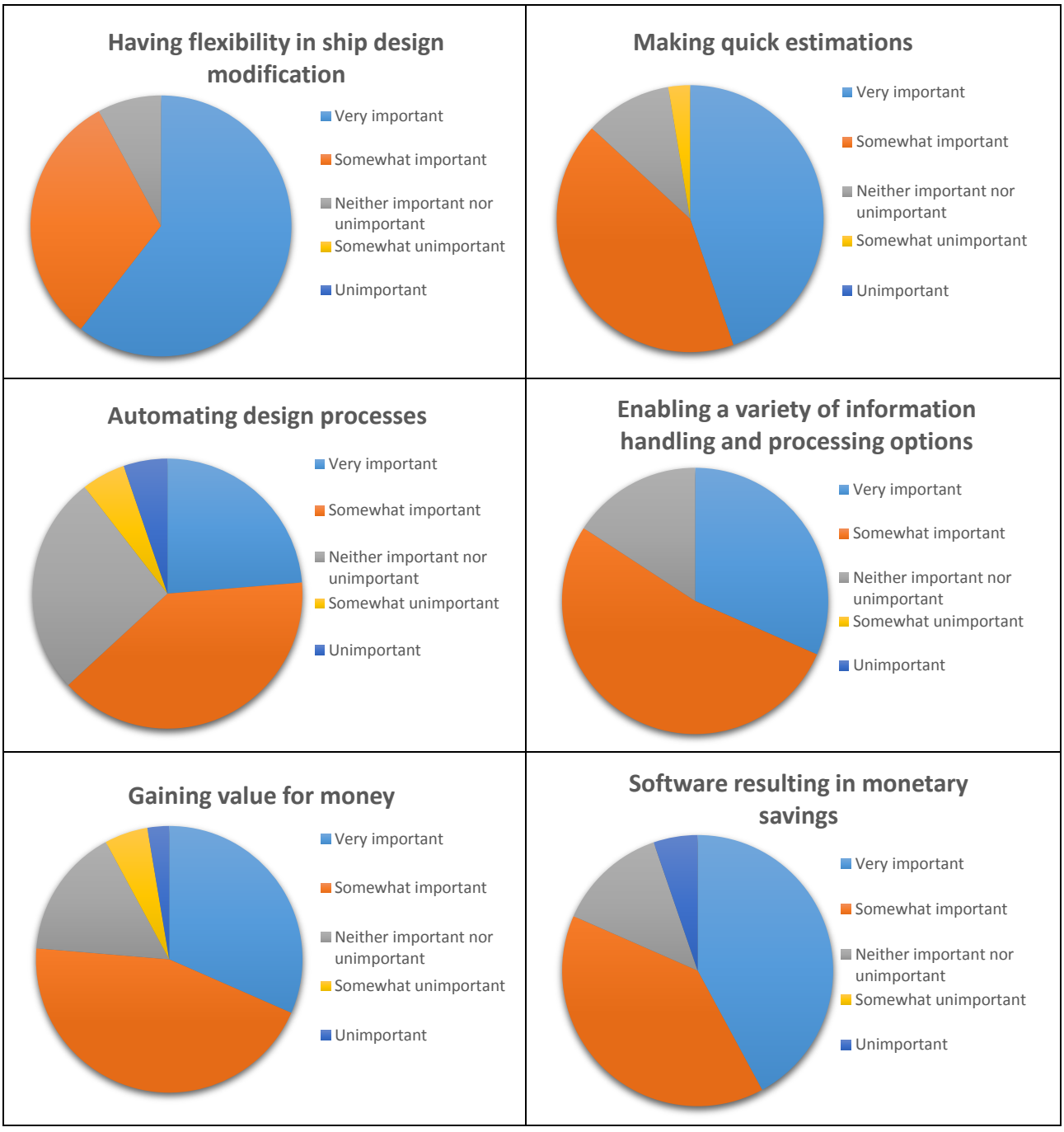









Having simple installation process
Very important
Somewhat important
Neither important nor unimportant
Somewhat unimportant
Unimportant
Estimating energy consumption, environmental impacts and risk
Very important
Somewhat important
Neither important nor unimportant
Somewhat unimportant
Unimportant
Making ship behavior predictions (seakeeping, flooding)
Very important
Somewhat important
Neither important nor unimportant
Somewhat unimportant
Unimportant
Being able to compare different ship designs and identify optimal solutions
Very important
Somewhat important
Neither important nor unimportant
Somewhat unimportant
Unimportant
Estimating design work activities and volumes
Very important
Somewhat important
Neither important nor unimportant
Somewhat unimportant
Unimportant
Providing input for production process
Very important
Somewhat important
Neither important nor unimportant
Somewhat unimportant
Unimportant


Having flexibility in ship design modification
Very important
Somewhat important
Neither important nor unimportant
Somewhat unimportant
Unimportant
Making quick estimations
Very important
Somewhat important
Neither important nor unimportant
Somewhat unimportant
Unimportant
Automating design processes
Very important
Somewhat important
Neither important nor unimportant
Somewhat unimportant
Unimportant
Enabling a variety of information handling and processing options
Very important
Somewhat important
Neither important nor unimportant
Somewhat unimportant
Unimportant
Gaining value for money
Very important
Somewhat important
Neither important nor unimportant
Somewhat unimportant
Unimportant
Software resulting in monetary savings
Very important
Somewhat important
Neither important nor unimportant
Somewhat unimportant
Unimportant

Appendix C Questionnaire results on Part 2 (pie charts showing 1-5 rankings in decreasing order of importance)