

<https://doi.org/10.70517/ijhsa4543>

A Novel Complexity Scoring Model for Non-Functional Requirements in Request for Proposal in Housing Industry Using FURPS Quality and Moscow Priority Model

Ekbal Rashid^{1,*} and Nikos E. Mastorakis^{2,3}

¹Postdoc Researcher, Technical University of Sofia, Sofia, Bulgaria

²Sector of Electrical Engineering and Computer Science, Hellenic Naval Academy, Piraeus, Greece

³English Language Faculty of Engineering, Technical University of Sofia, Sofia, Bulgaria

Corresponding authors: (e-mail: ekbalrashid2004@yahoo.com).

Abstract In this research paper we have attempted to elicit Non-Functional Requirements (NFR) which may or not have been explicitly added to the Request for Proposal (RFP) in housing industry but is important for the success of the program. We have used real time RFPs for the elicitation of NFRs. The sales proposal process, also known as response to RFP process, refers to the methodical steps a vendor takes when developing a proposal in response to a buyer's RFP which will have details of integrating systems, Software platforms, users, business outcomes, limitations, functional requirements, and may be non-functional requirements (NFR). There is some work done by researchers on assessing opaqueness of NFRs and traceability of NFR. But there has been no work on a complexity scoring model for NFRs which enables a vendor to respond to an RFP with best price and schedule. The paper proposes a novel complexity scoring model for NFRs in RFP in Housing industry by using FURPS (Functionality, Usability, Reliability, Performance, Supportability) quality attribute model in conjunction with MoSCoW (Must Have, Should Have, Could Have, Won't Have) priority model by using mathematical formula. The working model is broken down in steps for Sales and pre-Sales teams of the Vendor and is ready for adoption.

Index Terms housing, construction cost, complexity scoring model, RFP, NFR, FURPS, MoSCoW, linear regression, scatter plot

I. Introduction

It is the sales pitch that starts a dialogue between an organization and the stakeholders outside the project especially a vendor or service provider. An organization establishes a formal, logical presentation to an outside worker or project donor by creating a proposal. A project proposal (also called RFP) outlines the project's initial goals, objectives, and strategies for accomplishing them. It serves as the project's conceptual foundation. It is typical for a RFP to include a list of the tasks or activities that will be involved in the project development, and an explanation of the project goals and its significance.

To get executive support for a new project, program, or service, a proposal is first and foremost necessary. Second, it is employed to encourage team members to consider the same objectives and top priorities. The organization can use it to track when new hiring decisions, vendor assessment and onboarding, or budget adjustments are necessary.

RFP is a formal document along with questionnaire-style content which is distributed to potential vendors by the organization that wants to purchase a product, software, or a service. It also meant to collect vendor information in an organized, standardized format that makes comparisons quicker and simpler. The procedure is intended to be a methodical and objective approach to sourcing and buying. It is used to choose the best vendor more confidently for long-term partnerships because of how thorough and detailed it is. Strategic sourcing is another name for this all-encompassing method of procurement. Considering this, RFPs have been instrumental in assisting almost all businesses in achieving their goals.

Project Details may have business problem statements, Project goals and Scope including functional requirements. Whereas Technical requirements may have preferred platform, current environment, data accessibility, service communications, identity management, deployment [1] etc. Clear Functional and Non-functional requirements are critical to project success and having these in RFP essentially ties it to business problems or outcomes. But data indicates that one of the major causes of up to 70% of projects failing [2] is due to inaccurate requirements. The real issue, however, lies beneath that statistic: Teams generally pay too much attention to functional requirements (what the system does) and neglect to consider non-functional aspects (how the system does it). And one of the possible reasons is that NFRs are opaque in RFP and may need deliberate extraction.



Figure 1: Contents of RFP [3]

All parties involved in the design of a system solution are initially concerned with whether the system will function, that is, whether it will meet the functional requirements. But it's also crucial to consider how the solution will operate, in addition to what they will accomplish. The fact that the system functions at a fundamental level and completes its primary task does not guarantee that the end solution/product will offer a positive user experience or even satisfy the business outcomes. This is where Non-Functional Requirements become critical. Non-functional requirements help deliver a positive user experience and ensure that product performs to the standards set by stakeholders, clients, and customers.

Hewlett-Packard-created FURPS technique is now widely used [4] in the software industry for validating prioritized functional and non-functional requirements after understanding the needs and demands of the clients. Supportability, reliability, functionality, usability, performance, and are some of its top qualities. FURPS, an acronym of these qualities serves as the technique's very name. To define quality metrics to support each stage of the system development process, quality factors as defined by FURPS can be used.

- 1) Functionality refers to functional specifications that typically represent the primary solution features.
- 2) Usability is a non-functional requirement that considers aspects of the user interface like aesthetics and consistency.
- 3) Reliability is a kind of non-functional requirement which is concerned with attributes like the solution's accessibility.
- 4) Performance is a kind of non-functional requirement which is focused on the solution's throughput, response time, and recovery time.
- 5) Supportability is again a kind of non-functional concern focused on aspects of the solution like maintainability and scalability.

The MoSCoW method, a prioritization method used in project management, business analysis, overall management, and software development, is used to reach an understanding with stakeholders on the value they place on the delivery of each requirement.

MoSCoW is particularly effective when used in projects. Additionally, it resolves issues with simpler prioritization techniques based on relative priorities because definitions of these priorities are lacking or need to be defined, the use of a straightforward high, medium, or low classification is weaker. Additionally, this classification doesn't give the company a clear indication of what to anticipate. Indecision is also permitted by categorizations with a single option for the middle, such as "medium." Simple sequential priorities of 1, 2, 3, 4, etc. are weaker because they deal with similar-importance items

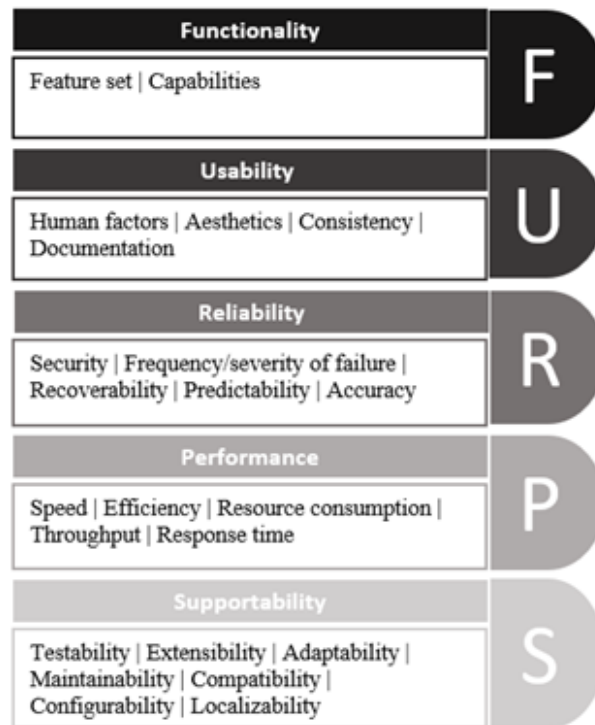


Figure 2: FURPS Model attributes [3]

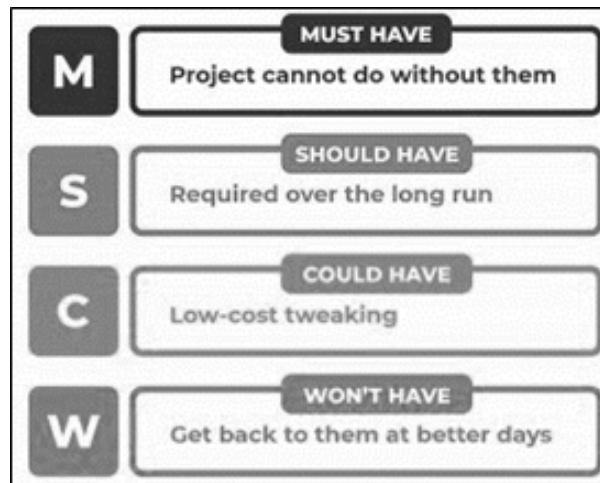


Figure 3: MoSCoW Model priorities [5]

less successfully. Whether an item should be one place higher or lower may be the subject of protracted and argumentative discussions.

These lists the "Minimum Usable SubseT" (**MUST**) of conditions that the project promises to meet. What will happen if this requirement is not fulfilled? It is a "Must Have requirement" if the response is "cancel the project; there is no point in implementing a solution that does not meet this requirement." It is a "Should Have" or "Could Have requirement" if there is a workaround, even if it is laborious and manual. A requirement that has been classified as a "Should Have" or "Could Have" simply means that it's not for the day0 and hence delivery may be not assured.

Examining the level of suffering brought on by the requirement's non-compliance, expressed in terms of business value or the number of individuals affected, is one method of separating a "**Should Have**" requirement from a "Could Have requirement".

Since they could only be delivered in full in the best-case scenario, these requirements serve as the primary source of contingency. One or more of the "**Could haves**" offer the first option for what could be dropped from the deadline when an issue arises, and it puts the project in jeopardy.

The project development team has decided that these requirements will not be delivered (during this timeframe). They are listed in the "Prioritized Requirements List" where they assist in defining the project's scope. This prevents them from being formally reintroduced later. This aids in controlling expectations that certain requirements might, at least temporarily, fail to be developed and delivered in the final solution. "Won't Haves" can be very effective in maintaining the current emphasis on the more crucial "Could Haves", "Should Haves", and especially the "Must Haves".

The paper will provide a step by step process of creating the Complexity Scoring model for the NFRs from RFP by using the FURPS in conjunction with MoSCoW.

II. Literature review

The business vision from RFP serves as the foundation for every project. A list of prioritized requirements that support achieving the business vision is linked to it [5].

A business case, which details the project in terms of the value it will return to the business, is also connected to the business vision. Depending on the organization, this business case may be an informal agreement, or it may be formally defined, outlining the expected "Return on Investment (ROI)" to support the project's cost.

When the suppliers (also referred as Service providers) are imperfect substitutes, the buyer can profit from a request for proposal (RFP) stage that comes before the negotiation stage [6]. There will be multiple suppliers to respond to the RFP. And suppliers are hard pressed to assess the scope accurately with minimal information in RFP so that they can calculate the viable price for the service.

The project scope and business goal stated in RFP would later become the input to software design and development. An initial set of NFRs which are stated as goals increased awareness and commitment and sparked systematic growth [7].

But eliciting NFR is not easy, rather its challenging because it needs very specialized knowledge [8]. The required knowledge typically relates to a specialty area, such as security, performance in terms of response time and resource use, accessibility, and interoperability, among other things.

Non-functional requirements specify characteristics of the software system that guarantee effectiveness while enforcing any limitations and constraints on the design [9].

Therefore, it is not always possible to define a good NFR with a cursory or just academic understanding of these topics [10]. Additionally, attempts by experts to define these requirements without the necessary knowledge frequently result in arbitrary or even useless specifications.

In most teams, eliciting and documenting NFRs is given very little importance or priority [11]. The effects of poor NFRs documentation cascade downward to many aspects of development practices, processes, and the quality of deliverables when very little resources are allocated to this area, especially in agile environments.

In the paper [12] the authors proposed a framework to identify poorly written RFPs in which they evaluate NFRs in RFP through a series of steps starting from selection of NFR metrics, Categorization & abstraction of NFR and definition of grading for evaluation. But again, just identification of poorly written RFP is not enough. The paper assumes that RFPs are well written in today's world. Whereas a quick review of available RFPs over the internet will clarify that RFPs are majorly written without too many details or low-level design. NFRs continue to be majorly opaque in RFPs. And the problem

Even though NFRs are extremely challenging and expensive to manage, the rising level of software complexity and industry competition have made it clear that NFRs must be considered as a crucial step in the software development process starting from response to RFP. The authors in their paper [13] discuss issues and shortcomings First, since the requirements analysis and specification levels, or the first levels of the development chain, NFR are treated informally. Second, given the wide variety of NFRs, it appears challenging to identify a single approach to handle them all. Finally, there hasn't been much research on NFRs as first-class requirements in the development process. Researchers must (i) extend and relax formal methods to support the majority of NFRs, (ii) model and analyze functional requirements and NFRs separately, and (iii) offer methodologies guidance throughout the entire development process if they are to close these gaps.

The authors in their paper [14], discuss the model of TANC for creating better traceability in software development, especially in Agile. While creating a trace is important for NFRs to bring more transparency but the authors do not go beyond the traceability and more importantly it is a framework which can be applied in Analysis/Design cycle for a Software Development only.

The current paper proposed a novel complexity scoring model in a step to bridge the gap, by assuming that NFRs in RFP are opaque and introducing a formal method to analyze the NFRs in the proposal stage.

The authors of the paper [15] elaborates other quality models. McCall's Quality Model which bridges the gap between Users and developers especially on quality factors which are more technical in nature. The Boehm's quality model is based on predefined metrics which are technical and in-depth knowledge is required of the system. Dromey's model is meant for products and is primarily applicable at implementation phase. Whereas FURPS is much simpler and technologically agnostic which can be used across any phase. And in the proposed model, the users are going to be sales or pre-sales teams who are not technical and would need a simple model for usage.

"Functionality, Usability, Reliability, Performance, and Supportability of Computer Systems Software", or "FURPS", is an acronym that is now appropriate for classifying a wide range of related systems and services [16]. Since it was first made public, the FURPS model, which was initially created at Hewlett-Packard, has been widely used to categorize the functional and "non-functional requirements" of systems [17].

When evaluating a product's efficiency, the FURPS quality model only takes the customer's perspective into account [18].

Majority of practitioners used the five quality dimensions (FURPS) of Functionality, Readability, Efficiency, Security, and Reliability to evaluate the quality of code snippets [19]. Really explaining the importance of assessing NFRs using the FURPS quality model.

The handling of "non-functional requirements" (NFRs), which would include "FURPS" requirements like performance, reliability, usability, and supportability, will be covered in this paper. Use case sources distinguish between NFRs that apply to a specific use case and those that apply to the entire environment, as we have noted. According to requirement sources, we advise including a non-functional requirement in a use case section designated for that purpose if a requirement is non-functional but still applies at the requirement or user story level [20].

In actual use, the various FURPS factors are further analyzed and then applied against a priority scale using the MoSCoW model.

MoSCoW, an acronym for "Must have, should have, could have, and won't have" (MSCW), is now used universally for system requirement prioritization. It was first employed within the "Dynamic System Development Method" to consider the priorities from a relatively simple perspective [21].

Within the RFP, every requirement or scope will have success criteria and associated extended scope. Each step in the main success scenario and its associated extension may be given a priority. This is crucial for ranking individual user story acceptance criteria in order of importance. We advise against ranking ordering [20]. Instead, we advise using MoSCoW prioritization. Make sure that each step in the main success scenario is given a priority that is higher than or equal to that of its extended requirement.

Business value is ranked as the most crucial factor in prioritization, but many organizations also take complexity, cost, and other factors (such as importance) into account. The respondents to the survey by the authors [22] appear to rely on simple techniques like MoSCoW when it comes to requirement prioritization.

By combining the "FURPS and MoSCoW models" into a single model, the analyst can evaluate and adjust the requirements considering these priorities and is given a great tool that is clear to and well-understood by most stakeholders. With this straightforward format, stakeholders can easily understand the requirements-priority matrix and participate more effectively in the prioritization process and the creation of the resulting backlog.

The financial implications across NFRs are critical to stand up the software/product and more importantly it continues to hold importance postproduction rollout.

So, it's important to really extract all explicit and opaque NFRs and use the MoSCoW model to grade them across the FURPS bucket and evaluate the overall complexity in developing the same.

III. Significance of the study

NFRs have always been treated informally from the Proposal stage to requirements analysis and specification levels up through the first levels of the development chain. NFRs are so diverse, it always seems difficult to define a single approach to handle them all. There aren't many studies that have looked at NFRs as first-class requirements for the development process. Professionals as well as Researchers face a variety of difficulties in bridging these gaps: To support the majority of NFRs, formal methods must be extended and loosened; functional requirements and NFRs should be modeled and analyzed separately; and guidance on methodologies including scoring should be provided starting RFP stage.

This paper has attempted to bridge that gap by proposing a novel model which provides step by step guidance on NFR elicitation, quality & priority scoring and then evaluating an overall Complexity Score using "FURPS and MoSCoW model" together. This will help organizations to assess the complexity of opaque NFRs and calibrate Cost of Delivery and Schedule at the proposal stage itself.

IV. Methodology

In this paper, we have used requirements from project scope of 2 real time RFPs. Created a step-by-step process of Project scope extraction, NFR correlation, ID assignment, FURPS Quality & MoSCoW Priority Score assignment and finally deducing the Complexity Score.

For data we have used the below 2 RFPs which are available over the internet.

- 1) "Request for Proposal" (RFP) by "IDBI For Developing a Digital Bank Application & Branch Digitization Services" [23]
- 2) "Request for Proposal" (RFP) by "Citizen Credit Bank for Web based Core banking System across their branches" [24]
 - A Extraction of Project Scope Requirement from RFP [24], [25]
 - B Correlating Non-Functional Requirement to Project Scope of RFP [24], [25]

We deduced Non-functional requirements from the scope. The NFRs were generally non-explicit at the RFP stage.

RFP	Scope/Requirement	Section/Page
Core Banking Solution & Implementation on ASP Basis or On-Premise Model [24]	Application required two-factor authentications for users.	Project Scope/Pg52
	It should also support accounting and MIS needs for the Bank.	Project Scope/Pg52
	The application must include adequate inbuilt security as well as sufficient audit logs to maintain historical footprint of financial & non-financial transactions performed by various users.	Project Scope/Pg53
	To implement the CBS solution, the Bank is looking at an solution which includes application software & customization, parameterization and implementation, DC site, DR site and Near DR (if required)	Project Scope/Pg50
	The application needs to be user friendly and flexible. Moreover, it should support Business Process Reengineering concepts.	Project Scope/Pg53
	The application should be able to integrate with the existing 3rd party applications as per the bank requirement.	Project Scope/Pg53

Table 1: Excerpts from RFP [24] (Refer Section and Page in the RFP)

RFP	Scope/Requirement	Non-Functional Requirement
Core Banking Solution & Implementation on ASP Basis or On-Premise Model [24]	Application required two factor authentications for users	Supporting auditability for any triage of fraud login
	It should also support accounting and MIS needs for the Bank	MIS reports must be available within one working week of the date.
	The application must include adequate inbuilt security as well as sufficient audit logs to maintain historical footprint of financial & non-financial transactions performed by various users	Duration for keeping the Historical Audit Logs
	To implement the CBS solution the Bank is looking at a solution which includes application software & customization parameterization and implementation DC site, DR site and Near DR (if required)	Time taken to bring up the DR site in an event of disaster
	The application needs to be user friendly and flexible Moreover, it should support Business Process Reengineering concepts	The app should be responsive to devices and screen resizes
	The application should be able to integrate with the existing 3rd party applications as per the bank requirement	API response time for real time integrations

Table 2: Deduced NFRs from Scope (RFP [24])

C Extraction of Project Scope Requirement from RFP [23], [25]

RFP	Scope/Requirement	Section/Page
RFP for Developing a Digital Bank Application & Branch Digitization Services [23]	While developing the interfaces, the Bidder must ensure and incorporate all necessary security and control features within the application as per PCIDSS PADSS standards and Digital Payment Security Controls of RBI to maintain confidentiality integrity, and availability of the data	Scope of Work/Pg22
	The application shall be in compliance with the latest Open Web Application Security Project (OWASP) Top 10 Mobile Vulnerability report guidelines which primarily covers Improper Platform Usage, Insecure Data Storage, Insecure, Communication Insecure Authentication Insufficient Cryptography Insecure Authorization Client Code Quality, Code Tampering Reverse Engineering Extraneous Functionality etc.	Scope of Work/Pg22
	The "Digital Bank Application will be a single stop for solution for new customers/existing customer including onboarding sales of banking products (assets & liabilities) through STP (or) near STP driven by Bank's operating model banking services (financial & non financial) for different segment of customers	Scope of Work/Pg23
	Beneficiary management module should be made available and respective API should be made available for consumption at other channels	Scope of Work/Pg24
	The account opening portal is aimed to replace the existing account opening process for savings and term deposit accounts It focuses on creating a seamless workflow for the customer and employee to create the account The new process will reduce data inputs from the customer due to the use of APIs, improve use experience and reduce the TAT for account creation	Scope of Work/Pg25
	Officers shall be able to login into the portal and capture all the loan visit details and submit the same instantly from the location	Scope of Work/Pg25

Table 3: Excerpts from RFP [23]

We deduced Non-functional requirements from the scope. The NFRs were generally non-explicit at the RFP stage.

D Correlating Non-Functional Requirement to Project Scope of RFP [23], [25]

E ID Assignment for all Non-Functional Requirements across both the RFPs.

Assigning IDs for each NFR.

F Score assignment for Non-Functional Requirements

FURPS Quality Score Labels.

MoSCoW Priority Score Labels.

Additionally, different vendors also use grading methods by assigning scores (weightage) to understand overall complexity from NFR which are generally opaque in RFP. The scores are based on prior experience and domain of the RFP.

The RFPs used for data are from banking industry and hence reliability is most important followed by other attributes,

Henceforth in the paper, Quality Score will be referenced as QScore, and Priority Score will be referenced as PScore. All the NFRs are bucketed in the FURPS quality and MoSCoW priority matrix.

RFP	Scope/Requirement	Non-Functional Requirement
RFP for Developing Digital a Bank Application & Branch Digitization [23]	While developing the interfaces, the Bidder must ensure and incorporate all necessary security and control features within the application as per PCIDSS/PASS standards and Digital Payment Security Controls of RBI to maintain confidentiality, integrity, and availability of the data	Encrypting Credit Card Number to align with PCI standards
	The application shall be in compliance with the latest Open Web Application Security Project (OWASP) Top 10 Mobile Vulnerability report guidelines which primarily covers Improper Platform Usage, Insecure Data Storage, Insecure Communication, Insecure Authentication, Insufficient Cryptography, Insecure Authorization, Client Code Quality, Code Tampering, Reverse Engineering, Extraneous Functionality etc.	Scrambling Customer PII data while storing
	The 'DigitalBank Application' will be a single stop solution for new customers/existing customers including onboarding, sales of banking products (assets & liabilities) through STP (or near STP) driven by Bank's operating model, banking services (financial & non-financial) for different segment of customers	Service response time within agreed thresh-hold (in milliseconds). Both Synchronous and Asynchronous service calls.
	Beneficiary management module should be made available and respective API should be made available for consumption at other channels	Public APIs with access to configured Channels supporting interoperability

Table 4: Deduced NFRs from Scope (RFP [23])

NFR-ID	Non-Functional Requirement
A11	Supporting auditability for any triage of fraud login
B13	MIS reports must be available within one working week of date
C12	Duration for keeping the Historical Audit Logs
D11	Time taken to bring up the DR site in an event of disaster
E14	The app should be responsive to devices and screen sizes
F15	API response time for real time integrations
G12	Encrypting Credit Card Number to align with PCI standards
H16	Scrambling Customer PII data while storing (service response time with agreed threshold)
I11	Public APIs with access to configured Channels Asynchronous service calls
J17	Maintaining Interoperability
K13	Supporting Data Integrity across workflows
L15	Resizing(pixel) the captured Images of documents for efficient management

Table 5: Assigned IDs to all NFRs from Scope (RFP [23], RFP [24])

V. Result analysis

When both FURPS and MoSCoW models are used together, it provides an easy way for vendors responding to RFP to assess the scope of work from effort and cost perspective by using the novel Complexity Scoring model.

Calculating Quality & Priority Score (QP Score) for all NFRs in scope.

$$\text{QP Score (NFRID = A11)} = \text{Q Score (X = "R")} + \text{P Score (Y = "M")} = 9, \quad (1)$$

$$\text{QP Score (NFRID = B13)} = \text{Q Score (X = "F")} + \text{P Score (Y = "M")} = 6, \quad (2)$$

$$\text{QP Score (NFRID = C12)} = \text{Q Score (X = "S")} + \text{P Score (Y = "S")} = 7, \quad (3)$$

$$\text{QP Score (NFRID = D11)} = \text{Q Score (X = "R")} + \text{P Score (Y = "S")} = 8, \quad (4)$$

$$\text{QP Score (NFRID = E14)} = \text{Q Score (X = "S")} + \text{P Score (Y = "S")} = 7, \quad (5)$$

$$\text{QP Score (NFRID = F15)} = \text{Q Score (X = "P")} + \text{P Score (Y = "M")} = 7, \quad (6)$$

$$\text{QP Score (NFRID = G12)} = \text{Q Score (X = "R")} + \text{P Score (Y = "M")} = 9, \quad (7)$$

$$\text{QP Score (NFRID = H16)} = \text{Q Score (X = "R")} + \text{P Score (Y = "S")} = 8, \quad (8)$$

$$\text{QP Score (NFRID = I11)} = \text{Q Score (X = "P")} + \text{P Score (Y = "S")} = 6, \quad (9)$$

$$\text{QP Score (NFRID = J17)} = \text{Q Score (X = "S")} + \text{P Score (Y = "S")} = 7, \quad (10)$$

FURPS Quality Score	Quality Label
1	Optional, Program can go live without this, and it can be covered in future releases.
2	Good for base functions, aesthetics, and consistency.
3	Important for basic modules including integrations with external systems.
4	Very Important for majority of modules including customer experience.
5	Super Important for overall success of the program

Table 6: FURPS quality score labels

MoSCoW Priority Score	Priority Label
1	Can be de-prioritized for the current release and can be covered in future releases.
2	These are low-cost items for tweaking.
3	These are required over long runs within program.
4	Must have and Program cannot go live without them.

Table 7: MoSCoW priority score labels

$$\text{QP Score (NFRID = K13)} = \text{Q Score (X = "R")} + \text{P Score (Y = "S")} = 8, \quad (11)$$

$$\text{QP Score (NFRID = L15)} = \text{Q Score (X = "U")} + \text{P Score (Y = "S")} = 5. \quad (12)$$

Calculating the Maximum QP Score Possible (refer Table 8 for scoring)

$$\text{Max QP Score} = \sum_{n=1}^{n=12} \text{Max Score(NFRID)} = 108. \quad (13)$$

Calculating the Minimum QP Score Possible (refer Table 8 for scoring)

$$\text{Min QP Score} = \sum_{n=1}^{n=12} \text{Min Score (NFRID)} = 36. \quad (14)$$

Calculating the Actual QP Score based on proposed model

$$\text{Evaluated QP Score} = \sum_{n=1}^{n=12} \text{Score (NFRID)} = 87. \quad (15)$$

With 12 NFRs in scope, the maximum "Quality & Priority" (QP) score could be 108 and the minimum could be 36. Whereas the Evaluated QP Score is 87.

Complexity Score ranges from 0 – 1 with 1 being the most complex group of NFRs in scope and 0 being items which are neither complex nor required for the current program release. This is derived based on an overall combination of quality and priority score.

We evaluated different methodologies to assess the most accurate way to derive the Complexity Score.

A Scatter Plot - Plotted a scatter plot of the evaluated scores and their corresponding complexities, and then fit a curve or line to the data to estimate the relationship between the two variables. Once you have the trendline or curve, we can use it to estimate the complexity for any given evaluated score. Simply find the point on the trendline or curve that corresponds to the evaluated score and read off the corresponding complexity. In this method, the evaluated QP Score is 87, we found the corresponding complexity which comes to approximately 0.72. The graphical estimate may not be as accurate as the mathematical estimate from the regression equation, but it can be a useful way to get a quick estimate of the complexity based on the evaluated score. This does appear to have a linear relationship.

B Linear Regression - Using a linear regression to estimate the relationship between the two variables (Complexity Score and QP Score).

To calculate the complexity using linear regression, we first need to build a linear model that relates the score to the complexity.

Let's assume that the complexity is a linear function of the score, such that:

$$\text{Complexity Score} = m * \text{QP Score} + b, \quad (16)$$

where m is the slope of regression coefficient, b is the intercept, and score is the evaluated QP score.

FURPS Model	Quality Score	MoSCoW Model	Priority Score
F - Functional	2	M - Must	4
U - Usability	2	S - Should	3
R - Reliability	5	C - Could	2
P - Performance	3	W - Won't	1
S - Supportability	4		

Table 8: Scoring - FURPS quality and MoSCoW priority

FURPS Model	MoSCoW Model			
	M - Must	S - Should	C - Could	W - Won't
F - Functional	B13			
U - Usability		L15		
R - Reliability	A11, G12	D11, H16, K13		
P - Performance	F15	I11		
S - Supportability		C12, E14, J17		

Table 9: Mapping NFRs in FURPS quality and MoSCoW priority matrix

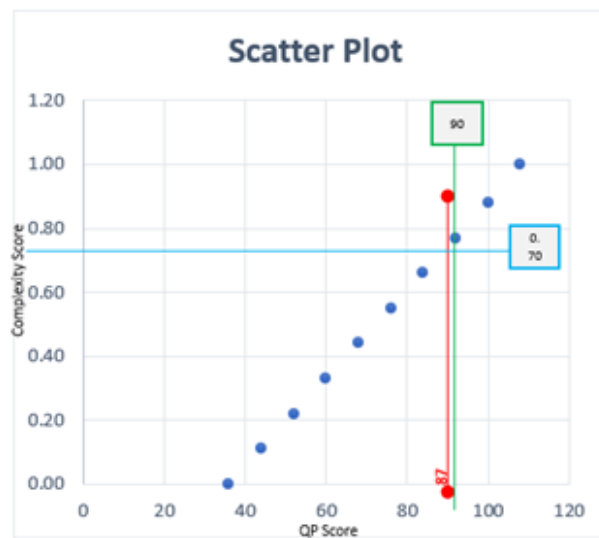


Figure 4: Scatter plot for complexity scoring

To find the values of m and b , we need to fit a linear regression model to a set of data that includes the maximum and minimum scores and their corresponding complexities. We can then use this model to predict the complexity for any given score.

For example, let's say we have the following data:

Max QP score = 108, Max Complexity Score = 1.0, Min QP score = 36, Max Complexity Score = 0.

Using these values, we can calculate the slope (m) and intercept (b) of the linear model as follows:

$$m = \frac{(\text{complexity}_{\max} - \text{complexity}_{\min})}{(\text{score}_{\max} - \text{score}_{\min})} = \frac{1.0 - 0}{108 - 36} = 0.012, \quad (17)$$

$$b = \text{complexity}_{\min} * \text{score}_{\min} = 0 - 0.012 * 36 = -0.432, \quad (18)$$

$$\text{Complexity} = m * \text{score} + b = 0.012 * 87 - 0.432 = 0.732. \quad (19)$$

Now we can use this linear model to predict the complexity for any given score. For example, if the evaluated QP score is 87, we can calculate the complexity. The complexity corresponding to the evaluated score of 87 is 0.732.

C Mathematical Formula

To calculate the complexity score given an evaluated score, we will use the following formula:

$$\text{Complexity Score} = (\text{evaluated score} - \text{lowest score}) / \text{median score}. \quad (20)$$

$$\text{Complexity Score} = \frac{87 - 36}{72} = 0.7083. \quad (21)$$

Therefore, the Complexity Score would be 0.7083.

We have used 3 different comparative methods (Scatter Plot, Linear Regression and Mathematical Formula) to calculate the complexity score. The mathematical formula has been found to be the most accurate and simple for usage. And moreover, it does not depend on large sample size of NFRs whereas Scatter Plot and Linear Regression depend on large NFR sample size for accurate Complexity scoring. So, our recommendation is to use the mathematical formula for Complexity Scoring Model for NFRs.

VI. Conclusion

The growing complexity of software and the fierce competition in the software industry and housing industry have made it necessary to consider NFRs as an integral part of the software development process, even though dealing with NFRs is very difficult and expensive. Numerous studies have examined them through the various stages starting from sales proposal to the software development life cycle for this purpose. This research paper has successfully demonstrated a working model for evaluation of NFRs during proposal stage using FURPS quality and MoSCoW priority and recommending a model to calculate Complexity Score. In the paper we attempted to evaluate the Complexity Score using Scatter Plot, Linear Regression and Mathematical Formula. The mathematical formula proved to be accurate, simple to use and more importantly does not depend on large sample size for accuracy as opposed to Scatter Plot and Linear Regression methods. The complexity score will help the service provider to assess the labor cost along with Software/hardware costs and project schedule. This ability also allows to trace design choices along with cost implications back clearly and succinctly to their original NFR sources – in this case RFP.

VII. Future Scope

This paper serves as an initial building block in an ever-changing ecosystem in the area of software industry and housing industry. With organizations moving towards complete agile, the RFP itself may not be a big bang document of project scope and business outcome. Future work can build on creating complexity models in Agile development eco systems.

With AI/ML findings more and more usage among quality modeling. It might be interesting to see if researchers would want to pursue creation of a framework which uses AI/ML models to extract opaque Non-Functional requirements from RFP and assign complexity scores which can translate to effort and cost.

References

- [1] Eleven Technical Requirements to Add to Your Next Software Product RFP <https://computas.com/blog/eleven-technical-requirements-to-add-to-your-next-software-product>
- [2] Project Management Statistics: Trends and Common Mistakes in 2022 <https://teamstage.io/project-management-statistics/>
- [3] Request for Proposal <https://slidemodel.com/templates/request-for-proposal-powerpoint-template/>
- [4] Non-Functional Requirements Quick Guide for the Business Analyst in 2023 <https://businessanalystmentor.com/non-functional-requirements/>
- [5] Moscow prioritization <https://www.agilebusiness.org/dsdm-project-framework/moscow-prioritisation.html>
- [6] Chu LY, Rong Y, Zheng H. The strategic benefit of request for proposal/quotation. *Operations Research*. 2022 May;70(3):1410-27.
- [7] Chung L, Nixon BA. Dealing with non-functional requirements: three experimental studies of a process-oriented approach. In *Proceedings of the 17th International Conference on Software Engineering 1995 Apr 23 (pp. 25-37)*.
- [8] Kassab M. Formal and quantitative approach to non-functional requirements modeling and assessment in software engineering (Doctoral dissertation, Concordia University).
- [9] Rahy S, Bass JM. Managing non-functional requirements in agile software development. *IET Software*. 2022 Feb;16(1):60-72.
- [10] Silva A, Pinheiro PR, Albuquerque A, Barroso J. Evaluation of an approach to define elicitation guides of non-functional requirements. *IET Software*. 2017 Oct;11(5):221-8.
- [11] Keong SL, Embi ZC. A systematic review on non-functional requirements documentation in agile methodology. *Journal of Informatics and Web Engineering*. 2022 Sep 15;1(2):19-29.
- [12] Saito Y, Monden A, Matsumoto K. Evaluation of non functional requirements in a request for proposal (rfp). In *2012 Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement 2012 Oct 17 (pp. 106-111)*. IEEE.
- [13] Matoussi A, Laleau R. A survey of non-functional requirements in software development process. *LACL*.
- [14] Arbain AF, Jawawi DN, Ghani I, Kadir WM. Non-functional requirement traceability process model for agile software development. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*. 2017 Oct 20;9(3-5):203-11.
- [15] Al-Obthani FS. Towards customized smart government quality model. *International Journal of Software Engineering & Applications (IJSEA)*. 2018 Mar;9(2).
- [16] Zimmer B. Software quality and productivity analysis at Hewlett-Packard. In *[1989] Proceedings of the Thirteenth Annual International Computer Software & Applications Conference 1989 Sep 20 (pp. 628-632)*. IEEE.
- [17] Grady RB. Measuring and managing software maintenance. *IEEE software*. 1987 Sep;4(5):35-45.
- [18] Ronchieri E, Canaparo M. Assessing the impact of software quality models in healthcare software systems. *Health Systems*. 2023 Jan 2;12(1):85-97.
- [19] Ndukwe IG, Licorish SA, Tahir A, MacDonell SG. How have views on Software Quality differed over time? Research and practice viewpoints. *Journal of Systems and Software*. 2023 Jan 1;195:111524.
- [20] Spurrier G, Topi H. Synergistically employing user stories and use cases in the practice and teaching of systems analysis and design. *Communications of the AIS*. 2022;561.
- [21] Achimugu P, Selamat A, Ibrahim R, Mahrin MN. A systematic literature review of software requirements prioritization research. *Information and Software Technology*. 2014 Jun 1;56(6):568-85.



- [22] Borhan NH, Zulzalil H, Hassan SA, Ali M. Requirements Prioritization in Agile Projects: From Experts' Perspectives. *Journal of Theoretical and Applied Information Technology*. 2022 Oct 15;100(20).
- [23] Request for Proposal (RFP) For Developing a Digital Bank Application & Branch Digitization Services [Downloaded 30/1/23] <https://www.idbibank.in/notices-Adv/pdf/RFP-Digital-Bank-and-Branch-Digitisation.pdf>
- [24] Citizen Credit Bank RFP for Web based Core banking System [Downloaded 30/1/23] <https://citizencreditbank.com/mybank/downloads-fdr/tenders/RFP-FOR-CBS-CitizencreditBank.pdf>
- [25] The extracted data along with the RFP is stored in GitHub <https://github.com/nadeemakhin/NFREvaluation.git>

...