

**Atal Bihari Vajpayee  
Indian Institute of Information Technology and  
Management, Gwalior**



**Coping with Quality Requirements in  
Large, Contract-Based Projects**

**Amit Raman**  
**ABV- Indian Institute of Information**  
**Technology**  
**Gwalior, India**  
[bcs\\_2019007@iiitm.ac.in](mailto:bcs_2019007@iiitm.ac.in)

---

**Kislay Singh**  
**ABV- Indian Institute of Information**  
**Technology**  
**Gwalior, India**  
[bcs\\_2019030@iiitm.ac.in](mailto:bcs_2019030@iiitm.ac.in)

---

**Dev Krishna Sadana**  
**ABV- Indian Institute of Information**  
**Technology**  
**Gwalior, India**  
[bcs\\_2019018@iiitm.ac.in](mailto:bcs_2019018@iiitm.ac.in)

**Sachin Toppa**  
**ABV- Indian Institute of Information**  
**Technology**  
**Gwalior, India**  
[bcs\\_2019050@iiitm.ac.in](mailto:bcs_2019050@iiitm.ac.in)

# Table of Contents:

ABSTRACT	3
1. Problem Stated	4
2. Approaches to Solve Issues	4
3. Results	6
4. Critical Analysis	8
5. Improvements	9
6. References	9
7. Individual Contributions	9

## ABSTRACT

This research paper is basically based on a study to determine a possible range of views on how architects cope with QRs. This report contains the view point and opinions of different software architects from all over the world on how they cope with QRs in their contract based project.

Requirements engineering (RE) considers quality requirements (QRs) and the software architect's perspective important. There have been a number of studies of how the software architects' perceive the QRs. The consensus of these studies was that software architects' and RE specialists' perspectives on QRs differ. The software architects' experiences came mostly from small and mid-sized projects. The research after interviewing 20 participants who belonged to different contract based projects complements these studies with findings from large, contract-based projects. As in those five studies, we found that software architects feel it's important to gain a deep understanding of the QRs and use this to deliver good architecture design. Research study revealed that all the software architects were actively involved in QRs refinement.

However, regarding most aspects studied, we found significant differences due to

1. the different architects' profiles,
2. the project organizations' sizes,
3. possible incentives in play, and
4. the fact that large, contract-based projects are managed differently and take place in more regulated and standardized contexts than smaller projects.

# 1. Problem Stated

The Research paper discussed a method to deal with various methods to cope up with quality requirements in large scale, contract based systems delivering projects.

In a contract to deliver a large software system, the client and vendor agree to undertake, or refrain from undertaking, certain actions in the course of delivering the system. The contract regulates the client-vendor relationship by defining each party's rights, liability, and expectations. It addresses a host of related issues such as system quality, timelines, delivery cost and effort, penalties for mistakes or missed deadlines, and service-level agreements (SLAs).

These issues are settled in a contract often creates various incentives for the participating parties. Compared to inhouse development, in contract-based projects the pressure to align incentives across parties is stronger, and the need to align the parties' understanding of system quality attributes is much greater. So, monitoring the project artifacts' quality is more complex.

## 2. Approaches to Solve Issues

### 1. Defining their role

- Architects work as a mediator such as they serve as a bridge to non-technical project stakeholders.
- Other architects regarded themselves as review gatekeepers as they served most of their time reviewing QRs, giving feedback, and evaluating contract compliance.

### 2. Use Standard To Ease Communication

Using **ISO** standards to check what ISO-compliant Quality Manual says to follow rules and avoid mistakes.

Mainly there are two streams **ISO** standards to keep common Interpretation of ORs terminology:

- ➔ **Managements Systems**
- ➔ **Technical Standards**

However there is no common rule to follow which stream follows which QR terminology so participants themselves decide which to use by cross-checking.

### 3. Discovering QRs through Refinement

- ❖ In a very large contract-based project we studied that there should be a Refinement phase through which QR must go through to improve little improvements and validate final QR.
- ❖ Most of the Participants use **Checklists** to refine QRs.
- ❖ And some of them used **ISO** standards in the **SQUARE**(System and Software Quality Requirements and Evaluation).
- ❖ Some of them used architecture frame-works that were specific to their company, business sector, or the client organization.
- ❖ Rest others used Creative Techniques such as video Streaming and game design in which users played a scheduling game by using an early prototype of the model to detect the performances and requirements.

### 4. Predefined QRs Templates

Most of the participants use predefined templates for documentation based on

- ISO Standard
- Vendor-specific standards
- Quality Function Deployment(QFD)

Some of the participants used plain text to define QRs, plus information on the end user to demonstrate that the system met the QRs.

We must update and ensure our commitments to quality levels are realistic as per contract-based.

### 5. Prioritize of QRs

Most of the architects felt that the project's business cases were the key driver in prioritising Quality requirements. There are four prioritising criteria of tradeoffs.

- ➔ *Cost and Benefits*
- ➔ *Perceived Risks*
- ➔ *Affordability*
- ➔ *Willingness to Pay*

### 6. Quantifying QRs

- There is no common method to quantifying QRs.
- Architects either use a size-estimation standard such as (International Function Point Users Group's non function- -nal-assessment method) , This approach ensures that a project explicitly accounts for all QRs implementation tasks .

- Some use a specific type of QR according to requirements such as security and scalability, etc. This approach allows for deeper analysis of single quality attributes.

## **7. Walk-through to Validate QRs**

Validation ensured that QRs are aligned with client interests and needs. Most of the architects consider QR validation their own responsibility and regularly perform requirements walkthroughs with clients and business analysts.

This was a part of the client expectation management process that the manager established.

Some use international architecture standards and QFD to demonstrate the related strength between a QR definition and operationalisation.

## **8. Tradeoffs in QRs**

The business case is the most important factor while resolving conflicting QRs.

Different people used different methods to justify tradeoffs. They used business cases, budget allocation figures, QFD or the Six thinking hats' method, etc.

## **9. Contract is Resource**

a contract which ensures that the system includes "the right things" and "those they needed in the first place, and that we are billing them for".

Regardless of documentation Participants have the review and update after achieving each milestone in the contract.

Participants actively used the contract to achieve different, yet complementary goals.

Knowledge of clients in system development also helps to understand contracts in a better way. Participants usually take the help of

# **3. Results**

## **1. Necessity of an architect**

The architect plays an important role to bridge the gap between Stakeholders and Developers to make sure that the business analysts and stakeholders know how their technical demands affect the business and are aware of all other possible choices. They also need to review the QRs and give appropriate feedback.

## **2. Following the standards set by International organizations**

Participants follow some standard set by ISO and other organizations to check whether they follow correct rules and regulations and facts. There is some confusion between which standard to use for one particular stream though having knowledge of all the

standards which are management systems standards and technical standards helps to choose between them by cross-references. This helps everyone in the team to understand the terms used and diminishes the difficulty in understanding between architects and business analysts.

### **3. QR refinement procedure before finalising QR**

Refinement Procedure is necessary to detect bugs and to add additional features through which participants can make QRs precise and accurate.

There are many methods of QR refinements such as Checklists(As a basis for the checklists, eight used ISO standards in the SQuaRE- System and Software Quality Requirements and Evaluation series).

### **4. Results of Using Predefined Templates**

By using predefined templates we can ensure that we are not going out of the league according to International standards . So having predefined templates can also help us to make QRs in current standard business models.

### **5. Advantages of Prioritizing QRs**

Prioritizing QRs helps participants in many ways. Priorities of things get done first before others are decided in the procedure so that stakeholders can be aware of tradeoffs. Priorities can also be dependent on the demand of clients. Priorities mainly decided on Cost and Benefits , Risks, Affordability and willingness to pay the client (if adding some extra features or implementing better alternatives technologies over others). It must be very clear about “must-have things” first before others.

### **6. Advantages of Quantifying QRs**

In today's world, there are endless possibilities and alternatives of adding features to one particular feature to the product. But it must be very clear about what and how much we want to use according to cost and benefits, Risks, affordability, and client willingness to accept the changes.

So Therefore Quantifying QRs can help to avoid getting into these difficulties. However we must also not forget about must have things which have been mentioned in Prioritizing QRs.

### **7. Walk through to validate QRs**



Validation is must while making QRs cause we all are bound to make mistakes. So validation after each updation and addition ensures that we follow the Contract-Based SLA(service level agreement). There are some International Standards set for QRs and some uses QFD.

## **8. Advantages of Tradeoffs in QRs**

Tradeoffs between client and business analysts lead to satisfy both the parties. Trade Offs happen because of conflicting opinions so there must be discussion with stakeholders and vendors to avoid issues and in that they should also make sure that all the features and requirements can be implementable by developers with existing technologies considering cost and prices.

## **9. Advantages of making QRs according to Contract**

We should be realistic about our commitments while contract-signing and service level agreements. We should not be doing things which are not in contract.

# **4. Critical Analysis**

Large and contract based projects require joint requirements and architecture design which ensures that architects become essential and necessary to the QRs process. The participants in our study give away four clear points for this copying strategy.

- First, the architects should engage in dealing with the QRs with the same rigour required by functional requirements and architecture design.
- Second, it is the architect's responsibility to reduce and diminish the risks and possibilities of contract disputes. The architect should take the leadership in ensuring that QRs tradeoffs align with the contract.
- Third, the architect's domain knowledge gives an edge in proactively clarifying the QRs.
- Finally, the architect is to be socially positioned as a bridge and gatekeeper for all the discussion and conversation on QRs.

Based on the day to day coping strategies of several architects on the large scale, contract based projects, an insight of this approach is that the contract regulates the client– vendor relationship by denying each party's rights, liability, and expectations. It addresses a host of related issues such as system quality, timelines, delivery cost and effort, penalties for mistakes or missed deadlines, and service- level agreements (SLAs). So applying those methods can avoid these issues in the best way possible

## 5. Improvements

The following improvements can be made in the approach discussed in the paper :-

- We can apply some empirical formulas to finalize the size of the project. In which we can also consider efforts, man power, time, cost, all the factors to estimate in a better way.
- We can also use some modern technologies to make QRs which are available in the market. We can also make a team which can demonstrate
- virtually to the client how the final project would be so that the client can understand and make his/her expectations accordingly to avoid later conflicts.
- Some software development models according to certain standards can be followed to save time

## 6. References

1. M. Daneva, L. Buglione, A. Herrmann, "Coping with Quality Requirements in Large Contract-based Projects", 2015
2. B. Berenbach et al., "Contract- Based Requirements Engineering," Proc. 3rd Int'l Workshop Requirements Eng. and Law, 2010, pp. 27-33.
3. E. de Bono, Six Thinking Hats, Little, Brown, & Co., 1985
4. M. Daneva et al., "Software Architects' Experiences of Quality Requirements: What We Know and What We Do Not Know?"

## 7. Individual Contributions

The analysis and documentation of the research paper was the result of the effort of the whole team. The team did work as a whole, but it's necessary to highlight the important contributions made by each team member. The individual contributions made by each member is mentioned below :

1. The research paper was read and a summary which includes problem statement, abstract of the paper is done by Sachin Toppa
2. The approach to solve the issues were noted by Amit Raman
3. After studying the research paper, the results and advantages of the model were listed and noted down by Kislay Singh.
4. After going through the research paper, a thorough analysis of the paper was done and possible improvements were suggested by Dev Krishna Sadana