Group 4

Sri Harsha Majeti msharsha555@gmail.com
Harshavardhan Miryala miryalaharshavardhan4@gmail.com
Sanju Prabhath Reddy reddyksanju@gmail.com
Utsav Mangal mangalutsav1@gmail.com
Dinesh Reddy dineshreddydumpa8037@gmail.com

Coping With Quality Requirements In Large,Contract-based Projects

Authors :

Maya Daneva, UT
Andrea Herrmann, H&E
Luigi Buglione, EII

Details Of the Paper

Contracts for delivering large software systems must address issues such as system quality, timelines, delivery cost and effort and service agreements. A study with software architects revealed how they coped with quality requirements in this context.

Summary Of The Paper

- Software Architect??

  Act as mediators between business analysts and clients on one side and developers on the other,and can constructively influence QRs engineering.

- Insight to Requirements Engineering

  RE refers to process of defining, documenting and maintaining requirements in software engineering.However perspectives of REs and software architects differ due to

  1. Different architects' profiles
  2. Project organizations' sizes
  3. Incentives in play
  4. More Regularisation in large projects.

- The Study

  The following are 3 vital areas concerning architects' coping strategies
  - Interaction with Stakeholders(Negotiations)
  - Engineering.(Documentation)
  - Contract Compliance(Role Of Agreements)

  After a comprehensive study on software architects in different countries, companies, projects,work-experience,the whole process has been divided 10 specific scenarios.

- Assuming a Mediator's Role
  - To bridge gaps between business clients and developers to make sure technology meets the demands.
  - But some considered as review gatekeepers serving most of their time reviewing QR's and giving feedbacks.
  - Some considered vendor relationship managers superior to them.

  Note : No participant directly dealt with user community

- Use of Standards to ease communication

  ISO-Compliant Quality Manual made their work easier.Management stream and technical standards are the common interpreters of QR terminology

- Discovering of QRs through Refinement

  The participants explicitly felt the necessity of refining QRs in the contract.Some of the techniques employed are checklists,architecture frameworks,storytelling and game-based techniques like prototypes.

- Usage of predefined QR templates in Documentation

  The participants preferred using templates based on
  1. ISO standards
  2. Vendor Specific Standards
  3. Quality Function Deployment
  4. Plain text

  Irrespective of the document format,they considered the updation and review of QRs important.

- Business case to prioritize QRs

  Prioritization criteria for making QR tradeoff based on business case are
  1. Cost and benefits

2. Perceived risks
3. Affordability
4. Willingness to pay for increasing a specific QR

Few considered the project's steering committee and others considered themselves as the key decision makers.Many of them considered this part as crucial and toughest one.

- Starting with the contract for quantifying QRs
    - Contract-based projects seem to prompt project organizations to express their QRs quantitatively.
    - But there was no common approach to quantifying QRs.
    - A common starting point was the contract's prespecified quantitative definitions.
    - Few experienced that contracts often confused QRs with design- level requirements.
    - Some of them are usage of a particular algorithm or a particular video-player.
    - They considered this issue critical because it signalled a misaligned understanding of what was really quantified and what measures were used.

- Walk-throughs to validate QRs
    - Validation ensures that QRs are aligned with the client's expectations.
    - It also confirms that the QRs are technically implementable and that the resulting architecture design satisfies the requirements defined in the contract.
    - Few considered it their own responsibility and the others deemed it as business analyst's job.

Note : No one used an automated tool for walk-throughs.

- QRs conflict resolution as the objective
    - Business Case plays the key role in resolving conflicting QRs.Eg : To justify Trade-Offs , Budget Allocation.
    - Approaches like QFD,EasyWinWin,SixThinkingHats are also adapted to resolve QRs

- Contract As a Resource
    1. Participants considered it as source of learning about QRs,to stay focused on the crucial part,to maintain control and stability.
    2. It helped to create intermediate goals which made the whole project clear and simple.Eg: Balancing Multiples Perspective Technique

- Effect Of Contract on Architects

  3 ways a contract shapes architects' coping strategies on QRs.They are :
    1. Participants Cost Consciousness.
    2. QR levels discussed with stakeholders.
    3. Pre defining priorities for crucial set of QRs.

  Some felt that contract is just a start of conversation but not a reference guide.

Note : In Summary,We wrote the common opinion of participants.Nevertheless there have been exceptions which cannot be ruled out.

Applications:

- Since the study done on the software architects is vague ,exploratory and is done to come up with a basic idea of possible strategies to cope with the QRs we can't completely generalize them.
- However we can use the strategies in situations where managers hire experienced architects who are aware of the standards and terminologies of the QRs.
- The advantages of ensuring that software architects are integral to QRs process are:
    1. Engaging the architects is instrumental in dealing with QRs with the same due diligence that the functional requirements and architecture design demand.
    2. Mitigating the risks of contract disputes becomes the architect's responsibility, who takes leadership in ensuring that QRs tradeoffs align with the contract.
    3. Leveraging the architect's domain knowledge helps pro-actively clarify QRs.
    4. Socially positioning the architect as a bridge and gatekeeper is conducive to the ongoing conversation on QRs.

Contribution of the paper :

The contribution of the research paper in software engineering field is a follows

- Coping with Quality Requirements in Large, Contract-Based Projects.
- This helps software architects to have a forward look on Service level agreements in the early stages of the development of the project.
- It also helped software architects to redefine our understanding of quality requirements (QRs) finding new optimal QR's by refining the existing ones and prioritising quantifying QR's in all stages of development.
- It also showed the ways in which a predefined contract affects architects in the development process and also that how QR's should be aligned with the SLA's.
- It also suggested the usage of standards in development to ease communication etc.

Drawbacks :

- The Author doesn't mention any drawbacks as she just discusses how it should be done, she did state at one point that Quality Requirements is often confused with Design Requirements, but I don't think that is a drawback as it can be easily cleared by specifically mentioning in the QR
- Author does mention that poor Quality Requirements Specification creates confusion and issues in the project thus wasting resources and time
- The Quality Requirements does create a lack of flexibility as often the most obvious features that are to be needed are not mentioned in the QR
- The Author stated that there isn't much communication between the management and the end user as a good thing but it isn't necessarily as it doesn't enable the engineers to get another angle in the perspective of consumer
- It also make things mechanical thus creating a little room for creativity and leaving little to imagination
- Checklist questions are usually limited to specific types of defects. Therefore, inspectors may not focus on defect types that have not been previously detected. Consequently, they may entirely miss classes of defects
- Author mentions how majority of managers prefer ISO standard QR which in case further decreases the flexibility and individual creativity

References :

1. S. Lauesen, Software Requirements: Styles and Techniques, Addison- Wesley
2. X. Song et al., “Categorizing Requirements for a Contract- Based System Integration Project,” Proc. 2012 Requirements Eng.
3. F. Buschmannetal,“Architecture Quality Revisited,”IEEE Software,vol.29,no.4.
4. E. de Bono, Six Thinking Hats, Little,Brown, & Co., 1985.

Group Contribution :

Summary and Details:Harshavardhan Miryala(15114045),SriHarsha Majeti(15114044).
Contribution in Software Field : Dinesh Reddy(15114026).
Applications : Sanju Prabhath Reddy(15114042)
Drawbacks : Utsav Mangal(15114075).