Architecture Coverage: Validating Optimum Set of Viewpoints

Sunia Naeem, Salma Imtiaz
Department of Computer Science and Software Engineering,
International Islamic University,
Islamabad, Pakistan
sunia.naeem@yahoo.com, salma.imtiaz@iiu.edu.pk

Abstract— There are various software architecture viewpoint models but none of them provides optimum coverage of software architecture domain. Software architecture coverage is the coverage of concepts that are required to effectively design and analyze software architecture. An optimum set of viewpoints can be selected from different software architecture viewpoint models that provide maximum coverage of software architecture domain than an individual architecture model. In this paper, an optimum set of viewpoints is selected by comparing five commonly used software architecture viewpoint models namely 4+1 RM-ODP, SEI, Siemens and Rational ADS via a common comparison framework. These architecture models are compared on the evaluation criteria, i.e., viewpoints, stakeholders and quality attributes. This evaluation criterion is based on IEEE Standard 1471 Recommended Practice for Software Architecture Description. The resulting optimum set is validated in industry via multiple case studies, and the results show that the optimum set of viewpoints provide greater coverage than any of the viewpoint alone.

Keywords-architecture coverage; optimum; viewpoints; stakeholders; quality attributes.

I. INTRODUCTION

Software architecture is system's high level structure and describes that system as computational components and interactions between them.

The need for documenting software architecture is its ability to communicate between stakeholders, to provide re-usable abstractions of software systems and to capture early design decisions [1]. The commonly used approach to model a complex architecture was to make use of a heavily overloaded, single model that does not adequately represent the system and difficult to understand and manage [2]. Some of the disadvantages of using this approach are unreliable notations, over emphasis of one aspect, mixing of architectural styles and overlooking of individual stakeholder concerns [1].

A great amount of work has been done to partition the architecture of the system into multiple views, where each view highlights a different perspective. This approach helps in comprehension and understandability from stakeholders' point of view. Architects also come to an understanding that to develop successful software architecture we should draw many different system structures simultaneously to handle the multi-faceted nature of architecture. It seems that software research community also have decided that the only way to design architecture is by representing system using several related models (or views) [3].

Viewpoints are used to choose which view to produce for a particular system, and what information to represent in that view. Views and viewpoints usage has various benefits, such as management of complexity, separation of concerns and improved communication with stakeholders. Viewpoint model [3] means a framework that describes the significant concerns that need to be taken care of while designing software architecture. Generally, software architecture models contain several viewpoints which define the models and concepts which can be used while dealing with the specific concern.

A research work by Nicholas May [1] surveys the different viewpoint models and highlights that existing viewpoint models need to be tailored because they do not address every concern of software architecture domain. The key purpose of this research work was to understand different software architecture models, their coverage of software architecture domain and their comparative strengths. The view point models are compared with respect to IEEE 1471-2000 Standard called the IEEE Recommended Practice for Software Architecture Description.

The author also proposed a classification of viewpoints within a common framework that allowed combining views from different viewpoint models and determining an optimum set of views with the purpose of providing maximum coverage to represent the architecture. Different vocabularies of models can be compared by common reference vocabulary. Optimum set, has the maximum coverage as compared to any individual viewpoint model.

Viewpoint models selected in this survey[1] are Kruchten's "4+1" View Model [20], Siemens Four View model [23], Software Engineering Institute (SEI) set of views [21], Rational Architecture Description Specification (ADS) [25] and ISO Reference Model of Open Distributed Processing (RM-ODP) [22]. All these five models describe software architecture from multiple perspectives. Each one of them identifies separation of concerns and specifies stakeholders. Also, these models focus on describing software architecture structures instead of describing particular notations for each of these structures.

 ISBN: 978-1-61208-367-4

Our research work focus on the extension of comparison criteria described in [1]. Section II presents the related work and compares it on a common criteria. The comparison criteria is presented in Section III where the mapping of stakeholder, viewpoints and quality attributes is presented in Section IV. The evaluation on the chosen criteria is done in Section V, and the proposed optimum set of viewpoints is presented in Section VI. The evaluation in this extended coverage criteria done in Section VII and validated in Section VIII. Conclusion and future work are discussed in Section IX.

II. RELATED WORK

This section, presents the comparison of different viewpoint models based on the ***Focus of Research, Criteria used for Comparison and Limitation of the Research Work.***

TABLE I OVERVIEW OF RELATED WORK

Paper Id	Description	Models reviewed/compared	Criteria for comparison	limitations
[3]	In this paper, the author surveyed some architecture Models and conduct a case study on the usage of software architecture documentation practices in the Telecommunications industry.	RM-ODP, US Department of Defense frameworks TAFIM, C4ISR, 4+1 view, Zachman framework	No specific criteria	Models are reviewed from literature and their details, benefits and deficiencies are based on literature review.
[4]	In this paper, viewpoint sets are applied to development of information systems and evaluated so weaknesses and strengths of every set of viewpoints is described and few general observations about their definition and use are presented.	4+1, RM-ODP, Siemens, Garland and Anthony	Industrial experience	Comparison is based only on observations of author. No common reference vocabulary is used for comparison.
[5]	This study provides analysis and comparison of six architecture frameworks categorized by major elements such as their inputs, outcomes and goals. It provides classification of architecture frameworks into Software Architecture Frameworks and Enterprise Architecture Frameworks and identifies some of their deficiencies.	Zachman Framework, 4+1, Federal Enterprise Architecture Framework (FEAF), RMODP, Department of Defense Architecture Framework (DoDAF), The Open Group Architecture Framework (TOGAF)	goals, inputs and outcomes	More focus is on classification of frameworks not on frameworks' deficiencies.
[26]	In this paper, the author provides overview of two classes of architecture frameworks Software Architecture Frameworks and Enterprise Architecture Frameworks and find some dimensions which can be helpful to understand architecture documents	Zachman Framework, The Information Framework (IFW), Integrated Architecture Framework (IAF), The Open Group Architecture Framework (TOGAF), Methodology forArchitecture Description (MAD), 4+1, Siemens	No specific criteria	More focus of comparison is on the difference between two classes of architecture framework.
[27]	In this paper, the author surveyed few architecture frameworks and compared them on the basis of methodologies and techniques they use and suggested that more architecture styles can be added to yield new architecture framework which focus on quality	Zachman Framework, 4+1, Federal Enterprise Architecture Framework (FEAF), RM-ODP, Department of Defense Architecture Framework (DoDAF), The Open Group Architecture Framework (TOGAF)	methodologies and techniques used in the framework	Focus of this comparison is to state only general advantages and disadvantages of architecture frameworks
[6]	This paper compares SEI with IEEE 1471 and show compliance of SEI with IEEE 1471.	SEI, IEEE 1471	Requirements imposed by IEEE 1471	Only compliance of one viewpoint model is considered .Compliance of other viewpoint models are missing.

III. COMPARISON FRAMEWORK ELEMENTS

To compare the software architecture viewpoint models, a common comparison framework is required. IEEE 1471-2000 Standard, called the IEEE Recommended Practice for Software Architecture Description [19] has been selected for evaluation, which consists of viewpoints, stakeholders and quality attributes, and their relationships for documenting the software architecture. IEEE 1471 considers stakeholders and their respective architectural concerns as essential elements in an architectural description. Architectural concern is a matter of importance to one or more stakeholders relating to the architecture. Another major element of ANSI/IEEE 1471 is that every architecture view in an architecture description is defined relative to an architectural viewpoint as we know architecture description is planned into multiple views and each one of them denotes the system architecture with reference to a set of related architectural concerns. So an architecture viewpoint captures the rules for analyzing and constructing a particular view and acts as a view template so it can be reused across many architectural descriptions.

A. Software Architectural Viewpoints

Viewpoints reason about quality attributes so architecture description should provide enough details or information necessary to analyze quality attributes. We have added conceptual viewpoint, in the list of viewpoints stated by Nicholas. Conceptual viewpoint [8] describes the system in form of system's major design elements and relationship between them. Conceptual viewpoint is very important because it is strongly linked with the problem domain and acts as an important means of communication when the architect interacts with domain expert. It helps in clearly defining modules in module view and impact of changes in requirements can be minimized. Viewpoints are not system specific so they are pre-defined and reusable.

B. Software Architectural Stakeholders

Stakeholder of software architecture is someone who has a vested interest in it, who implicitly or explicitly motivates the whole shape and direction of the architecture [16].

Stakeholders are consumers of software architecture description and architecture description serves as a means of communicating design decisions between stakeholders. Architecture should be communicated in a way that stakeholders use it properly for their respective use [24]. There is variety of stakeholders and their use with respect to architectural documentation varies. Nicholas`s list [1] of stakeholders can be extended to incorporate all stakeholders meant for required comparison. Our analysis will be based on the stakeholders who are consumers of software architecture's documentation. These stakeholders will make the analysis of viewpoints possible as they provide coverage of stakeholders that different viewpoints address e.g., product managers, business analysts and marketers.

C. Software Quality Attributes

Software architecture description should address stakeholder's concerns otherwise it is considered incomplete [16].

Concerns [18] are normally driven by the need for the system to exhibit a certain quality attributes rather than to provide a particular function. There is inherent need to consider quality attributes in each architecture view. Quality attributes are considered as concerns. Quality attributes can be classified into three types: Run-time, development-time and business. Nicholas`s list of concerns [1] does not include important concerns such as business quality attributes which repeatedly form a system`s architecture. Table 2 shows elements of our comparison framework comprising viewpoints, stakeholders and quality attributes.

TABLE II ELEMENTS OF COMPARISON FRAMEWORK

Viewpoints[12][8]	**Stakeholders[12][24][10][11]**	**Quality Attributes[17][12]**
Conceptual Decomposition Uses Layered Class/Generalization Process Concurrency Shared Data Client-Server Deployment Implementation Work Assignment	Architects Requirements Engineers Sub-System Architects and Designers Implementers Testers Integrators Maintainers External System Architects and Designers Managers Product Line Managers Quality Assurance Team Users Customers Project Manager Production Engineers Suppliers System Administrators Business Analysts Product Managers Marketers Support Staff	**System Run-Time** Functionality Performance Capacity/Space Availability Reliability Security Safety Usability Supportability configurability Scalability Interoperability **System Development-Time** Modifiability Reusability Testability Portability Evolvability Localizability Integrability **Business** Time to market Cost and benefit Projected lifetime of the system Targeted market Rollout schedule Integration with legacy systems

IV. MAPPING BETWEEN STAKEHOLDERS, VIEWPOINTS AND QUALITY ATTRIBUTES

Thus, evaluation done by Nicholas [1] can be extended on all three attributes which are viewpoints, stakeholders an quality attributes. In case of stakeholders and quality attributes, only those are covered that are explicitly stated by viewpoint models. We identify implicit quality attributes and stakeholders by investigating the relationship between stakeholders,viewpoints and quality attributes. Implicit stakeholders will be satisfied if all their concerns are

 ISBN: 978-1-61208-367-4

addressed by viewpoints and similarly different viewpoints address different quality attributes.

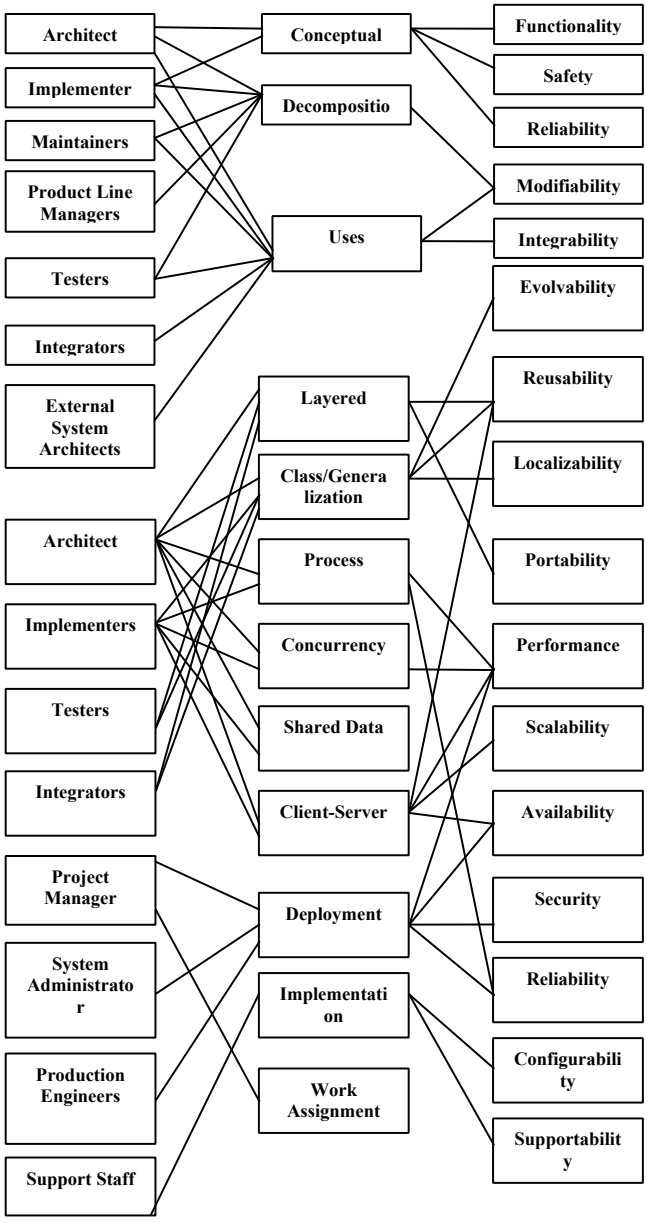

Figure 1. Mapping between Stakeholders, Viewpoints and Quality Attributes

V. MODELS EVALUATION AND COMPARISON FRAMEWORK COVERAGE

Tables 3, 4 and 5 show the coverage of viewpoints, stakeholders and quality attributes by the five software architecture viewpoint models. The coverage is found individually for each of the elements of comparison framework's concepts of stakeholders, quality attributes and viewpoints. Each viewpoint model provides different coverage of comparison framework elements. In case of viewpoints and quality attributes, SEI provides greatest coverage. As far as stakeholders are concerned, SEI and Rational ADS provide good coverage of stakeholders.

TABLE III MODELS COVERAGE OF VIEWPOINTS

Viewpoints	"4+1"	SEI	RM-ODP	Siemens	Rational ADS
Conceptual	Y	N	Y	Y	Y
Decomposition	Y	Y	Y	Y	Y
Uses	N	Y	N	N	N
Layered	Y	Y	Y	Y	Y
Class/Generalization	Y	Y	Y	N	N
Process	Y	Y	N	Y	Y
Concurrency	Y	Y	Y	Y	Y
Shared Data	N	Y	Y	N	N
Client-Server	Y	Y	Y	Y	N
Deployment	Y	Y	Y	N	Y
Implementation	N	Y	N	Y	Y
Work Assignment	N	Y	N	N	N

TABLE IV MODELS COVERAGE OF STAKEHOLDERS

Stakeholders	"4+1"	SEI	RM-ODP	Siemens	Rational ADS
Architects	Y	Y	N	Y	Y
Requirements Engineers	Y	N	N	Y	Y
Sub-System Architects and Designers	Y	Y	N	Y	Y
Implementers	Y	Y	Y	N	Y
Testers	Y	Y	Y	N	Y
Integrators	Y	Y	Y	Y	Y
Maintainers	N	Y	Y	N	N
External System Architects and Designers	N	Y	N	N	N
Managers	Y	Y	Y	Y	Y
Product Line Managers	Y	Y	N	N	N
Quality Assurance Team	N	N	N	N	Y
Users	Y	N	Y	N	Y
Customers	Y	N	Y	N	Y
Project Manager	N	Y	N	N	N
Production Engineers	Y	Y	Y	N	Y
Suppliers	N	Y	N	N	N
System Administrators	Y	Y	Y	N	N
Business Analyst	N	N	Y	N	Y
Product Manager	N	Y	Y	N	N
Marketer	Y	N	Y	Y	Y
Support Staff	N	Y	N	Y	Y

TABLE V MODELS COVERAGE OF QUALITY ATTRIBUTES

Quality Attributes	"4+1"	SEI	RM-ODP	Siemens	Rational ADS
Functionality	Y	N	Y	Y	Y
Performance	Y	Y	N	Y	N
Capacity/Space	N	Y	Y	N	Y

 ISBN: 978-1-61208-367-4

Availability	Y	Y	N	Y	Y
Reliability	Y	Y	N	Y	Y
Security	N	Y	Y	N	N
Safety	N	N	N	Y	N
Usability	N	N	N	N	Y
Supportability	N	Y	N	Y	Y
Configurability	N	Y	N	Y	Y
Scalability	Y	Y	N	N	Y
Modifiability	N	Y	Y	N	N
Reusability	Y	Y	N	N	Y
Testability	N	N	Y	N	Y
Portability	Y	Y	Y	Y	Y
Evolvability	Y	Y	Y	N	N
Localizability	Y	Y	N	N	N
Integrability	N	Y	N	N	N
Interoperability	Y	Y	N	Y	Y
Time to market	Y	Y	Y	Y	Y
Cost and benefit	Y	N	Y	Y	Y
Projected lifetime of the system	N	Y	Y	N	N
Targeted market	Y	N	Y	Y	Y
Rollout schedule	N	Y	Y	N	N
Integration with legacy systems	Y	N	N	N	Y

Y: provides Coverage
N: Does not provide coverage

VI. Optimum Set Of Viewpoints

When combining views from different viewpoint models, the biggest obstacle is dependency between views of viewpoint models. In case of" 4+1" model the views are dependent on each other, i.e., being an iterative method there is strong data flow between views. The views of the SEI and the RM-ODP model are comparatively independent. The views of Siemens model are less tightly coupled. In Rational ADS, context of lower views are provided by higher views so there is strong dependency between views.

Therefore, when combining views from different viewpoint models, we see that SEI model provides good coverage of viewpoints, stakeholders and quality attributes and also its views are independent so its three views that are module, component and connector and allocation are considered for merging. The missing stakeholders such as users, customers and business analysts which are not addressed by SEI can be incorporated by including Use Case View from Rational ADS. There is a dependency between Rational ADS views as Use Case being the highest view is not dependent on any other view. Use Case View also covers the usability concern which is not covered by SEI model. Siemens's Conceptual view is also included in optimum set as SEI model does not cover the conceptual structure and its related concern, which is functionality. Conceptual viewpoint [8] describes the system in form of system's major design elements and relationship between them. This viewpoint is very important because it is strongly linked with the problem domain.

Rational ADS Test View is added in optimum set of views to address testability Rational ADS's Test view addresses testability by enabling one to perform test realization, preparing test cases and then forming whole test procedure also satisfying the Quality Assurance Team. As we know that in Rational ADS that context of lower views are provided by higher views so we investigated and found that SEI Allocation view type overlaps well with Rational ADS Realization viewpoint which contains Implementation and Deployment View. So, context of Test View can be provided by Allocation View type of SEI model. RM-ODP views are not considered for merging because RM-ODP uses language for architecture description and not a notation so it supports communication between different systems developers and not among other stakeholders of the same system. Figure 2 shows optimum set of views from different viewpoint models.

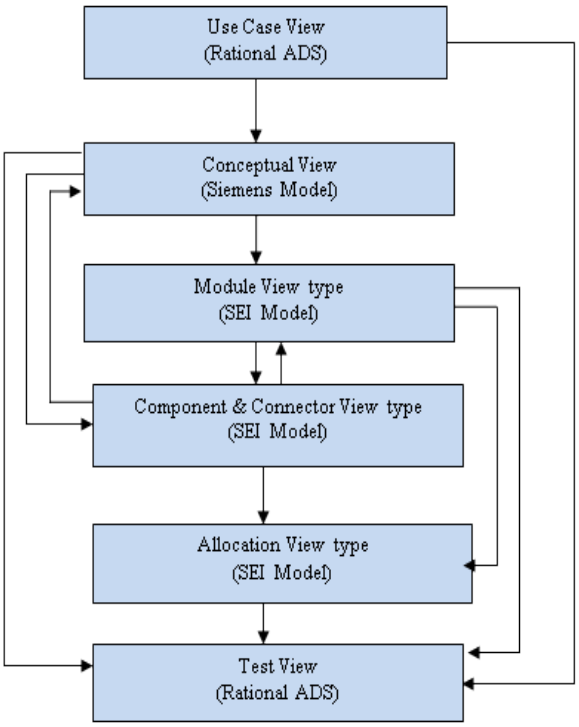

Figure 2. Optimum Set of Viewpoints

VII. Coverage of Optimum Set of Viewpoints

Tables 6 and 7 show stakeholders and quality attributes addressed by Optimum Set of Viewpoints.

 ISBN: 978-1-61208-367-4

TABLE VI STAKEHOLDERS ADDRESSED BY OPTIMUM SET OF VIEWPOINTS

Views	Stakeholders Addressed
Use Case	Users, Customers, Business Analysts
Conceptual	Architect, Implementers
Decomposition	Implementers, Maintainers, Product Line Managers, Architect, Testers
Uses	Implementers, Maintainers, Architect, Testers, Integrators, External System Architects and Designers
Generalization	Implementers, Architect, Testers, Integrators
Layered	Implementers, Architect, Testers, Integrators
Component And Connector	Implementers, Architect
Deployment	Project Manager, Testers, Integrators, Architect, System Administrator, Production Engineers
Implementation	Support Staff
Work Assignment	Project Manager
Test	Testers, Quality Assurance Team

TABLE VII QUALITY ATTRIBUTES ADDRESSED BY OPTIMUM SET OF VIEWPOINTS

Views	Quality Attributes Addressed
Use Case	Usability
Conceptual	Functionality, Safety, Reliability
Decomposition	Modifiability
Uses	Modifiability, Integrability
Generalization	Evolvability, Reusability, Localizability
Layered	Portability, Modifiability, Reusability
Pipe-and-Filter	Performance
Shared-Data	Security
Client-Server	Performance, Scalability, Availability, Reusability
Peer-to-Peer	High Availability, High Scalability
Communicating Processes	Performance, Reliability
Deployment	Performance, Reliability, Availability, Security
Implementation	Configurability, Supportability
Test	Testability

VIII. VALIDATION OF OPTIMUM SET OF VIEWPOINTS

A. Research Design

In order to validate optimum set of viewpoints, we conducted multiple-case study[7] of three software intensive projects of medium to large complexity whose architectures were built using our proposed optimum set of viewpoints either by software architect or personnel who have sound knowledge of developing software architecture by using software architecture viewpoint models. We have chosen a multiple-case study approach as multiple sources of evidence allow a better validity for the findings and used purposeful sampling. We looked for projects of those software development companies that had experience in using software architecture viewpoint models and also have experienced personnel who have sound knowledge of applying views for developing architecture of applications.

B. Data Sources

We collected data using semi-structured scripted interviews so the questions were prepared in advance and pre-defined questionnaire were used and filled in print. We could not manage to conduct face to face interviews or interview via Skype Out calls because of nature and secrecy of projects and work load.

C. Data Analysis

The purpose of filling the questionnaire was to find out optimum set of views coverage of software architecture concepts (i.e., viewpoints, stakeholders and quality attributes) that are required to efficiently design and analyze software architecture after applying it on the case projects and discuss its coverage as compared to the software architecture viewpoint model which they usually use to develop architecture of their applications. To analyze data, frequency distributions related to coverage of viewpoints, stakeholders and quality attributes by our research outcome i.e., optimum set of viewpoints in all three cases are developed separately in the form of graphs in section E .

D. Overview of Case Studies

1) Project A

Project A is software project developed by a software house (CMMI Level 3) that specializes in developing Financial, Business Management and E-government applications and project A is E-government in nature. Project A's architecture is built using optimum set of viewpoints by their software architect who has eight years experience in developing architecture of applications and Software Architect has used all views of optimum set to develop application's architecture due to project's complexity.

After analyzing data of questionnaire we found out that according to architect's views and analysis of questionnaire optimum set of viewpoints provide more coverage with respect to viewpoints and stakeholders' concerns as compared to the viewpoint model (i.e., Rational ADS with customization) which they usually follow for developing architecture because it ignores the internal structures of the application and hence the performance and reliability behaviors are not explicitly and individually captured, so these types of problems are sufficiently covered by optimum set of views. In case of quality attributes optimum set of views provides all applicable attributes. Suggestion given by Architect is that optimum set should define how things in one view are connected and complimented in the next view such as how uses cases are linked to class and sequence diagrams and how they are connected to test cases so an overall detailed inter connectivity needs to be defined.

2) Project B

Project B is software project developed by a software house (CMMI Level 2) that specializes in Data Management (Data warehouse, Business Intelligence, Data Mining,

 ISBN: 978-1-61208-367-4

Document Management Application Dev., Document Management Services) in Telecom and Banking Domains. They did not give much detail of project. Project B's architecture is built using optimum set of viewpoints by their Project Manager who has five years plus experience in developing architecture of applications and after that our questionnaire is filled by him in order to find coverage of optimum set of viewpoints. Project Manager has used all views of optimum set to develop application's architecture due to project's complexity.

After Analyzing data of questionnaire we found out that according to architect's view and analysis of questionnaire optimum set of viewpoints provide more coverage of business needs and maximum completeness of software architecture aspects i.e., viewpoints, stakeholders and quality attributes by customizing already available software architecture solutions. Being the SEI / CMMI certified firm they usually follow SEI's views with customization to work for implementation of data warehouse and business intelligence projects.

3) Project C

Project C is software project developed by a software house that specializes in managing the entire office automation system and providing IT support to defense organizations and project C is web based document management and filing system. Project C's architecture is built using optimum set of viewpoints by their project manager who has four years experience in developing architecture of applications. Software Architect has used all views of optimum set except Component & Connector View type to develop application's architecture.

After Analyzing data of questionnaire we found out that according to architect's views and analysis of questionnaire optimum set of viewpoints provide more coverage with respect to viewpoints and quality attributes as compared to the software architecture processes or models (i.e., RUP and Rational ADS with customization) which they usually follow for developing architecture. In case of quality attributes optimum set of views provides high availability as compared to approach followed by them. Suggestion given by project manager is use case viewpoint should be added in list of viewpoints.

E. Case Studies Results

1) Coverage of Viewpoints

Figure 3 shows coverage of software architecture viewpoints by optimum set of viewpoints after applying it on case projects. Out of 12 viewpoints optimum set of viewpoints provides 100% coverage, i.e., 12 viewpoints in first case study, 92% coverage, i.e., 11 viewpoints in second case study and 83% coverage, i.e., 10 viewpoints in third case study.

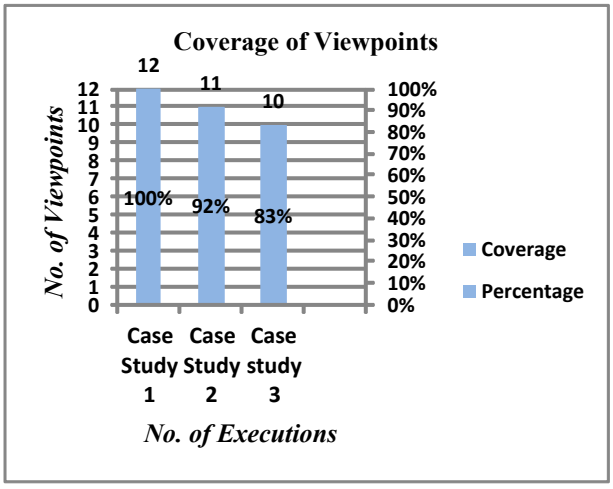

Figure 3. Coverage of Viewpoints by Optimum set of Viewpoints

From analysis of questionnaire results it is shown that viewpoints such as shared data, uses, generalization, implementation and work assignment which are not covered by most of models, are covered in detail by optimum set of viewpoints.

2) Coverage of Stakeholders

Figure 4 shows coverage of software architecture stakeholders by optimum set of viewpoints after applying it on case projects. Out of 21 stakeholders optimum set of viewpoints provides 100% coverage, i.e., 21 stakeholders in first case study, 100% coverage, i.e., 21 stakeholders in second case study and 76% coverage, i.e., 16 stakeholders in third case study.

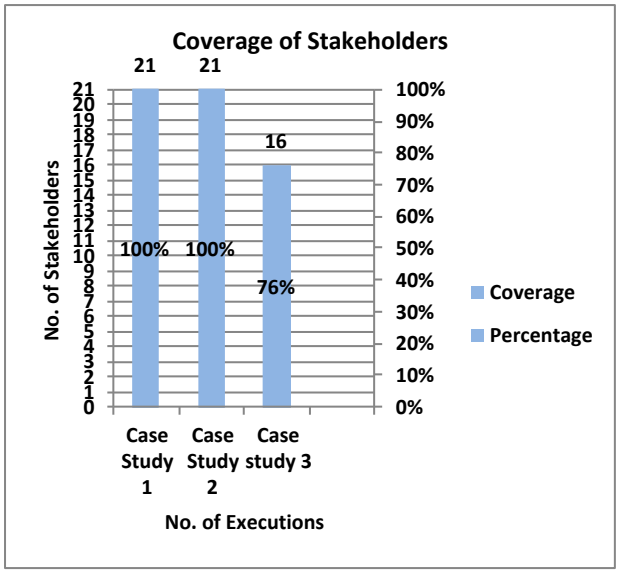

Figure 4. Coverage of Stakeholders by Optimum set of Viewpoints

From analysis of questionnaire results it is shown that stakeholders such as Designers, External System Architects Quality Assurance Team, Product line Managers, Suppliers, Support Staff and Project Managers which are not covered by most of models are covered in detail by optimum set of viewpoints.

3) Coverage of Quality Attributes

Figure 5 shows coverage of software architecture quality attributes by optimum set of viewpoints after applying it on case projects. Out of 25 quality attributes optimum set of viewpoints provides 100% coverage i.e., 25 quality attributes in first case study, 80% coverage i.e., 20 quality attributes in second case study and 96% coverage i.e., 24 quality attributes in third case study.

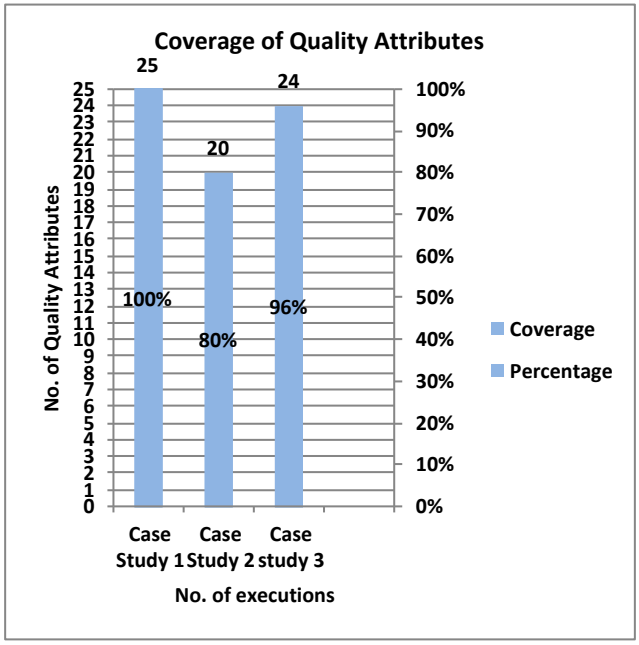

Figure 5. Coverage of Quality Attributes by Optimum set of Viewpoints

From analysis of questionnaire results it is shown that quality attributes such as security, modifiability, integrability, safety, supportability, projected lifetime of the system and testability which are not covered by most of models are covered in detail by optimum set of viewpoints.

F. Discussion

Mean coverage of concepts that are needed to efficiently design and analyze software architecture i.e., viewpoints, stakeholders and quality attributes is calculated for optimum set of viewpoints and compared to coverage of viewpoints by five software architecture viewpoint models and it is shown that optimum set of viewpoints provide more coverage of concepts than surveyed individual models.

Figure 6 shows comparison between optimum set of viewpoints and surveyed individual models with respect to coverage of viewpoints. Optimum set of viewpoints provide more coverage as compared to individual models. SEI coverage and optimum set of viewpoints coverage is same in case of viewpoints because our comparison framework is based on IEEE 1471 Standard i.e., Recommended Practice for Architectural Description of Software-Intensive Systems and SEI model provides template for more than one representation to describe contents of view in order to conform with the IEEE 1471 and can cover all details.

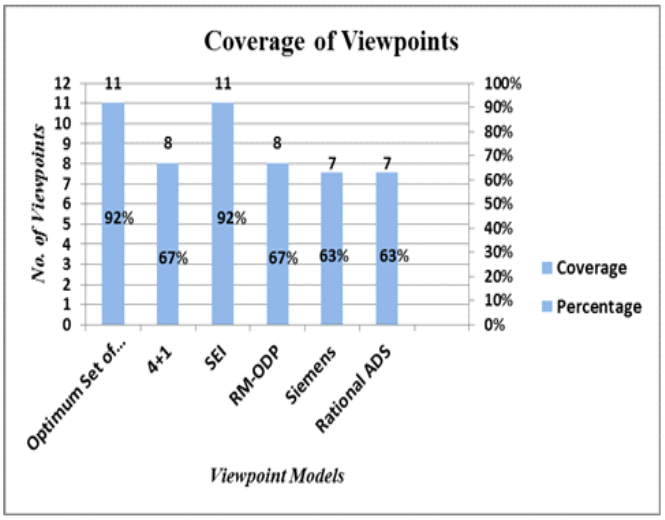

Figure 6. Comparison of Coverage of Viewpoints by Optimum set of Viewpoints with individual viewpoint models' coverage

Figure 7 shows comparison between optimum set of viewpoints and surveyed individual models with respect to coverage of stakeholders. Optimum set of viewpoints provide more coverage as compared to individual models.

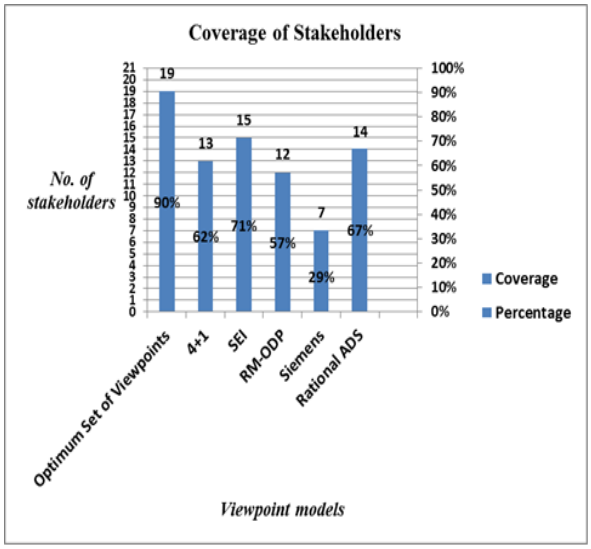

Figure 7. Comparison of Coverage of Stakeholders by Optimum set of Viewpoints with individual viewpoint models' coverage

 ISBN: 978-1-61208-367-4

Figure 8 shows comparison between optimum set of viewpoints and surveyed individual models with respect to coverage of quality attributes. Optimum set of viewpoints provide more coverage as compared to individual models.

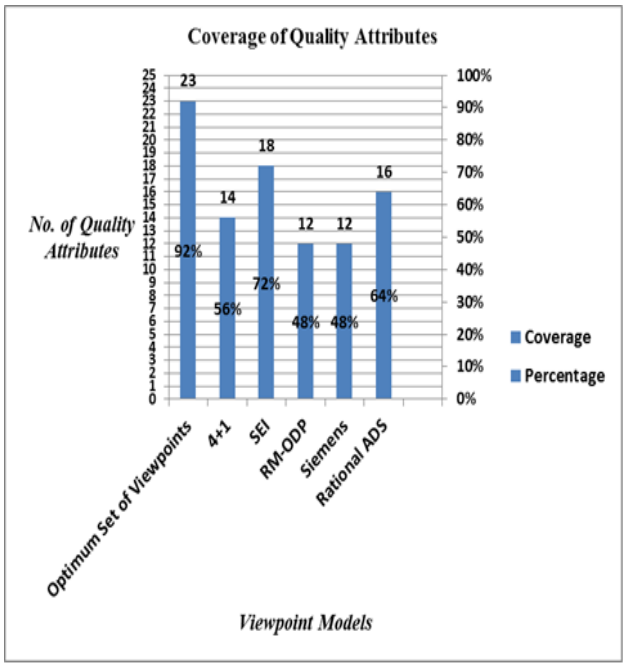

Figure 8. Comparison of Coverage of Quality Attributes by Optimum set of Viewpoints with individual viewpoint models' coverage

Also from case studies, it is concluded that Optimum set of views provide more coverage with respect to viewpoints, stakeholders and quality attributes of software architecture domain, than what can be achieved via individual architecture model alone.

G. Limitations

Due to resource limitations and confidentiality issues, we were not able to triangulate our findings by software architectural documentation analysis and face to face interviews which can provide in depth analysis. Furthermore, close ended questions in questionnaire has Yes\No\Partial\Not Applicable options, so while analyzing questionnaire results we assign same scale to partial option as Yes option regarding coverage of Software architecture concepts because we have to compare coverage of optimum set of viewpoints with coverage of surveyed viewpoint models whose coverage were determined by review of literature not by software architectural documentation analysis and from review of literature partial coverage cannot be find out.

IX. Conclusion And Future Work

A. Conclusion

There are a number of viewpoint models that create architecture document by means of the separation of the concerns. Each one of them describes viewpoints set and recognizes the concerns that each of them address. But none of them provides complete coverage of software architecture domain. So, a set of optimum viewpoints is selected from different software architecture viewpoint models after comparing them on a common comparison framework that allows combining views from different viewpoint models.

Then we present a Multiple-case study on the application of optimum set of viewpoints to three software development projects. From the results of case studies it is concluded that Optimum set of views provide more coverage with respect to viewpoints, stakeholders and quality attributes of software architecture domain, than what can be achieved via individual architecture model alone.

B. Future Work

In the future, this work can be augmented by additional case projects and data can be collected and analyzed from several sources i.e., architectural documentation and face to face interviews to get a more complete understanding of coverage of software architecture concepts.

Furthermore, by modeling system from architectural documentation with five surveyed models we can get a clearer picture of their coverage of software architecture concepts and also their partial coverage of concepts can be found, which cannot be found via literature.

References

[1] N. May, "A survey of software architecture viewpoint models", Proc. of the Sixth Australasian Workshop on Software and System Architectures, March 2005, pp. 13-24.

[2] N. Rozanski and E. Woods, "Applying Viewpoints and Views to Software Architecture", 2005.

[3] K. Smolander," What is included in software architecture? A case study in three software organizations", Proc. Of the Ninth Annual IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, April 2002, pp. 131–138.

[4] E. Woods, "Experiences using viewpoints for information systems architecture: an industrial experience report", Proc. of Software Architecture First European Workshop EWSA, 2004, pp. 182-193.

[5] A. Tang, J. Han and P. Chen, "A Comparative Analysis of Architecture Frameworks", Proc. of the 11th Asia-Pacific Software Engineering Conference, 2004, pp. 640-647.

 ISBN: 978-1-61208-367-4

[6] P. Clements, ”Comparing the SEI's Views and Beyond Approach for Documenting Software Architectures with ANSI-IEEE 1471-2000”, Technical Note 017, Software Engineering Institute, Carnegie Mellon University, 2005.

[7] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering." Empirical software engineering, vol. 14.2, 2009, pp. 131-164.

[8] L. Bass and R. Kazman, “Architecture-Based Development”, Technical Report 007, Software Engineering Institute, Carnegie Mellon University, 1999.

[9] M. Shaw and D. Garlan, “Software Architecture: Perspectives on an Emerging Discipline”, Prentice-Hall, 1996.

[10] K. Smolander and T. Päivärinta , "Describing and Communicating Software Architecture in Practice: Observations on Stakeholders and Rationale", Proc. of the Fourteenth International Conference on Advanced Information Systems Engineering, 2002, pp. 117-133.

[11] Z. Baida, “Stakeholders and Their Concerns In Software Architectures”, October 2001.

[12] L. Bass, P. Clements and R. Kazman, “Software Architecture in Practice”, 2nd edition, Addison Wesley, 2003.

[13] S. Fricker, T. Gorschek and P. Myllyperkiö, “Handshaking Between Software Projects and Stakeholders Using Implementation Proposals”, Proc. of the 13th International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer, 2007, pp. 144 – 159.

[14] W.H. Huen, “Systems Engineering of Complex Software Systems”, Proc. of 37th ASEE/IEEE Frontiers in Education Conference, October 2007.

[15] J Asundi, R. Kazman and M. Klein, “An Architectural Approach to Software Cost Modeling”, SEI Interactive, March 2000.

[16] N. Rozanski and E. Woods, “Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives”, Addison-Wesley, 2005.

[17] Malan, R. and D. Bredemeyer, “Defining non-functional requirements”, Bredemeyer Consulting white paper, 2001.

[18] F. Losavio, L. Chirinos, N. Lévyand and A. Ramdane-Cherif, “Quality Characteristics for Software Architecture”, Journal of Object Technology, Vol. 2, No. 2, 2003, pp. 133-150.

[19] IEEE. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. Institute of Electrical and Electronics Engineers, Sept. 2000.IEEE Std 1471-2000.

[20] P. Kruchten. “Architectural Blueprints - The “4+1” View Model of Software Architecture”, IEEE Software, vol. 12, Issue no. 6,1995,pp. 42–50.

[21] P. Clements et al. “A practical method for documenting software architectures”, Proc. Of the 25th ICSE, 2003.

[22] ISO. Reference Model of Open Distributed Processing (RMODP). International Organization for Standardization, Technical Report 10746, 1994.

[23] D. Soni, R.L. Nerd, and C. Hofmeister, “Software architecture in industrial Applications”, Proc. of International Conference on Software Engineering,1995, pp. 196–207.

[24] P. Clements et al., “Documenting Software Architecture: Views and Beyond”, Addison Wesley, 2002.

[25] D. Norris, “Communicating Complex Architectures with UML and the Rational ADS”, Proc. of the IBM Rational Software Development User Conference, 2004.

[26] D. Greefhorst, H. Koning and H. van Vliet, “The Many Faces of Architectural Descriptions”, Information Systems Frontiers, vol. 8, Issue no. 2, 2006, pp 103-113.

[27] S. Roselin Mary and P. Rodrigues,” Software Architecture- Evolution and Evaluation”, (IJACSA) International Journal of Advanced Computer Science and Applications, vol. 3, no.8,2012.

 ISBN: 978-1-61208-367-4