

Criteria for Designing Cloud Migration Framework

¹Pooja Parnami, ²Aman Jain, ³Navneet Sharma

¹Assistant Professor, ²Professor, ³Associate Professor

¹Computer Science Department,

¹IIS University, Jaipur, India

Abstract : *Migrating a legacy software application to cloud environment is a challenging job for IT industry and it requires a systematic approach. The paper provides a review of existing cloud migration approaches and highlights their similarities and key differences. It provides a list of common concerns which are important while designing a cloud migration process model. The paper provides a broad overview of current cloud migration scenario to practitioners and researchers of cloud computing community. The paper also identifies the gaps in existing approaches and provides future research directions for designing a situation specific approach for migrating a legacy application to cloud.*

IndexTerms – Legacy Application, Cloud Computing, Cloud Migration

I. INTRODUCTION

The cloud community proposed various methods for migrating legacy application to cloud. An effective cloud migration methodology can help a developer to successfully migrate a legacy application to cloud, in a predefined way. The existing literature focuses on one or few aspects of cloud migration like cost estimation, migration tools, testing etc. These studies or methodologies provide only partial understanding of cloud migration. Currently there is a need to identify common activities or steps to migrate a legacy application to cloud. This paper attempt to answer following research questions:

1. What are the existing approaches for legacy to cloud migration?
2. What generic criteria in terms of software engineering supported by these approaches?
3. What are cloud specific criteria are supported by these approaches?

II. SURVEY

To provide state of art of “legacy to cloud migration process” a Systematic Literature Review (SLR) has been conducted with the help of guideline provided by (Kitchenham et al., 2009)[1]. Following steps have been followed in SLR process.

- Defining the Search String
- Selecting the Study Sources
- Defining Inclusion and Exclusion Criteria
- Conducting Review

201 primary papers were identified as a result of the first iteration. In second iteration after extensive study of each paper’s content, the papers were scrutinised on the basis of inclusion and exclusion criteria, Eighty five papers were the output of this step.

III. EVOLUTION CRITERIA

After collection of primary papers, to assess the collected cloud migration methodologies and answer the research questions some evolution criteria required. For a fair evolution of a framework, Karam and Casselman[2] suggested some criteria. According to them a framework:

- Should be Generic
- Precisely characterize similarities and differences of methodologies
- Should cover all important requirements of methodologies

The above mentioned criteria could be analyzed with the help of certain characteristics of existing frameworks. These characteristics could be divided into two main sets of criteria:

1. Generic
2. Cloud Specific

3.1 Development of Generic evolution criteria:

For the development generic criteria, the review of existing frameworks has been performed. Following is the list of meta-criteria suggested by different researcher.

- i. A cataloging framework for software development methods. [2]
- ii. A guide to the assessment of software development methods [3]
- iii. Process-centered review of object-oriented software development methodologies[4]
- iv. A framework for evaluating agent-oriented methodologies [5]
- v. Comparison of ten agent-oriented methodologies [6]

These frameworks define the evolution criteria for software development methodologies.

After collection of generic criteria from the above mentioned sources, a list of distinct criteria derived. Following table defines the list of generic criteria along with their definitions. A short description of activity is marked as partially-supported and detailed description is rated as fully-supported.

Table 3.1: Generic Criteria

#	Criteria	Definition
GC1	Clear description of cloud migration steps	Does the approach provide a clear description of migration activities/ steps for cloud migration?
GC2	Detail description of steps to perform cloud migration process	Does the approach define detail activities and supportive techniques for cloud migration?
GC3	Provides guidelines for tailoring purpose	Does the approach provide mechanism or guidelines for its tailoring to meet different migration scenarios?
GC4	Provides Modeling Language	Does the approach provide notation and semantic rules for work products?
GC5	Provides Tool Support	Does the approach offer or refer tools to automate migration activities?

GC1 : Clear description of cloud migration steps:

This criteria will help the developers or organizations who seek for a step by step process for legacy to cloud migration. According to our study around 79% of approaches provides full description of the suggested activities.

GC2: Detail description of steps to perform cloud migration process

It describes that if a given approach also provides the supportive techniques to perform an activity. Around 45% of approaches provides what activity to perform and how to perform.

GC3: Provides guidelines for tailoring purpose

Tailorability is an important aspect of reengineering. This property examines that an approach provides mechanism or guidelines for tailoring adherence during cloud adoption. Less than 5% of approaches inculcate this key concern.

GC4: Provides Modeling Language

A modeling language provides explains the semantic rules to express the outcome of an activity. This can increase the automation level and productivity of design process. This criterion is to find out that if an approach uses a notion to describe the output of each activity? Only two approaches use UML language as modeling the artifacts, and other four approaches uses some graphical diagrams and texts.

GC5: Provides Tool Support

For the automation of migration process, an approach can offer its own tools or suggests some third party tools. With this respect only 20% of offers some tools.

3.2 Development of Specific evolution criteria:

This approach defines the level of harmonization of existing cloud migration approaches with cloud migration process.

Table 3.2: Specific Criteria

#	Criteria	Definition
SC1	Context Analysis	Does the approach provide activities related to feasibility analysis of legacy application?
SC2	Understanding Legacy application	Does the approach collects the information regarding the legacy application's : data, functionality, programming language, tiers?
SC3	Functional and non-functional requirement analysis	Does the approach identify the objectives and requirements of target architecture?
SC4	Migration Planning	Does the approach provide a plan for rest of migration process and also a rollback plan?
SC5	Platform Selection	Does the approach identify candidate cloud service or platform for cloud migration?
SC6	Re-architecting of legacy application	Does the approach re-architects the legacy application to utilizes the cloud services like elasticity, multi-tenancy etc.?
SC7	Training	Does the approach address the skills and training needs for users and developers?
SC8	Testing	Does the approach provide activities to test the target architecture against the legacy application?
SC9	Continuous Monitoring	Does the approach monitor the application's components which are using cloud services?

SC1: Context Analysis

Feasibility analysis is an important activity. It is a process to guide how the legacy application will operate, function and maintained in the target environment. This criterion examines the level of context analysis followed by an approach. The study shows that only 25% of approaches explicitly includes context analysis.

SC2: Understanding Legacy application

Before migration it is important to identify the characteristics of legacy application like: architecture, programming language used, quality of code, dependencies, process flow, reusability etc. Around 53% of studies does not offer any activity to understand legacy application.

SC3 : Functional and non-functional requirement analysis

This criterion is to identify that if an approach suggests activities to find out the functional and non-functional requirements. 65% of approaches fully, 23% partially and 12% does not support this criterion.

SC4: Migration Planning

After feasibility analysis the detailed plan for migration is essential. But only 32% of approaches included this activity in their study.

SC5: Platform Selection

The alignment of attributes of application and cloud can result in low cost and effort. If the selected cloud technology is compatible with legacy, fewer code modifications are required. But only 19% of approaches support this criterion.

SC6: Re-architecting of legacy application

Re-architecting involves evolution of legacy applications towards a new architecture to utilize the cloud services. Few important concerns like defining target cloud architecture, enabling elasticity, enabling multitenancy, resolving incompatibilities should be considered during migration. This criterion is almost neglected by all the available cloud migration methodologies.

SC7: Training

The target application build on cloud uses new set of tools and API. It is highly recommended by some approaches to include training activity for IT staff in the migration process. But around 20% of methodologies fully or partially support this criterion.

SC8: Testing

The testing must be performed to check the functionality of target application against the functional and non-functional requirement captured in a previous activity. But only 23% of approaches inculcate this activity in their suggested migration method.

SC9: Continuous Monitoring

A continuous monitoring is recommended by few approaches due to unpredictable nature of cloud. 25% of approaches included activities to assure successful migration of legacy applications.

For the purpose of evaluation, a criterion must be measurable. To answer that an approach supports a criterion or not, scale points must be defined. The framework defines three levels of scale : Fully Supported (1), Partially Supported (0.5) and not-supported (0).

Table 3.3: Criteria evaluation scale

Generic Criteria(GC)		Score		
		Total= 1	Partial= 0.5	No= 0
GC1	Clear description of cloud migration steps			
GC2	Detail description of steps to perform cloud migration process			
GC3	Provides Modelling Language			
GC4	Provides guidelines for tailoring purpose			
GC5	Provides Tool Support			
Specific Criteria (SC)				
SC1	Context Analysis			
SC2	Understanding Legacy application			
SC3	Functional and non-functional requirement analysis			
SC4	Migration Planning			
SC5	Platform Selection			
SC6	Re-architecting of legacy application			
SC7	Training			
SC8	Testing			
SC9	Continuous Monitoring			

IV. CONCLUSION

The paper provides review of existing approaches, suggesting a methodological solution of migrating a legacy application to cloud. This was conducted through an exhaustive literature review and manual search on electronic scientific database. After screening of databases 83 papers have been selected related to legacy to cloud migration. This search helped to answer the first research question. While answering second research question, various key concerns have been found. Firstly most of the research does not use a sound research methodology to get cumulative understanding of cloud migration field. Secondly there is a lack of situation specific approach for migrating a legacy application to cloud, because each project has different characteristics and needs, therefore a single approach cannot be applicable to all migration scenario. Thirdly, the existing approach poorly supports features like multi-tenancy, elasticity, test and continuous integration and must add activities to empower cloud migration method.

Each year ample number of research papers, experience reports and solutions are published in cloud migration field, which explains it's the maturity of this field itself and demands for a generic reference model of migrating legacy application to cloud environment.

REFERENCES

[1] Kitchenham, Barbara, et al. "Systematic literature reviews in software engineering—a systematic literature review." *Information and software technology* 51.1 (2009): 7-15.

- [2] Karam, Gerald M., and Ronald S. Casselman. "A cataloging framework for software development methods." *Computer* 26.2 (1993): 34-44.
- [3] Wood, Bill, et al. *A guide to the assessment of software development methods*. No. CMU/SEI-88-TR-8. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 1988.
- [4] Ramsin, Raman, and Richard F. Paige. "Process-centered review of object-oriented software development methodologies." *ACM Computing Surveys (CSUR)* 40.1 (2008): 3.
- [5] Sturm, Arnon, and Onn Shehory. "A framework for evaluating agent-oriented methodologies." *Agent-Oriented Information Systems*. Springer, Berlin, Heidelberg, 2004.
- [6] Tran, Quynh-Nhu Numi, and Graham C. Low. "Comparison of ten agent-oriented methodologies." *Intelligent Information Technologies: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2008. 1375-1397.

