Risk Perception of Migrating Legacy Systems to the Cloud

Breno G. S. Costa and Priscila Solis
Department of Computer Science, University of Brasília, Brasília-DF, Brazil

Keywords: Cloud Computing, Legacy Systems, Migration, Reference Model, Risk Perception.

Abstract: With the utilization of Cloud Computing as the way to provide Information Technology services, the organizations can migrate legacy systems to the cloud in order to reach several benefits. There are in the literature several proposals to model the critical elements of migration and many have been validated by specific case studies. Also several reference models were defined on top of these proposals and were created with the intention to consolidate different researches, trying to expand their applicability. Based on the above, this paper selects a reference model for migrating legacy systems to the cloud and proposes a method for calculating a score of perceived risk associated with each legacy system migration. The paper presents a proof of concept on government domain to show the method's applicability.

1 INTRODUCTION

In the last years Cloud Computing (CC) changed the way Information Technology (IT) is provided and consumed and created a new landscape in which organizations manage their own computing infrastructure to another, where IT is consumed as a service (ISACA, 2012).

According to (Gartner, 2017), the massive use of public cloud services appears in the significant growth rates of service providers such as Amazon Web Services (AWS) (AWS, 2017) and Microsoft Azure (Microsoft, 2017), among others. The study surveyed almost 3000 participants and found that 21% of them have already used public cloud services, while 56% planned to implement the cloud by the end of 2017. For many of these organizations, IT modernization is almost synonymous of increasing the use of CC.

The migration to this new paradigm has a potential cost reduction of IT infrastructure, noticed by several authors (Armbrust et al., 2009; Armbrust et al., 2010; Gartner, 2017), mainly when related to the use of public cloud. According to (Pahl and Xiong, 2013), migration to the cloud is the process of deploying, in whole or in part, the digital assets, services, IT resources, or systems of an organization on the cloud. With the use of CC, government organizations can reduce the number of contracts and consequently reduce the opportunities for irregularities while improving efficiency. They can also focus on providing services to the population by reducing the operational effort to maintain the infrastructure and IT services platform (Kundra, 2011).

Migration to the cloud may involve retention of part of the infrastructure within the organization. For example, a legacy system can be merged with a complementary cloud solution, with integration over the Internet. The trend of increased use of CC, including on government organizations, may benefit of using a process (guidelines and methods) to migrate legacy systems to the public cloud.

The goal of this paper is to present an extension to a reference model of legacy systems migration to the cloud. The extension will be validated with a proof of concept in government domain. This paper makes the following contributions:

- based on a review of related work, proposes a method for calculating a score of perceived risks associated with the legacy migration to the cloud;
- describes a proof of concept showing the applicability of the method for the government domain.

The remainder of this article is divided into four sections. Section 2 presents the theoretical concepts and related work and also describes the motivation of evaluating the risk perception of applying the reference model. Section 3 presents the proposed evaluation method. Section 4 describes the proof of concept. Finally, Section 5 presents the conclusions and future work of this research.

Costa, B. and Solis, P.
Risk Perception of Migrating Legacy Systems to the Cloud.
DOI: 10.5220/0006801706340642
In *Proceedings of the 8th International Conference on Cloud Computing and Services Science (CLOSER 2018)*, pages 634-642
ISBN: 978-989-758-295-0

2 THEORETICAL CONCEPTS AND RELATED WORK

2.1 Migration Methods and Strategies

Software migration is the process of moving (or adapting) a system from one operating environment to another (ISO/IEC, 2006). Software migration can be seen as part of a broader context that is software maintenance and reuse (Ionita, 2013). Migration to the cloud is the effort to adapt legacy systems running locally on the organization's infrastructure to transport them to the cloud (Jamshidi et al., 2013).

To take advantage of CC benefits and protect existing investment in a legacy system, organizations tend to migrate their legacy systems to the cloud gradually when feasible (Zhao and Zhou, 2014; Gartner, 2017; Ionita, 2013). As discussed in (Pahl and Xiong, 2013), the migration process requires careful consideration, planning, and execution to ensure system security and integrity after migration, and to remain compliant with the organization's requirements.

According to (Babar and Chauhan, 2011), unsuccessful cloud migration can cause business processes to fail, increase security incidents, and increase maintenance costs. In addition to these considerations, the work of (Zhao and Zhou, 2014) discusses migration strategies and methods. For them, different migration strategies should consider different migration processes, with their subdivisions into specific tasks. In the study of (Binz et al., 2011), the authors propose an application migration framework for the cloud and between cloud providers.

One alternative point of view regarding migration to the cloud is the use of adaptation, addressed by (Andrikopoulos et al., 2013). The authors identify four types of legacy systems migration. In (Zhao and Zhou, 2014), the authors analyze and compare the two migration definitions cited in (Binz et al., 2011) and (Andrikopoulos et al., 2013), among other definitions and categorized migration to the cloud into three major areas: migration to Infrastructure as a Service, migration to Platform as a Service and migration to Software as a Service.

One of the main benefits of using CC is the potential cost reduction of IT infrastructure investments (Armbrust et al., 2009; Armbrust et al., 2010; Gartner, 2017). However, according to (Sun and Li, 2013), this benefit typically only considers the cost of cloud services after migration. The authors developed a method to estimate the cost of the migration process from legacy systems to cloud. The method considers that, although described sequentially, the execution flow of the migration tasks iterates in a way in which errors in the execution of a task can cause the repetition of previous tasks. Thus, according to the size and complexity of the legacy system and also to the maturity of the team responsible for the migration, the cost of the migration process can be calculated using the method they have defined.

2.2 Characterization Model

Many publications address how to structure the migration of legacy systems to the cloud. The Cloud-RMM model (Jamshidi et al., 2013) categorizes twenty-three studies on the migration of legacy systems to the cloud and serves as a guide to the analysis of these studies. The model is composed of four processes and each process is a group of tasks.In the model, there is also an indication of the artifacts generated at the end of each process.

In this model, there is only a consolidation of processes, tasks and artifacts, found in the twenty-three studies analyzed in the systematic literature review undertaken by (Jamshidi et al., 2013). There is no discussion about the need and relevance of each process and task.

2.3 Evaluation Model

The work of (Gholami et al., 2016) provides a detailed assessment of existing CC migration approaches from the perspective of process modeling and software development methodologies. The authors propose an evaluation model to compare existing approaches, highlighting their resources, similarities and differences. The approach used by the authors differs from other related works because it focuses on the aspect of the cloud migration process to understand which core activities and concerns are involved during this transition. According to their analysis, none of the reviewed studies provides an in-depth discussion of proposed migration features and activities and also did not bring useful experience of applying those approaches in practice. In addition, the article provides a meticulous analysis of existing approaches through an evaluation model which encompasses twenty-eight criteria classified into two dimensions. That is, eleven generic criteria and seventeen specific criteria for cloud computing. The proposed framework was derived from a literature review and validated through a web-based questionnaire survey of 104 academics and experts in the field of cloud computing (Gholami et al., 2016).

As challenges for future work, (Gholami et al., 2016) acknowledge that there is a large amount of research on cloud migration, which is currently dispersed and fragmented, and suggest that a generic reference model must be defined with the goal of consolidating the existing literature.

2.4 Reference Model

According to (Fettke and Loos, 2003), a reference model is a conceptual framework that can be used as the model for information systems development. Reference models are also called universal models, generic models, or standard models. To use reference models, they must be tailored to the requirements of a specific domain. The study presents several perspectives for the evaluation of reference models. Among them, the empirical perspective, from which we mention two approaches that relate to the object of this research: case study and survey.

In a more recent study, (Gholami et al., 2017) responded to the challenge proposed in (Gholami et al., 2016) and worked to find out the critical activities, artifacts, concerns and key recommendations regarding the migration of legacy systems to the cloud. Results were validated empirically, through the collection of experts perceptions to increase the reliability.

Based on an intensive qualitative analysis of existing constructs in the literature, the authors created a migration to the cloud reference model. The relevance and robustness of the model was further confirmed using quantitative research and qualitative feedback from domain experts. Figure 3 shows a summary of the resulting reference model. To simplify the view, the summary shows only the model's key elements, without the subdivisions of activities into smaller tasks and without the flow of information between them. The detailed view of phases Plan and Design will be better described in Section 4.

Although this reference model has been created from the literature and evaluated using survey with CC experts, the authors do not claim its general applicability. In the opposite sense, they assert that there is no universally superior or applicable method for all cloud migration scenarios and therefore, methods must be tailored to the specific characteristics of the application domain.

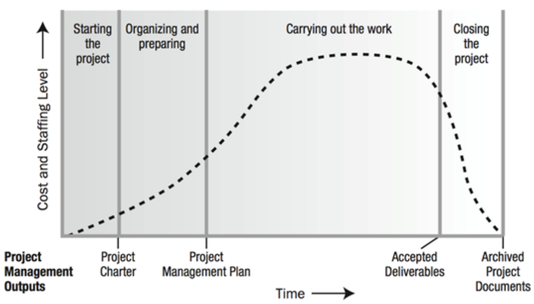

Figure 1: Project life cycle as defined on PMBOK (PMI, 2013).

2.5 Legacy Systems Migration Viewed as an IT Project

Each legacy system migration to the cloud can be viewed as a different project, following what is outlined in the Project Management Body Of Knowledge (PMBOK)(PMI, 2013): it is a temporary endeavor and creates a unique product: the legacy system migrated to the cloud. PMBOK defines the life cycle of a project as being the series of phases, usually sequentially linked, that a project goes through. Figure 1 shows the generic structure of a project life cycle and the level of costs and personnel required at each stage of the life cycle.

According to the PMBOK, the generic structure of the life cycle presents, among others, the following characteristics:

- The personnel and cost levels are low at the beginning, they reach a maximum value while the project is running, and fall quickly as the project is finalized. Figure 1 illustrates this pattern.
- The risks and uncertainties are greater at the beginning of the project. These factors decrease over the life of the project as decisions are made and deliveries are accepted, as shown in Figure 2.
- The ability to influence the final characteristics of the project's product, without significant impact on costs, is higher at the beginning of the project and decreases as the project progresses to its end. Figure 2 illustrates the idea that change costs and bug fixes generally increase significantly as the project nears completion.

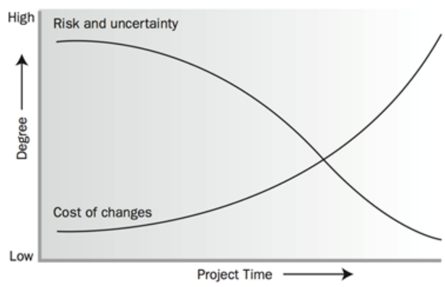

Figure 2: Project risks and cost of changes through the time (PMI, 2013).

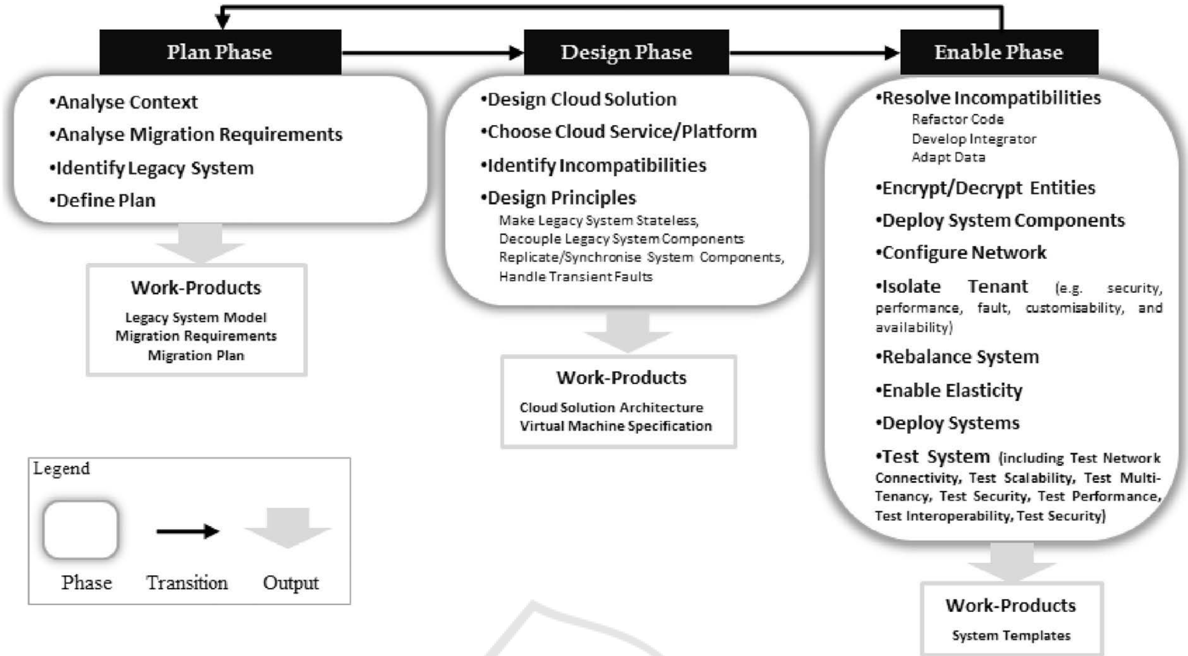

Figure 3: Key elements of Legacy to Cloud migration Reference Model proposed by (Gholami et al., 2017).

3 REFERENCE MODEL EVALUATION

The previous section presented some definitions and classifications of migration models and analyzed three of them: a characterization model that grouped processes, activities and artifacts referenced in 23 migration proposals published between 2010 and 2013; an evaluation model that, based on 28 relevant criteria, evaluated 43 articles published between 2009 and 2015; and a reference model (Gholami et al., 2017), selected as the basis for this study, defined from the consolidation of 78 proposals published between 2008 and 2015. The selected reference model is divided into three phases: Plan, Design and Enable as shown on Figure 3. The Plan phase is responsible for collecting relevant information about the system to be migrated (technical information, organizational context) and, in accordance to migration requirements, for creating an appropriate migration plan. The Design phase uses the knowledge and artifacts generated in Plan phase and is responsible for choosing one or more CC providers and also for defining the new architecture that the legacy system will have on the cloud. The Enable phase represents the actual migration. It comprehends the implementation of legacy system code adaptations, the build of integrators when appropriate, the configuration of the CC services defined in Design phase, testing and go live of migrated system.

As shown in Figure 1, the effort required to execute the third phase is potentially bigger than what will be spent in the two initial phases. This is due to the fact that first phases have mostly analytical work, while the third, mostly implementation work. Implementation in this context means both the development related to the adaptations and new integrations, when necessary, and cloud services configuration.

Thus, it is important to have an early assessment of migration risks before Enable phase. The assessment may indicate an increased risk perception (migration additional cost and time, for instance) for a given system when compared to the others, and thus provide an objective measure of comparison. It is possible to rank the systems that will be migrated and set up an order of execution that prioritizes the systems with less perceived risk, thus increasing the success rate and organization's confidence on the migration project.

As a way to provide an objective and early assessment of perceived risks of migration of each legacy system to the cloud using the Reference Model, we defined the following procedure: in each of the two initial phases (Plan and Design), one must select the tasks that together better represent the critical points of each legacy system migration to the cloud. Some tasks have a job of merely collecting and aggregating information, while others implement constraints, decisions or characteristics that are intrinsic to the system and the organization and thus have the potential to impact the rest of the migration. One should choose between tasks in the second category. It is important to realize that different application domains may have different sets of tasks that better evaluate the risk perception of migrating legacy systems to the cloud on that domain.

After choosing the set of evaluation tasks, it is necessary to define the weights that each task will have in relation to the other tasks on the set. The weight of tasks is related to the importance of that task in the domain of application. For instance, on government domain the task related to procurement is perceived as being more risky than the one related to training, due to legal and regulation rules.

The evaluation function defined here has the following format:

$$EvF(Pl, De) = \frac{Pl + De}{2} \tag{1}$$

where:

$$Pl = \frac{\sum_{i=1}^{n} V_{Pl_i} W_{Pl_i}}{\sum_{i=1}^{n} W_{Pl_i}} \tag{2}$$

$$De = \frac{\sum_{j=1}^{n} V_{De_j} W_{De_j}}{\sum_{j=1}^{n} W_{De_j}} \tag{3}$$

and:

Pl – Plan Phase Evaluation Indicator.

De – Design Phase Evaluation Indicator.

V_{Pl_i} – Risk perception rate for task i of the Plan Phase evaluation set.

W_{Pl_i} – Weight of task i in Plan Phase evaluation set.

V_{De_j} – Risk perception rate for task j of Design Phase evaluation set.

W_{De_j} – Weight of task j in Design Phase evaluation set

The function defined in (1) considers that both phases have the same relative importance on calculating the the score of perceived migration risks. But specific circumstances could be considered on another evaluation function that balance the phases differently.

After selecting the set of evaluation tasks in each phase and the weight of each of them, they must be evaluated. Note that the work required on each task must be performed as defined in the Reference Model. At the end of the execution of each phase, and consequently at the end of the execution of all tasks of that phase, is when the evaluation method is performed. This is an important consideration because it is necessary to know each system in an appropriate level of detail to do a more accurate evaluation and this can not be done by executing only the tasks on the evaluation set. So, after the end of each phase's work, it is necessary to evaluate each task that is part of the evaluation set. The proposal is to use a scale of 5 values for perceived risk, whose meaning could be 1 – High risk, 2 – Moderate risk, 3 – Average risk, 4 – Somewhat risk and 5 - Low risk. For each task, the 5 levels of the scale can also be defined textually, in order to facilitate the choice and evaluation of each system. This must be done only once and only for the tasks that are part of the evaluation set.

Table 1: Plan Phase evaluation task set.

Task	Weight
Analyze migration cost	3
Identify dependencies	3
Select migration scenario	1

Once the scale is defined and the Plan Phase is finished, the evaluation can start by defining, for each task in the evaluation set, the rating of perceived risk associated with that task's subject in relation to legacy migration and calculating the evaluation indicator of the Plan Phase, *Pl*, as described in (2). The same should be done to Design Phase by calculating the evaluation indicator for the Design Phase, *De*, described in (3). After calculating the two indicators, one can calculate the perceived risk score of migrating the legacy system to the cloud, $EvF(Pl, De)$, as described in (1).

After calculating the evaluation function for each system that will be migrated, a ranking can be established that indicates, among other things, which systems should be migrated first, which will be the systems with higher evaluation values (higher perceived risk scores). This strategy is reinforced by (Reza Bazi et al., 2017) who states that it is better to select a pilot project at first, when performing migration process. Systems with lower evaluation values may receive additional analysis to identify whether work modifications to the evaluation tasks that are causing low evaluation are feasible. Even when could not be possible to change something in systems with low evaluation values, knowing that there is an increased risk perception is of great value. In addition, when using evaluation function values to rank systems, one can also use them to categorize systems on a previously defined risk scale. For this purpose, evaluation thresholds can be defined. For example, one can define that evaluation function results that are below or equal to a certain threshold $T1$ mean that the system should have its migration decision reanalyzed. Another example is when the value calculated by evaluation function is above the threshold $T1$, but the value of one of the indicators (*Pl* or *De*) is below a given threshold $T2$.

The threshold values can be defined by using a certain value of the Likert scale used, but can also be defined by the history of evaluations already done and considering the result of these migrations. While there is no history of assessments, it is suggested to use the central value of the scale as the threshold $T1$ and $T2$ and adjust them as soon as new values have been calculated from actual migrations.

4 PROOF OF CONCEPT

The proof of concept in this section presents an analysis of the selected activities and defines weights that best support a risk perception evaluation for the migration of legacy systems in the government domain.

4.1 Plan Phase

For the Plan phase, we selected the tasks and define the weights shown in the Table 1. The reasons why the tasks were selected are the following:

Analyze migration cost - This activity results in an estimate of migration cost. Cost is associated with the effort required to migrate the legacy to cloud as well as the cost of running the application after migration. As cost of migrating a system increases, also increases the risk perception for the entire migration process. On the other hand, a low-cost migration system can be a good candidate to validate the migration process and to gain experience with both the process and the provider to which the system will be migrated.

If the cost of the migration added to the cost of running the system in the cloud provider does not represent an economy relative to the current cost, this may indicate a system that will not benefit from the cloud if there is no value aggregation by other means, such as the use of intrinsic cloud characteristics (self-scalability, higher availability, for example) (Gholami et al., 2017). According to (Gholami et al., 2017), the objectives of Analyze Context activity (which is an aggregate of Analyze migration cost and other tasks) are cost estimate and risk mitigation.

Identify dependencies - This activity has the effect of identifying which other local systems and components the system being analyzed depends on to function properly on the cloud after migration. With this task, it is possible to identify which other systems would have to be previously migrated to the cloud. Or, in the case of a hybrid cloud and if the dependencies are not migrated yet, one will need to evaluate and consider the network latency between the local infrastructure and the cloud. According to (Reza Bazi et al., 2017), this is a critical factor to be considered, since legacy systems were developed on older platforms than the current version supported by cloud providers. The objective of Recover Legacy application knowledge (activity which aggregates Identifying dependencies task) is Understand legacy system dependencies (Gholami et al., 2017).

Select migration scenario - According to the characteristics of the legacy system and also according to the amount of effort that is intended to be spent in the migration, one may choose the migration type among five options (Andrikopoulos et al., 2013; Gholami et al., 2016). Migration type V, for instance, is a type associated to a low risk of failure when compared with other types, since it is based on moving the whole system stack to the cloud. Thus, the choice of the scenario may be directly related to the risk intensity that is accepted. This is not the same to say that a scenario of low risk is also the option that will bring the greatest benefit (cost reduction, increased availability), because scenario V, for instance, could make costly the implementation of elasticity (Andrikopoulos et al., 2013). The objectives of Define Plan (activity that aggregates Select migration scenario task) are Project management and Risk mitigation (Gholami et al., 2017).

Table 2: Design phase evaluation task set.

Task	Weight
Negotiate with cloud provider	3
Train	1
Identify incompatibilities	1

4.2 Design Phase

For the Design phase, we selected the tasks and define the weights shown in the Table 2.

Negotiate with cloud provider - This task is critical because only after its execution one or more cloud providers are available to provide the cloud services that will support migrated systems. If one have not still hired the provider at migration time of a particular legacy system, this may delay the migration or cause rework. Note that even after a provider has already been hired and some legacy systems have been migrated to the cloud, it is possible that the contract could be reaching its end, or even that the provision of the service is not satisfactory and that the organization has decided to replace the provider, even before the end of the contract.

Training - This task is responsible for acquiring and maintaining the necessary knowledge to properly design, operate and monitor the CC services. A

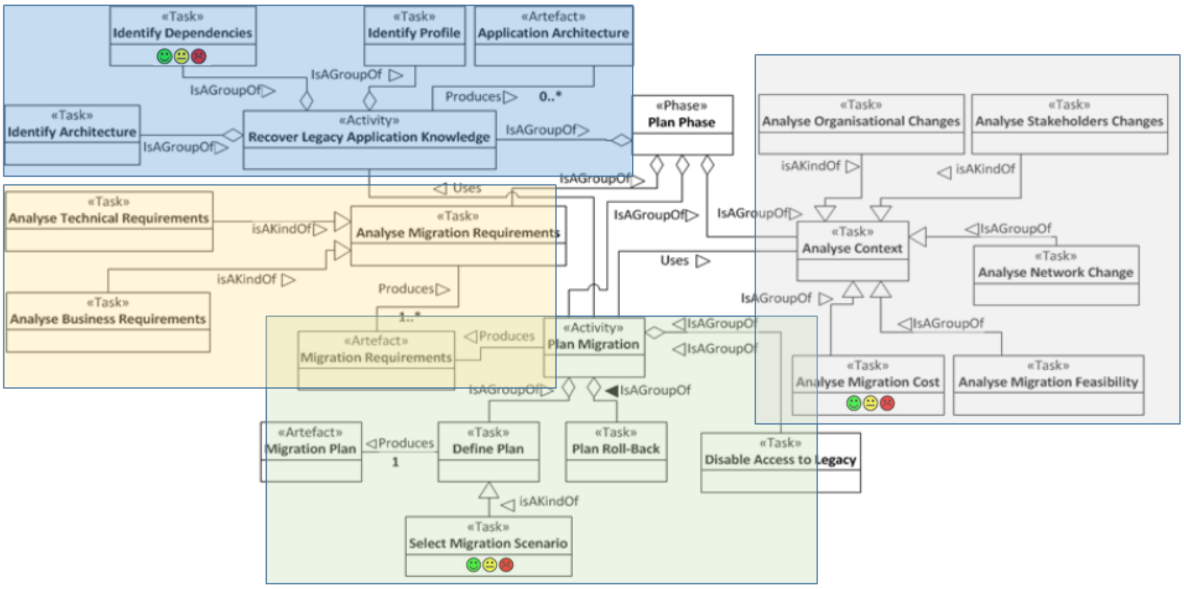

Figure 4: Reference Model's Plan phase with visual indicative of task evaluation set.

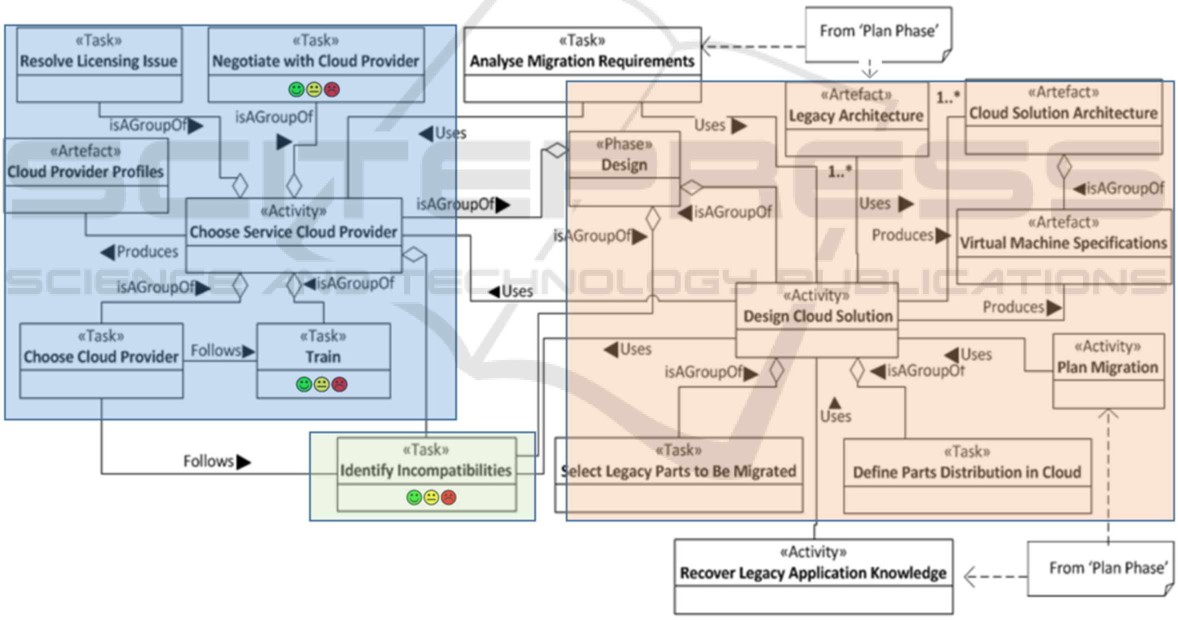

Figure 5: Reference Model's Design phase with visual indicative of task evaluation set.

higher need for training may indicate a higher risk in managing the migration project of a given legacy system. According to (Reza Bazi et al., 2017) organizations must extend their cloud knowledge as a way to guarantee a successful start. The objective of Choose Cloud Provider (activity that aggregates both Negotiate with the cloud provider and Train tasks) is to find the best providers that meet migration requirements (Gholami et al., 2017).

Identifying incompatibilities - This task is responsible to find the incompatibilities between the legacy system and the set of CC services defined in the design phase as being required to run the system after migration to the cloud. These incompatibilities will require specific effort to be solved. The objective of the activity Identifying incompatibilities is to estimate effort and cost to resolve incompatibilities (Gholami et al., 2017).

The above explanations of what motivated task selection for the government domain are illustrated in a

Table 3: Rating of risk perception associated with tasks on evaluation set (shaded lines:Plan phase; white lines:Design phase).

Task	Weight	Rate	Reason
Analyze migration cost	3	5	Low complexity system and low estimated migration cost.
Identify dependencies	3	4	System that has almost no dependency.
Select migration scenario	1	4	Scenario type V, lift and shift.
Negotiate with cloud provider	3	4	Providers selected.
Train	1	2	High necessity of training. No experience on CC.
Identify incompatibilities	1	5	No incompatibilities found.

detailed view of Reference Model phases with a visual identification of the tasks in Figures 4 and 5. The figures show how each key element is associated to the others. Each key element has been associated with a color, which covers the tasks that compound that key element. In addition, there is a specific indication (three colored smileys) in the tasks that are part of the evaluation set.

4.3 Use Case

To show the method's applicability, we chose a legacy system in use by a government body that decided to migrate its legacy systems to the cloud. The legacy system is called Civic Cloud and exposes, on the Internet, updated data about educational and health institutions around the country and data about medicines that are authorized by the proper government agency. Although the system has cloud on its name, it runs on local infrastructure.

The system is composed by a set of loosely coupled web services that can be used by developers and companies to add value to their applications. For instance, one can develop and publish a mobile application that automatically captures user location and suggests the health institution that is closer to the user. The system was developed using Java language with the support of Spring MVC framework. It is running on application server JBOSS EAP, sharing computational resources with other legacy systems. The database management system is Oracle (Oracle, 2017).

As the system is publicly available, any developer can build an application that makes calls to its services and uses the provided data. If one of these applications becomes a killer application, with hundreds or thousands of transactions by hour, there is a chance that the computing infrastructure becomes inadequate due to the lack of elasticity.

We worked with the team responsible by the system in order to apply the reference model for migration to the cloud. After doing the work prescribed on initial phases, Plan and Design, we applied the method described on the Section 3, using the tasks and weights defined on the Section 4.

The team shared their perception that the system was a good choice to be the first legacy to be migrated by this government body. This is explained due to the low number of integrations with other legacy systems and to the low complexity and size of system code.

The team then rated the risk perception for each task on the evaluation set of both phases. The values can be viewed on table 3, along with the main reason that justified the rating.

Using the weights defined on tables 1 and 2, the rates given by the team (table 3) and applying equations (2), (3) and (1), we have: $Pl = 4.43$, $De = 3.80$ and $EvF = 4.11$. The calculated value of EvF is the score of risk perception to migrate legacy system Civic Cloud to the cloud with the use of Reference Model. As suggested on Section 3, $T1$ and $T2$ are defined to be 3.0. Since EvF, Pl and De are higher than these thresholds, this indicates a low risk perception for this system migration.

5 CONCLUSIONS AND FUTURE WORK

This paper presented a review of software migration strategies, compared three conceptual migration models and selected a reference model as a basis for legacy systems migration to the cloud. Based on the concepts, we proposed a method for calculating a score of perceived risk to the systems that will be migrated to the cloud. Method's steps are described along with a proof of concept in the government domain. The proposed method can be used to rank the systems to be migrated to the cloud, offering an opportunity to assess migration risk perception before migration execution.

The future work of this research aims to apply the method on other government systems and collect the results in different scenarios. Also, these results could be validated with a survey, applied over different government organizations, to improve and verify the proposed method.

REFERENCES

Andrikopoulos, V., Binz, T., Leymann, F., and Strauch, S. (2013). How to adapt applications for the cloud environment. *Computing*, 95(6):493–535.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and others (2010). A view of cloud computing. *Communications of the ACM*, 53(4):50–58.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., Stoica, I., and others (2009). Above the clouds: A berkeley view of cloud computing. Technical report, EECS Department, University of California, Berkeley.

AWS (2017). Amazon Web Services. [online]. available from:. https://aws.amazon.com/.

Babar, M. A. and Chauhan, M. A. (2011). A tale of migration to cloud computing for sharing experiences and observations. In *Proceedings of the 2nd international workshop on software engineering for cloud computing*, pages 50–56. ACM.

Binz, T., Leymann, F., and Schumm, D. (2011). CMotion: A Framework for Migration of Applications into and between Clouds. In *Service-Oriented Computing and Applications (SOCA), 2011 IEEE International Conference on*, pages 1–4. IEEE.

Fettke, P. and Loos, P. (2003). Multiperspective evaluation of reference models–towards a framework. *Conceptual modeling for novel application domains*, pages 80–91.

Gartner (2017). Developing a Public Cloud IaaS Adoption and Migration Framework. [online]. available from:. https://www.gartner.com/doc/3645347/developing-public-cloud-iaas-adoption.

Gholami, M. F., Daneshgar, F., Beydoun, G., and Rabhi, F. (2017). Challenges in migrating legacy software systems to the cloud—an empirical study. *Information Systems*, 67:100–113.

Gholami, M. F., Daneshgar, F., Low, G., and Beydoun, G. (2016). Cloud migration process—A survey, evaluation framework, and open challenges. *Journal of Systems and Software*, 120:31–69.

Ionita, A. D. (2013). Introduction to the migration from legacy applications to service provisioning. *Migrating Legacy Applications: Challenges in Service Oriented Architecture and Cloud Computing Environments*, pages 1–11.

ISACA (2012). Guiding principles for cloud computing adoption and use. [online]. available from:. http://www.isaca.org/Knowledge-Center/Research/Pages/Cloud.aspx.

ISO/IEC (2006). Systems and Software Engineering- Software Life Cycle Processes - Maintenance, ISO/IEC 14764:2006.

Jamshidi, P., Ahmad, A., and Pahl, C. (2013). Cloud migration research: a systematic review. *IEEE Transactions on Cloud Computing*, 1(2):142–157.

Kundra, V. (2011). Federal Cloud Computing Strategy. *Washington: The White House*.

Microsoft (2017). Microsoft Azure. [online]. available from:. https://www.azure.com/.

Oracle (2017). Oracle Database. https://www.oracle.com/database/index.html.

Pahl, C. and Xiong, H. (2013). Migration to PaaS clouds-migration process and architectural concerns. In *Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA), 2013 IEEE 7th International Symposium on the*, pages 86–91. IEEE.

PMI (2013). A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Fifth Edition.

Reza Bazi, H., Hassanzadeh, A., and Moeini, A. (2017). A comprehensive framework for cloud computing migration using meta-synthesis approach. *Journal of Systems and Software*, 128:87–105.

Sun, K. and Li, Y. (2013). Effort estimation in cloud migration process. In *Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on*, pages 84–91. IEEE.

Zhao, J.-F. and Zhou, J.-T. (2014). Strategies and methods for cloud migration. *international Journal of Automation and Computing*, 11(2):143–152.