Science Arena Publications
Specialty Journal of Electronic and Computer Sciences
ISSN: 2412-7485
Available online at www.sciarena.com
2019, Vol, 5 (2): 1-16

Generalization of Software Maintenance to Enterprise Architecture Maintenance

Safoura Oveysi[1]*, Fereydoun Shams Aliee[2]
[1] Senior Graduate of Information Technology, Shahid Beheshti University, Tehran, Iran
[2] Associate Professor of Information Technology, Shahid Beheshti University, Tehran, Iran

***Corresponding Author**

Abstract: *Today, one of the biggest challenges faced by Enterprise is strategic alignment of information technology with business; and Enterprise use various methods to achieve this strategic alignment. Enterprise architecture is an effective approach that allows optimal management of Enterprise's information technology and strategic alignment of IT applications and business requirements. But the question is whether only implementation and running of Enterprise architecture resolves this challenge? Considering today's dynamic environments, there is the possibility of any kind of change in Enterprise's conditions, especially after implementation of Enterprise architecture. This change of conditions manifests itself in a variety of ways. In these cases, decision making should be done as appropriate to these changes, and an appropriate response to the changes should be predicted, since otherwise it would lead to lack of proper response and thus readjustment of Enterprise architecture document, which imposes many human and financial costs on the Enterprise. In addition, the Enterprise's progress toward competitive advantages will be stopped. The response and adaptability appropriate to these changes is a concept represented under the title of adaptive maintenance of Enterprise architecture. In the present study, with the help of a software maintenance method and examining the possibility of generalizing it to Enterprise architecture maintenance, through a case study on Power Distribution Company of Golestan province, we show how we can assist improvement of adaptive maintenance of Enterprise architecture and provide the possibility of implementation of various types of changes in business and information technology without disturbing the Enterprise.*

Keywords: *Enterprise Architecture, Strategic Alignment, Enterprise Architecture Maintenance, Adaptability.*

INTRODUCTION

Enterprise architecture has three main phases as follows (Shams and Mahjourian, 2015)

- Strategic planning of information technology
- Planning of Enterprise architectural
- Implementation of Enterprise architecture

Given that implementation of Enterprise architecture is not the last phase of Enterprise architecture, then, after implementation of Enterprise architecture, how can we respond to different types of changes and updates existing in Enterprise architecture, such that the Enterprise will always be up to date? Answering this question and similar questions requires familiarization with a concept called adaptive maintenance which should be implemented in the Enterprise in order for it to have the capability of responding to different types of existing changes. However, the main problem is how this concept can be represented in Enterprise architecture?

One of the best strategies to represent adaptive maintenance in Enterprise architecture is using the help of this concept in similar areas.
Given the age and diversity of software maintenance methods, this area can be selected as the closest area to Enterprise architecture maintenance. The difference between software architecture and Enterprise architecture is only in the scale and life span of their life cycle.
We are looking for a method that, while covering the concept of adaptability in the Enterprise, has the ability to be generalized to Enterprise architecture maintenance. Therefore, in the first step, the problem is whether software maintenance methods can be generalized to Enterprise architecture maintenance? If yes, how much is it? And, in the case of generalizability, how these methods can be used in order to satisfy maintenance capability in Enterprise architecture in the post-implementation phase?

Review of Literature and Research Method

Concept of adaptive maintenance in Enterprise architecture

In Enterprise architecture, there is no resource that studies classification of different types of maintenance in detail. Most of the studies in the area of maintenance have looked at this feature generally, and refrained from addressing details and types of maintenance (Davoudi & Sheikhvand, 2012; Lagerström, 2007). For this reason, in Enterprise architecture, no special definition for adaptive maintenance was found. Therefore, considering the similarities between software architecture and Enterprise architecture which were mentioned before, redefinition of types of software maintenance in the area of Enterprise architecture will be as follows:
With regard to the above, Enterprise architecture maintenance can be classified into three categories (PerOlof Bengtsson, 1999): corrective, adaptive, and perfective, which we will be redefined below in the area of Enterprise architecture:
Corrective maintenance involves an operation that eliminates defective components of Enterprise architecture (replacing new business information systems instead of old systems).
Adaptive maintenance operation involves the operations required to adapt the changes in each of different views of Enterprise architecture (the operation required to adapt to the increase of information systems in Enterprise).
Perfective maintenance involves changes that are organized by user requests and lead to perfection of part or parts of the Enterprise (increasing and restoration of infrastructure layer of the Enterprise).
What is clear from the above definitions is the importance of adaptive maintenance, as in all three types of maintenance, directly and indirectly, the issue of adaptive maintenance has been addressed. In corrective maintenance also after elimination of defective components of Enterprise architecture, in case that there will be no appropriate adaptability to other components in order to find the proper alternative, then the Enterprise's architecture will face problems. In perfective maintenance also after the request by the user, in case that this request causes disorder in other processes or related components, this request will not be done and it requires adaptability of each request with other processes and components of Enterprise architecture, which itself, implies adaptive maintenance. For this reason, addressing adaptive maintenance is the basis for addressing other maintenance types that will be addressed in future researches. Therefore, in this study, focusing on adaptive maintenance, the basis will be provided for implementation of other maintenances in the Enterprise.
The above definitions were redefinitions of types of software maintenance in the area of Enterprise architecture; but in order to make the concept of adaptive maintenance clearer, the definition of this type of maintenance according to software adaptive maintenance in the general scenario can be used which is as described below:
Stimulation: adding, eliminating and changing implementation priority of each component existing in each view of Enterprise architecture

Stimulator: end-user, developer of Enterprise architecture, architect, CIO
Environment: architecture maintenance and support phase
View: business view, data architecture view, application architecture view, technology architecture view
Product: Each of the different views of Enterprise architecture
Possible responses: identification of any part affected by the changes; appropriate adaptability of the sectors engaged in each view of Enterprise architecture with the occurred change; level of satisfaction of Enterprise's ultimate goal from the extent of changes; identification of the amount of effort needed to make the changes and the adaptability appropriate to them.
Response criterion: importance, participation, and satisfactoriness of each view

Works related to software maintenance

Among the software analysis methods, scenario-based methods are methods that through statement of functional states of system and examination of reactions specified in the architecture, try to analyze appropriateness of the architecture and the related qualitative features, including software maintenance.
There are several methods for scenario-based evaluation, each of which refers to different qualitative features. Table 1 shows these methods and their applications better.

Table 1: Software scenario-based evaluation methods

Method	SAAM (2007)	ATAM (2018)	SAAMER	ALMA (2002)	ALPSM (1999)
Maturity stage	Refinement/ Dormant	Refinement	Dormant	Refinement/ Development	-
SA definition	In charge of the user	In charge of the user	In charge of the user	Not provided	-
Process support	Not clearly specified	Fully covered	Having a framework for process support	Incrementally implementing this capability	fully
Method purpose	Risk analysis	Analysis of sensitive and equilibrium points	Evaluation of soft architecture for evolution and capability of being reused	modifiability	Analysis of maintenance capability
Qualitative feature	Capability of change	Diverse	Reuse and evolution	Modification	Maintenance
Tool support	SAAM TOOL	-	-	-	-

The three ALMA[1], SAAM and ATAM[2] methods are methods that address modifiability feature that, according to the proposed qualitative models below, is considered a criterion of maintenance capability; but among these methods, the ALPSM[3] method, focusing on the qualitative feature of maintenance capability, is most consistent with the purposes of this research. So, through getting more familiar with this method, we try to generalize this method in Enterprise architecture.
Given that these methods are specific to evaluation of software architecture, therefore, the techniques used in these methods are mostly unique to the software systems layer, and there is need for more studies beyond the scope of this study in order to determine whether they can be used in the Enterprise too.

- **ALPSM method (Bengtsson, 2002)**

ALPSM, due to its successful results, estimation of the effort required for repair and maintenance in the system in accordance with the built architecture, and estimation of the system's compatibility with scenarios based on prediction of maintenance capability, is considered a separate evaluation method and has several stages.

[1] Architecture-level analysis of modifiability
[2] Architecture tradeoff analysis method
[3] Architecture Level Prediction of Software Maintenance

This method takes the quadruple information including 1) features of requirements, 2) architectural design, 3) software engineers' expertise, and 4) historical data, as the inputs and provides a prediction of the average effort to maintain and repair the system. Through this method the scenarios for identification of maintenance requirements are aggregated and it helps architectural analysis for prediction of maintenance
This method is consisted of a series of change scenarios which are in fact a set of probable changes that are highly likely to occur and cause change in hardware or software. This method includes six essential steps as below:

1. Identification of the examined branch for maintenance
2. Selection of scenarios
3. Evaluation of the weight of each scenario
4. Estimation of the size of the elements
5. Scripting scenarios
6. Calculation of the predicted effort for repairing and maintenance

What was discussed above is in fact a comprehensive method for expressing software architecture maintenance. Although this method has merely been implemented for software architecture, however, given the proximity of its concepts to Enterprise architecture, it can be generalized to that too. In the proposed solution, ALPSM method will only be used for introduction and advancement of the research purposes so that in this way, while calculating the effort required for maintenance in Enterprise architecture, our main objective which is to improve adaptive maintenance in Enterprise architecture will also be obtained.

Previous research

- **Analysis of modifiability of Enterprise architecture (Busch & Zalewski, 2018)**

Since modifiability is considered an obvious feature of adaptive maintenance, its analysis can play a very important role in guiding the method toward adaptive maintenance. In this study, software architecture evaluation methods have been used and it shows how these methods can help to analyze modifiability feature of Enterprise architecture. This method is a generalization of software architecture analysis method (SAAM). In this method, according to SAAM, firstly, a set of probable scenarios that the Enterprise may face in future are shared and then these scenarios are analyzed and prioritized. At this point, it uses BPMN in order to express and display business scenarios. Then, in order to analyze the data, through ArchiMate language and other languages such as UML, it models other components such as data layers, information systems, and infrastructure. The purpose of this phase is to design the architecture such that it will be understandable for all stakeholders. Then it prioritizes direct and indirect scenarios and ultimately, all types of existing changes in these scenarios and the involved interactions are identified. Given the involved components, the amount of required cost for modifiability per person per day is calculated based on the opinions of Enterprise experts.
In this method, different modeling languages have been used to model the components of different layers. In addition, SAAM method used to generalize this method has not been used only for modifiability, but also has addressed other features too. This method shows that software architecture evaluation methods can be generalized to Enterprise architecture and therefore, it has an important role in proving the proposed method. An example of this modeling has been shown in Figure 1. This modeling does not specify the extent of the influence of change on other components of other layers. Moreover, adaptability method has not been shown in this method and only the type of scenario and the affected components have been specified.

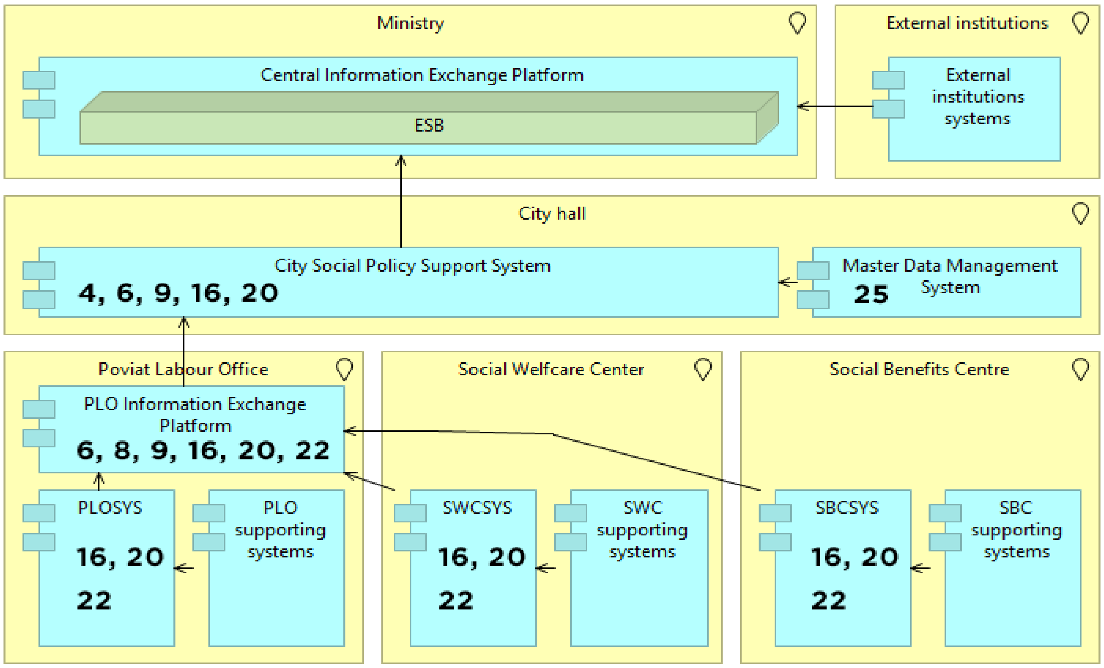

Figure 1.: Modeling modifiability using ArchiMate language and specifying the components involved in each change scenario

- **A new method based on genetic algorithm and ATAM to prioritize Enterprise architecture scenarios (case study: Enterprise architecture maintenance) (Kazman et al., 2000)**

This method is one of the few methods that have used software evaluation methods in Enterprise architecture. In this study, ATAM has been used in order to analyze Enterprise architecture. This method uses different qualitative scenarios so that in this way, while identifying different decisions and compromising between them, will determine whether or not these decisions, given the architectural structure, are consistent with other qualitative features?

After prioritizing scenarios by ATAM, an appropriate scenario is found among the prioritized scenarios through genetic algorithm. This method examines maintenance and interactivity with the help of ATAM method and genetic algorithm.

In this method, firstly, the indicators of maintenance are prioritized by the stakeholders (Razavi, 2009); then the change scenarios for maintenance and interactivity are specified and prioritized by the stakeholders. After this stage, the degree of influence of each scenario on the qualitative feature is determined through quantitative values, and finally, these prioritized values are injected as inputs into genetic algorithm (Karimi et al., 2014), and after implementation of genetic algorithm in MATLAB software, the scenarios with the highest priority and the most appropriate scenario are selected as the desired change scenario, so that in this way, the cost and time of Enterprise maintenance will be reduced.

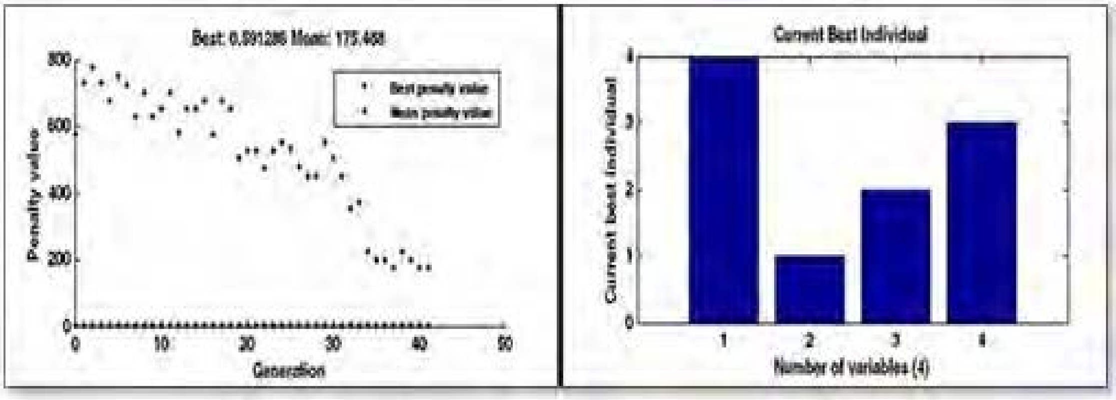

Figure 2.: Display and selection of the scenario appropriate for change in MATLAB tool with the help of genetic algorithm

This method, despite being a very new idea in Enterprise architecture and reducing complexity of implementation time, has not proposed anything about the affected components and how to deal with these change scenarios.

Challenges

All of the methods mentioned in the previous section examine Enterprise architecture maintenance. In most of these methods, change scenarios have been considered as the input of the proposed method, which is itself the main incentive of the present research in order to use scenario-based methods. But the problem which can be seen in most of these methods, despite very high importance of business layer, is very little attention to this layer, since these researches have mostly focused on information systems layer rather than business layer. In addition, in these methods special attention has been made to selecting appropriate scenario for Enterprise architecture maintenance, but little attention has been paid to how to apply this scenario, identification of involved components, the amount of influence on other components, and adaptability method. In this study we are seeking a strategy through which, while selecting the appropriate scenario for maintenance, the other existing challenges can also be removed.

The Proposed Method

The proposed method, in addition to providing a solution to improve adaptive maintenance, calculates the effort required for adaptive maintenance in Enterprise architecture. It should be noted that given the wide range of Enterprise architecture and impossibility of examining all its layers in this study, only two important and challenging layers of business and information systems are examined. Also, the data layer which is located between these two important layers will be examined as Enterprise decisions, and examination of other layers will be part of the proposed works. This method has been expressed by generalization of software maintenance and ALPSM method. The proposed method has largely been generalized from ALPSM method and is consisted of four stages that the first three ones address improvement of adaptive maintenance of Enterprise architecture and the last stage estimates the effort required to maintain Enterprise architecture. Figure 3 shows the stages of the proposed method, along with its inputs and outputs.

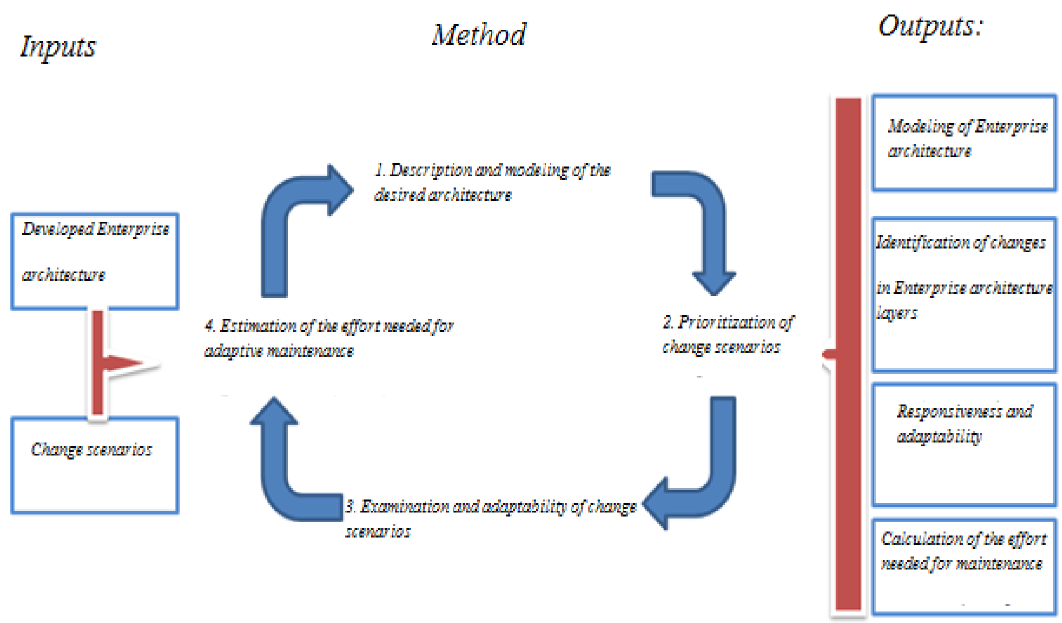

Figure 3: Proposed method for improving adaptive maintenance in Enterprise architecture

[*Inputs:*
Developed Enterprise architecture
Change scenarios
--
Method:
1. Description and modeling of the desired architecture
2. Prioritization of change scenarios
3. Examination and adaptability of change scenarios
4. Estimation of the effort needed for adaptive maintenance
--
Outputs:
Modeling of Enterprise architecture
Identification of changes in Enterprise architecture layers
Responsiveness and adaptability
Calculation of the effort needed for maintenance

- **Description and modeling of the desired Enterprise architecture**
    - **Description of the desired Enterprise architecture**

At this stage, we will select and determine the layers of Enterprise architecture in which improvement of adaptive maintenance is going to be done.

- **Modeling of the described Enterprise architecture**

In the second part of the first stage, the Enterprise architecture described in the previous part is modeled. The modeling language of URN and JUCMNAV tool are used for modeling (Hasan Salim Alkaf et al., 2017). This modeling has been implemented down to top. The changes occur in lower layers, and then the impact of changes is transferred to higher layers of Enterprise architecture.

The advantage of using this modeling is using three very important parameters of contribution level, importance level, and satisfaction level of the elements used in the model, which is considered the strength of the proposed method.
According to Enterprise experts, these three parameters are in fact considered three key parameters for adaptability; and after any change and the adaptability appropriate to that, these three are among the most used parameters that must be examined.
The contribution level parameter is based on collaborative links between the internal modeled elements and shows the amount of contribution of the elements in the lower layer to the elements in the higher layer in order to achieve main objective of the Enterprise.
The importance level parameter is located inside each element and expresses the importance of each element among other elements involved in the same layer.
The satisfaction level parameter shows level of satisfaction of the high level layer element in the face of changes applied to low level layer elements.
In order to calculate the first two parameters, namely, the contribution level and the importance level of each element, AHP method is used, whose inputs (prioritization of elements of each layer) are obtained through Enterprise experts.
In order to calculate the third parameter, namely satisfaction level, bottom up propagation algorithm is used which has been written by Amyot et al. (2010). If the element X is located in the layer higher than A, B, C elements with the contribution level of a, b, c and satisfaction level of d, e, f, then, satisfaction level of the element X is obtained using Formula One:

$$\text{Satisfaction level of element X} = (a * d) + (b * e) + (c * f) \qquad (1)$$

In general, in this section, firstly each layer of Enterprise architecture described in the first section is modeled separately according to the above points. Then, the layers are linked to each other through cooperative links, and finally, correctness of this relationship is examined with the help of adaptation rules.
After automatically examining the rules, the desired algorithm is executed so that satisfaction level of the elements in the higher layers, and ultimately, the Enterprise's goal, in the conducted modeling, will be determined.

- **Prioritization of change scenarios**

At this stage, various types of changes that may occur in the Enterprise architecture layers are examined in the form of change scenarios. These changes are categorized into three categories of simple, incremental and decreasing changes:
Simple changes are the changes that affect implementation priority of Enterprise architecture elements, goals, functions, and information systems.
Incremental changes are changes that lead to addition of an Enterprise element, goals, functions, and information systems in the desired layers.
The decreasing changes are changes that are done as the result of elimination and reduction of Enterprise architecture elements, goals, functions, and information systems in the desired layers.
At this stage, different change scenarios in the desired layers are examined and, similar to ALPSM method, they are prioritized based on the opinions of Enterprise experts. Determining these scenarios is the responsibility of the Enterprise's architect.

- **Examination and adaptability of change scenarios**

At this stage, which is considered the most basic stage in this method, we will examine how the model reacts and adapts to the applied changes?
As previously stated, this adaptability will be done in the face of simple, incremental, and decreasing changes in the desired layers which are business layer and information systems in this research.

- **Estimation of the effort required for adaptive maintenance**

At this stage, with the help of the models built before and after the changes, the amount of effort required for adaptive maintenance can be obtained. In order to achieve this very important formula, the total number of people involved in the Enterprise must be determined. In the model implemented in this research in the case study of section 5, individuals or actors are in fact parts of the three layers of information systems, business, and decision-making that contribute to achieving Enterprise goal.

In this research, the parameters determined for each actor (part) and indicated in the desired modeling include the level of satisfaction from each actor based on satisfaction from internal elements of each actor; however, in order to obtain the effort required for adaptive maintenance based on person/day, the very important parameter based on which the individuals' allocation is made is importance level parameter. According to a survey conducted on the Enterprise experts, the importance level parameter with 49% was selected as the highest factor in allocation of forces to each sector, and the parameters of contribution level and satisfaction level with 30% and 21% were placed in the second and third ranks, respectively. According to the opinion of Enterprise experts and the Enterprise's architect, in order to allocate people to each unit, the importance level of each unit plays a very significant role in allocation of individuals, because the more important each unit is, the more employees it requires; while, according to its importance, it has more contribution, directly or indirectly, in achieving Enterprise goal, while the level of satisfaction from the unit has a lower relation to the employees allocated to that unit.

For this reason, in this section we need to know the importance level of each actor (part), which this is also obtained through binary AHP comparison.

Firstly, the number of people required for each actor (part) is determined through the following formula:

$$\text{N (actor)} = \text{p(actor)} * \text{Ntotal} \qquad (2)$$

P (actor): Importance level of each actor

Each actor has a number of internal elements that are shown in the form of goal, information system, or decision (tasks). In order to obtain the individuals involved in each internal element, the following formula is used:

$$\text{N (initial element)} = \text{N (actor)} * \text{p (initial element)} \qquad (3)$$

These formulas help us obtain the number of people involved in each unit and each goal or information system or each decision making. After the changes are made, these formulas are re-calculated and the number of people based on each change is obtained; Finally, the following formula is used in order to obtain the effort required for adaptive maintenance:

$$\text{E(EAAM)} = \text{N(ChangeElement)} + \sum_{I=1}^{\text{Ntotal}-\text{N(changecomponent)}} \text{N(The effectedElements)} \qquad (4)$$

E (EAAM): The effort required for adaptive maintenance of Enterprise architecture per person per day

N (ChangeElement): The number of people involved in changing the desired element

N (The effectedElements): The number of people involved in the elements affected by the created change

Evaluation of the Proposed Method

The proposed method was presented with the subject of providing a method for improving adaptive maintenance of Enterprise architecture. There are several methods for evaluation, and each of these methods is used for a specific application. Among these methods, the most appropriate evaluation solution for the proposed method is using of a real sample. One of the most effective ways in order to indicate improvement in adaptive maintenance of Enterprise architecture is to compare the real sample with the sample implemented with the help of the proposed method; and case study allows for access to this method. This case study, as Enterprise architecture has been fully developed and implemented in that, is considered a very good case for this research. This company due to having a dynamic environment is very suitable for examination of adaptive maintenance. Enterprise architecture documentations for Power Distribution Company of Golestan province have been provided by Golestan Software Management Group and it includes the architecture of the existing situation and the architecture of the favorable situation. In the following, the steps of the proposed method will be evaluated:

- Description and modeling of the desired architecture

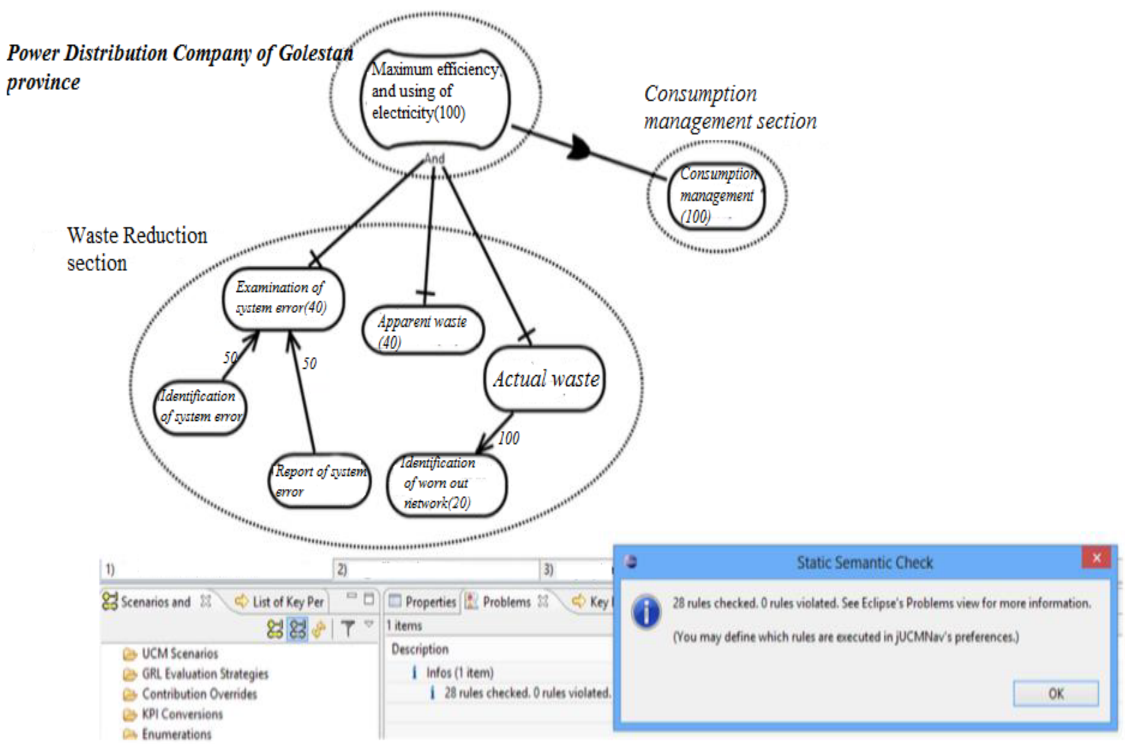

Figure 4: Examination of correctness of the levels implemented by Power Distribution Company of Golestan province

[**Power Distribution Company of Golestan province:**
Maximum efficiency and using of electricity
Consumption management section:
Consumption management
--
Waste Reduction section:
Examination of system error/ Apparent waste/ Actual waste
Identification of system error/ Report of system error/ Identification of worn out network]

The implemented model is a bottom-up model and satisfaction levels of the modeled structure have been developed with the help of bottom-up algorithm. The initial valuation of information system indicators is obtained by specifying the 4 values of goal, the mean, the worst, and the assessment mode that has been valued by the experts of Power Company. For example, indicators of PM system have been examined and evaluated as a sample (Table 2). Then, these indicators, based on the evaluated value, are implemented by the developed algorithm in order to obtain satisfaction levels of the desired indicator. These indicators, with the help of the formula proposed in the desired algorithm, transfer satisfaction levels to the level of information systems and then to higher levels

Table 2: Valuation of PM information system indicators

	Goal	Mean	The worst	Evaluation
Relation with other units	90	70	50	72.5
Information quality	90	70	50	80
Service quality	90	70	50	90
Encapsulation level	90	70	50	65
Satisfaction of the user from the system	90	70	50	80
Cost	90	70	50	75

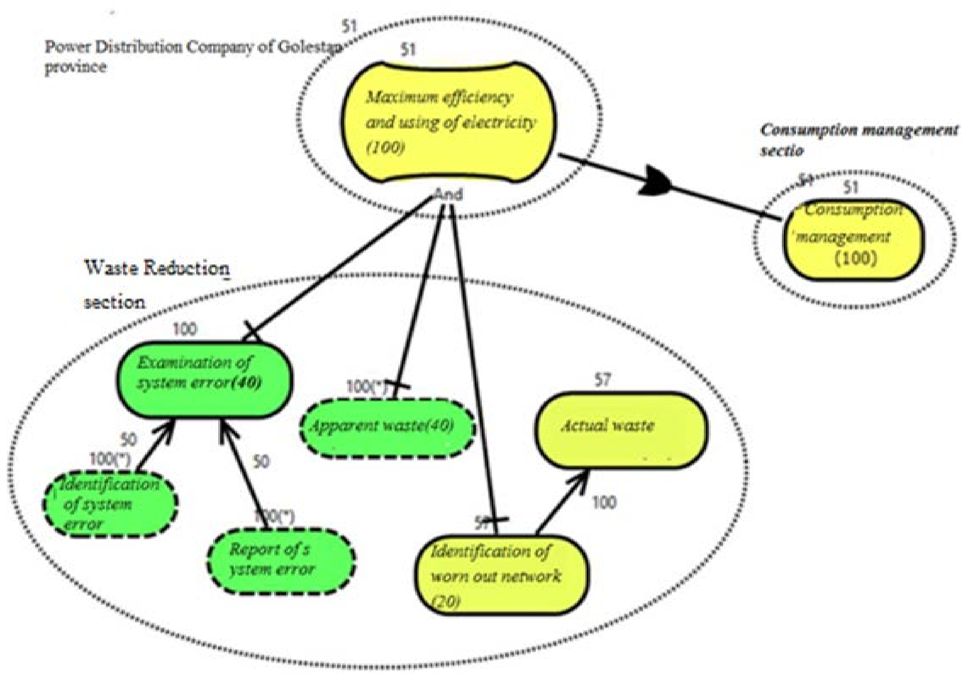

Figure 5: Implementation of the algorithm on business level in order to obtain satisfaction levels

- Prioritization of change scenarios

Change scenarios include three types of deletions, additions, and simple changes to the implementation priorities of internal elements and links. In order to select scenarios, we need to implement scenarios that have actually been implemented in this company so that when calculating adaptive maintenance of Enterprise architecture of this company, we can make an acceptable comparison between the actual amount of maintenance and the maintenance value obtained by the proposed method. One of the scenarios that had occurred in this company and had failed due to lack of sufficient understanding of adaptability issue was removing GIS information system and integrating it with System 121. Despite the fact that this scenario could have had positive impacts on the Enterprise, it has failed due to lack of appropriate adaptability and lack of adequate understanding of the positive effects of this change on other levels.

- Examination and adaptability of change scenarios
- ✓ Removing GIS information system

GIS system is one of the most important systems used in Power Distribution Company.

With a brief looking at GIS capabilities, it is seen that one of the mostly used GIS capabilities is identifying worn-out networks, fixing network defects, and easier decision making about network which plays a significant role in preventing power losses.
After removing GIS information system, with the help of AHP paired comparison, the model will be adapted and the effect of this change is applied to the higher levels.

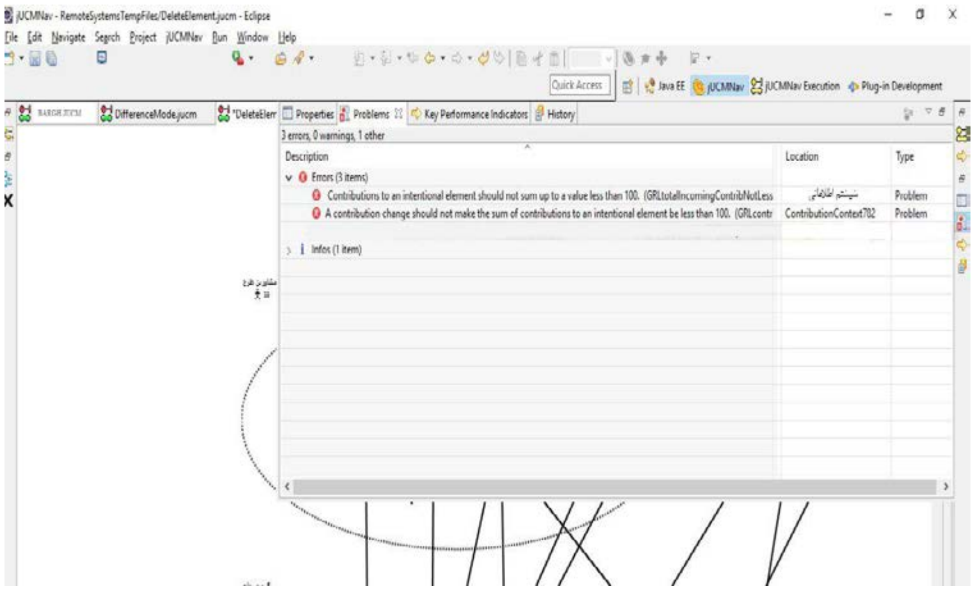

Figure 6: How to fix the errors occurred due to change

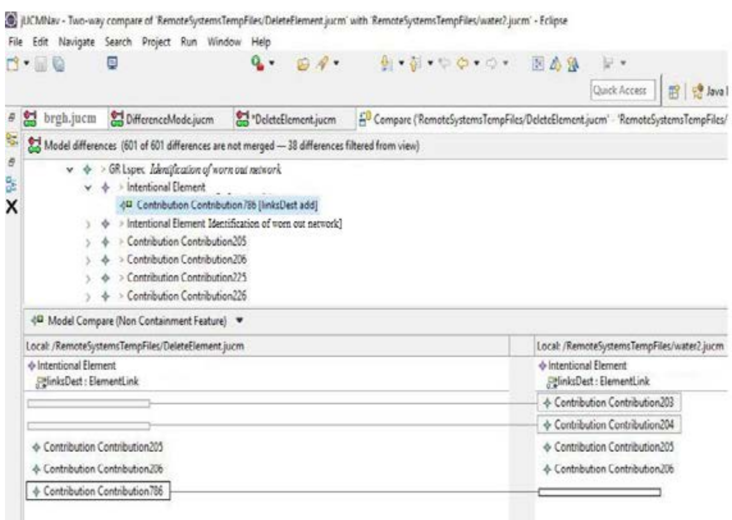

Figure 7: Showing the elements of other levels before and after removal of the GIS information system

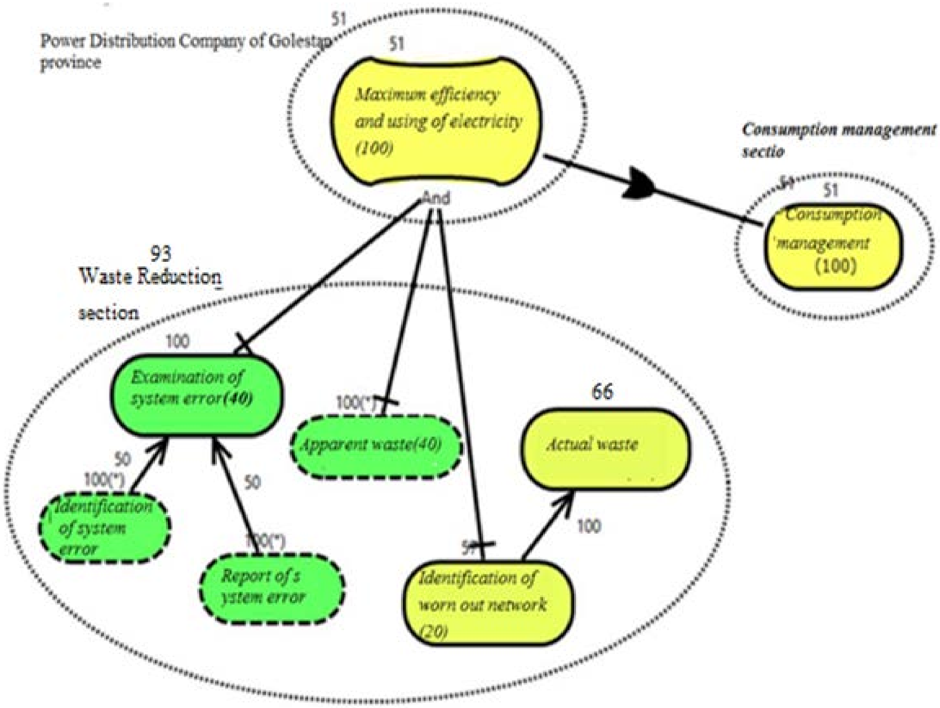

Figure 8: The impact of removal of GIS information system on business level

By removing this information system and integrating it with System 121, we will have the amount of satisfaction level increased on other levels which can be seen in the Figure.

✓ Calculation of adaptive maintenance of change scenarios

In order to calculate adaptive maintenance, firstly the number of people involved in each section must be identified. The total staff involved in this target are 145 people who must be distributed among various actors (units or sectors including goals, tasks, or information systems), and each actor, based on higher importance level, allocates more people to itself.

Actors who are involved in this goal include project consultants, waste reduction section, subscribers and branch section, utilities section, consumption management section, information systems section and technical engineering section. Using AHP method, the importance level of each actor is determined, and then the number of people allocated to that is specified.

With the help of the obtained figures, given that the total number of actors is 145 people, the number of individuals related to each actor is determined.

Now, using the number of people obtained for each actor, the number of people involved in each target, sub-target, task, and information system are determined.

Now, having detailed information about the number of actors in each target, activity and task, the amount of adaptive maintenance is calculated per person per day.

$$\mathrm{E(EAAM)} = \mathrm{N(ChangeElement)} + \sum_{\mathrm{I}=1}^{\mathrm{Ntotal-N(changecomponent)}} \mathrm{N(The\ effectedElements)}$$

According to this formula, firstly the number of people involved in the considered change which is removal of GIS information system must be identified, that the number of these individuals is limited to 6 people. The number of people involved in sectors affected by this change is obtained.

According to the mentioned interpretations and Formula One, adaptive maintenance of Power Distribution Company of Golestan province in case of removal of GIS information system per person per day is: 74 + 6 = 80 person per day.

While according to the information obtained from Power Distribution Company, before implementation of the proposed method, due to this change in the company, the amount of adaptive maintenance of this company had reached 113 person per day, that due to lack of sufficient knowledge about the affected elements and lack of appropriate adaptability, about 33 people had been used inappropriately and extremely compared to actual maintenance of the Enterprise; that this is a significant amount.

Qualitative evaluation of the proposed method

Since the methods proposed in the area of Enterprise architecture maintenance in fact predict this qualitative feature, according to previous researches, four criteria of accuracy, cost of using the model, cognitive complexity and error control can be considered for evaluation of prediction models.

The models built for other qualitative features have been examined in terms of these four criteria. In this source, the following seven characteristics have been described in order to meet these four criteria:

1. Formal prediction theory
2. Automatic prediction
3. The rules set for the considered qualitative feature
4. Clarity and fluency of the formal method used
5. The reusable prediction theory
6. Prediction based on probabilities
7. Model abstraction

The first feature results in higher accuracy of the model. Since our model rules are normalized as OCL, there is no possibility of misinterpretation, and this helps higher accuracy of the model. On the other hand, these formal rules can be used for implementing the method through tools, that this reduces the cost of using the method and human error and results in higher accuracy in prediction. In addition, the model has been implemented completely separate and leveled. This leads to easy use of the model and reduction of its learning curve, which will ultimately lead to positive impact on cognitive complexity of the model. In order to increase accuracy, the formal rules used in this model are needed to be sufficiently clear and fluent and can describe the problem space clearly. For example, in the area of maintenance, there are many concerns about examination of adaptability. For this purpose, the used formal method must allow description of adaptive architectural structure.

Using of OCL language, by expressing the rules, allows for query about the model and use of logical calculations and operators which provide the analyst with sufficient power. It also reduces the cost of using the model due to not needing a specialist in the field and reduced its cognitive complexity, because the analyst only needs to focus on modeling and does not need to define relationships in each scenario, and the formal rules have this responsibility. Generally, the following four criteria are shown in this method as follows:

Accuracy: Accuracy of the proposed method depends on data collection method. If the analyst determines the relationships between the levels of Enterprise architecture and model elements based on his own judgment, this accuracy will be low, and if data collection methods based on literature review, data mining, or expert surveys are used, more accurate results can be expected. So, in this method, due to the use of research literature, views of Enterprise experts and the use of OCL adaptable rules, we have a higher accuracy.

Cost: The use of the proposed method, due to its lack of complexity, does not require a very high cost; in addition, this method, after one time of implementation, provides the analysts with the reusable results.

Cognitive complexity: The use of the proposed method, due to determining clear and specified steps, is estimated to be relatively simple. Since this method has been generally presented, it can also be used for adaptability of other qualitative features and the learning chart can be expected to be short.

Error control: In the compromise between accuracy and cost, in the information collection phase the researcher can use available resources, such as personal opinion, subject literature, data mining, and expert survey methods. It is clear that accuracy of the model will be higher based on the amount of effort spent on collection of basic information (cost).

Discussion and Conclusion

Answer to the first question:

The first step in ALPSM method selects the maintenance branch, that in Enterprise architecture maintenance method, adaptive maintenance branch was selected. In the second and third stages, the scenarios are selected and weighed in order for prioritization, that in the proposed method, through survey of the Enterprise experts, in the second stage, the scenarios are prioritized and scenarios with the highest priority are selected. In the fourth stage, the amount of the component which is in fact the amount of effort for changing the desired component is estimated, that in the proposed method, using importance level parameter and determining the people specific to each Enterprise component at the scenario examination and adaptability stage, this amount will be obtained. Also, in the fifth stage, ALPSM scripts the proposed scenarios so that in this way the involved components and the impact of the change on other components can be determined, that in the proposed method, using the parameters of contribution level, importance level, satisfaction level, and developed algorithm and adaptation rules, the components affected in the related layers and the amount this effect are easily determined, and finally, in ALPSM method, the effort required for maintenance is estimated, that given the size of Enterprise architecture compared to software architecture and differences in some calculations, the effort required for Enterprise architecture maintenance is obtained given the specified parameters.

Answer to the second question:

The time of adaptive maintenance, due to using of OCL rules and AHP method, is significantly reduced. In order to calculate maintenance cost, with the help of importance level parameter, the number of people allocated to each actor and the internal elements of each actor are identified and the amount of effort required to apply changes and the adaptability appropriate to the desired change were calculated.

Recommendations for future research

1. As was noted in this study, this research despite the possibility of covering all layers only examines the two layers of information systems and business systems that future studies can investigate other layers.
2. Adaptability rules in this study are specific to the two layers of business and information systems that one of the other things that can be done in future is to write general adaptable rules that can be applied to all Enterprise architecture layers.
3. This method only shows the impact of bottom-up changes. In future studies, using an algorithm that can show the impact of changes in both bottom-up and up-bottom sides can play a significant role in advancement and development of this research.
4. The algorithm considered in this study has been valued based on contribution level and satisfaction level parameters. If future researches can find a formula for more effective implementation of this algorithm, such that it includes all the three provided parameters, it can play an effective role in more accurate understanding of the impact of changes on other elements.

References

1. Alkaf, H. S., Hassine, J., Hamou-Lhadj, A., & Alawneh, L. (2017, October). An automated change impact analysis approach to GRL models. In International SDL Forum (pp. 157-172). Springer, Cham.
2. Bengtsson, P., (2002). Architecture level prediction of software maintenance.
3. Bengtsson, P., Lassing, N., Bosch, J., & van Vliet, H. (2004). Architecture-level modifiability analysis (ALMA). Journal of Systems and Software, 69(1-2), 129-147.
4. Busch, N. R., & Zalewski, A. (2018). Enterprise Architecture Modifiability Analysis. In Towards a Synergistic Combination of Research and Practice in Software Engineering (pp. 119-134). Springer, Cham.
5. Davoudi, M. R., & Sheikhvand, K. (2012). An approach towards enterprise architecture analysis using AHP and fuzzy AHP. International Journal of Machine Learning and Computing, 2(1), 46.
6. Karimi, M., Sharafi, S. M., & Dehkordi, M. N. (2014). A new approach based on genetic algorithm for prioritizing quality scenarios in enterprise architecture evaluation. International Journal of Computer Science Engineering (IJCSE), 3(01), 21-31.
7. Kazman, R., Bass, L., Abowd, G., & Webb, M. (1994, May). SAAM: A method for analyzing the properties of software architectures. In Proceedings of 16th International Conference on Software Engineering (pp. 81-90). IEEE.
8. *Kazman, R., Klein, M., & Clements, P. (2000). ATAM: Method for architecture evaluation (No. CMU/SEI-2000-TR-004). Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst.*
9. Lagerström, R. (2007). Analyzing system maintainability using enterprise architecture models. In Workshop on Trends in Enterprise Architecture Research (TEAR 2007), St. Gallen, Switzerland, June 6, 2007 (pp. 31-39). Telematica Instituut.
10. PerOlof Bengtsson, (1999). Architecture Level Prediction of Software Maintenance, Department of Software Engineering aComputerScience University of Karlskrona/Ronneby S-372 25 Ronneby Sweden.
11. Razavi M., (2009). Presentation of a new framework for analysis of qualitative features of Enterprise architecture, PhD thesis, Tehran University of Science and Technology.
12. Shams, Fereydoun, Mahjourian, Amir, (2015). Introduction to the principles and basics and architectural methods of service-oriented Enterprise, Shahid Beheshti University, Tehran.