Hybrid-Based Maintainability Impact Analysis for Evolving Systems

Samer I. Mohamed (1), Islam A. M. ElMaddah (2) and Ayman M. Wahba (3)

(1) Department of Computer and Systems.Ain Shams University (Egypt)
E-mail: samer.mohamed@eds.com

(2) Department of Computer and Systems.Ain Shams University (Egypt)
E-mail: islam_elmaddah@yahoo.co.uk

(3) Department of Computer and Systems.Ain Shams University (Egypt)
E-mail: ayman.wahba@gmail.com

ABSTRACT

Software maintenance becomes an integral part of software life cycle and constitutes the most important fraction of the total cost of the software lifecycle. Around 50-80 percent of the total lifecycle cost is consumed by maintenance for evolving system. Thus systems with poor maintainability are difficult to modify and require more cost to maintain. This difficulty arises from the impact of the system components where the new requirements/goals will be implemented. These new goals will result in modification of existing components and creation of new components. In this paper, we present the foundation for a new Hybrid-Based Maintainability Impact Analysis (HBMIA) methodology for assessing the impact of the new goals to be selected for implementation on the system new and existing components. (HBMIA) uses not only the system history but it also gets benefit from the expert's experience. (HBMIA) balances between the system historical data and experts data based on the organization' maturity and experts experience for system components. A case study is performed to demonstrate the added value of the proposed (HBMIA).

Keywords: Software maintenance – Difficulty of Modification – Impact Analysis.

1- INTRODUCTION

According to IEEE standard software maintenance defined as follows, "Software maintenance is the process of modifying a software system or component after delivery to correct faults, improve performances, prevent problems or adapt to a changed environment" [13]. This definition covers the different types of software maintenance like corrective, perfective, preventive, and adaptive maintenance [19, 14]. Adaptive maintenance involves the modifications to the software system required by changes in the software operating environment [17]. Perfective maintenance refers to the changes originate from the new user requirements. Corrective maintenance includes all the changes required to fix any faults or bugs in the system [27]. Preventive maintenance focuses on preventing problems in the future [4]. The definition also reflects the common view that software maintenance is a post-delivery activity that starts when the system is released and encom-

passes all the activities that keep the operational [1, 23].
Our proposed (HBMIA), targeting perfectly the perfective maintenance in which the system maintenance is done through new requirements requested from the users to improve the total system performance. But (HBMIA) can be also used for corrective, preventive and adaptive maintenance context. Through which the proposed methodology will apply the changes requested from the users as new requirements/goals to correct bugs, prevent problems and adapt to new working environment.
Impact analysis is the activity of assessing the potential effects of a change with the aim of minimizing unexpected side effects [40]. It also involves the identification of the system's components that need to be modified or created as a consequence of the proposed modification [3, 30]. Impact analysis has a great benefit for reducing the risks and unexpected outcomes from the system before implementing the changes. Impact analysis information also can be used in planning different project activities like resource estimation, schedule and cost allocation. This information also can be used to reduce the rework cost and result in higher quality [28, 29].
The organization of the paper is as follows. In the next section, we will refer to the related work for our research. In section three we will elaborate the rationale and research objectives for the proposed methodology (HBMIA). In section four, we will discuss the (HBMIA) methodology specifications. Section five will illustrate the practical advantages from the (HBMIA) through a case study. In section six, we will validate the practical benefits from new (HBMIA) methodology through a comparison versus other recent techniques. The final section summarizes our conclusions and introduces our future research.

2- RELATED WORK

2-1 Difficulty of Modification (DoM)

Difficulty of Modification (DoM) is used for assessing the impact of the existing components where the new goals to be implemented. (DoM) acts as measure for how the existing components in the system to be maintained will be impacted by the new goals to be implemented [15]. There is a set of factors affecting (DoM) measurement. These factors are assessed based on the lower level criteria that can be directly measured from the historical data available in the organization metrics [6]. The set of factors that will subject to our analysis and contribute to (DoM) of the existing components are size, complexity, health, understandability and functionality as shown in fig. 1. These factors are not assumed to be necessary orthogonal. Size refers to that factor which measures the ration of the added/modified code to the total component size. The most common metric used for assessing the component size is the Source Line Of Code or (SLOC) [10]. Complexity identifies that factor which measures the code complexity [39]. The component complexity is affected by the relations between the components themselves in the systems which are measured by the coupling between components. The most common metric used to assess component complexity is the McCabe's Cyclomatic complexity [36]. Health refers to that factor which measures the operational failure reported against the component during field usage of the system. Unhealthy components will result in a high risk for any small modification [5]. The health for any components can be calculated as the ration between the numbers of defects against these components to the total number of defects affecting system for any specific period [11]. Understandability refers to that factor measures the ease with which component can be understood by the developer who modifying it. This will be function of the expertise of whoever is making the change [16], how long this component is part of the system [32], and quality of the documentation. The most common metric used to assess the component understandability is the Halstead level [11]. Functionality identifies how much functionality implemented per each component. The most common metric used to assess functionality is Weighted Method per Class (WMC) or can be calculated by the ration of the added/modified functions per component to the total number of functions within this component.

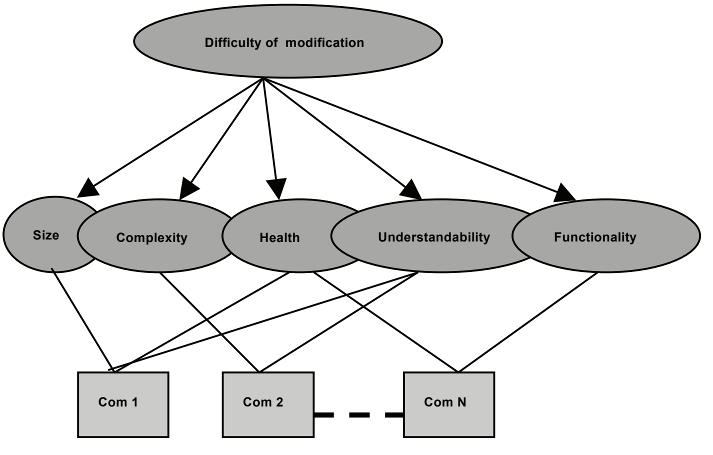

Figure 1 Difficulty of modification factors

2-2 Difficulty of Creation (DoC)

Difficulty of Creation (DoC) is used for assessing the impact of the new components where the new goals to be implemented. (DoC) acts as a measure for how many of the new components in the system to be maintained will be impacted by the new goals to be implemented. The groups of factors that contribute to (DoC) assessment are size, complexity, criticality, understandability and dependability as shown in fig. 2. These factors as indicated in (DoM) factors are not assumed to be orthogonal, or by other means they may affect each other [31]. Size factor can be calculated as the ration between the new component size to the total size of the new implemented components within a specific period of time. The most common metric used for assessing the component size is the Source Line Of Code or (SLOC) [26]. Complexity of the new component will be similar to that of the existing components which indicated in the previous section. Criticality refers to that factor measures how critical the component. Understandability of the new components will be similar to that of the existing component as indicated in the previous section. Dependability identifies that factor which measures the relation between the new component and other system components. The most common metric used to assess Dependability is coupling between components [38].

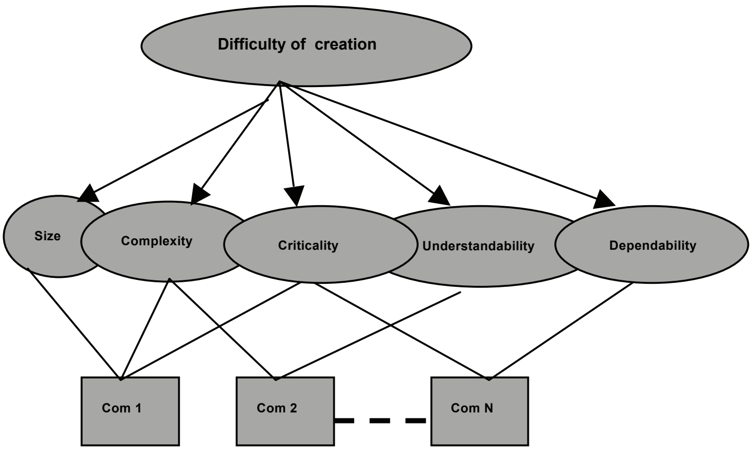

Figure 2 Difficulty of Creation factors

2-3 Metric-Based approach

Metric-based approach refers to that approach which gets benefit from the system historical data to measure to which extent each component exhibit when subjected to modification [22]. This measure will be used later to assess the impact of the new goals on the components where these goals will be implemented. The ability to adopt this approach depends on the organization maturity of collecting relevant historical data over the product life cycle [40]. The main advantage of this approach is that it gets benefit from the component history to identify the future trends of the component when subject to modification. The main drawbacks of this approach arise from its dependency on the organization maturity [21].

2-4 Expert-Based approach

Expert-based approach identifies that approach which uses the knowledge and experience of experts who are familiar with the system components under consideration [16]. The necessity to use this approach will depend on the availability of experts within the organization. The main advantage of this approach over the metric approach is that it can be used while there are problems of data availability [20]. But the main drawback of this approach arises because it's highly driven by the data entered by the experts even when there is no evidence behind it. This approach will be perfectly used to measure the Difficulty of Creation (DoC) because there is no history available for new system components result from the new goals to be implemented [18].

3- RATIONALE AND RESEARCH OBJECTIVES

3-1 Rationale

The rationale behind the (HBMIA) is to develop a new hybrid-based methodology that combines both metric-based and expert-based approaches to gain the advantages of both. Our contribution in the paper is two fold; first, develop a new hybrid-based methodology that combines both expert-based and metric-based approaches to evaluate the Difficulty of Modification (DoM) of exiting components. Second, use the expert-based approach to evaluate the Difficulty of Creation (DoC) of new components imposed by the new goals. The evaluation from both (DoM) and (DoC) will be combined with Goals-Driven Impact Analysis (GDIA) to asses the impact of goals selected for implementation on the system components [40].

3-2 Research objectives

1-Design a characterization methodology for the system components that can be used to assess the components based on quality attributes such as Difficulty of Modification (DoM) for existing components and Difficulty of Creation (DoC) for new components. This characterization methodology identifies the different factors that affect both DoM and (DoC) and gets the proper system metrics that can be used for the assessment of each factor.
2- Develop a methodology to incorporate both system metrics and experts contributions together for each factor of the quality attributes.
3-Imeplement a technique to identify those components that would be impacted by the implementation of the proposed goals. This technique will be called Goals-Driven Impact Analysis (GDIA). (GDIA) will determine how the implementation of each goal would impact system components.
4-Use both the characterization methodology and (GDIA) to calculate the total goal impact from the components where the goal to be implemented.
5-Design a prototype that uses the calculated goal impacts to identifies the best goals to be selected for implementation.

4- HBMIA TECHNIQUE DETAILS

4-1 HBMIA architecture

We will demonstrate the different steps required to calculate the impact on the system components where the new goals to be implemented as in fig. 3:

- Step 1: Use historical data repository of the system within the organization to identify all the system components.
- Step2: Determine the different factors that affect Difficulty of Modification (DoM) as shown in fig. 1.
- Step3: Determine the different factors that affect Difficulty of Modification (DoC) as shown in fig. 2.
- Step4: Get the experts contributions for each component factor of both (DoM) and (DoC). Each expert will be assigned weight according to his/her

experience with the component. Experts also will be assigned relative weights based on their importance within the organization.

- Step 5: Aggregate the data collected from the experts along with the data collected from the system historical data for each component. System historical data will be assigned weight to reflect the maturity of the data collected within the organization. This weight along with the relative expert's weights will control the portion by which each source will impact the total calculated value for each component.
- Step 6: For each new goal to be implemented, identify those components that will be impacted either through modification or creation.
- Step 7: Determine the eXtent of Modification (XoM) for each impacted component. (XoM) determine to which extent the impacted component will be modified by the proposed goal. For new component (XoM) will equal 1 but for existing components it will be value less than or equal to 1.
- Step 8: Calculate the new component Difficulty of Creation (DoC) value as will be indicated in the next section in details.
- Step 9: Calculate the existing component Difficulty of Modification (DoM) value as will be indicated in the next section in details.
- Step 10: Aggregate the calculated values of (DoM) and (XoM) for exiting components along with (DoC) and XoM for new components to calculate the total impact value of each goal.

4-2 HBMIA details

In order to show the HBMIA process details, we follow a set of steps to calculate both (DoM) and (DoC) for each existing and new components respectively along with the total impact associated with each goal. The following set of parameters will be used in the following formulas.

1- β: Prioritization vector for DoM/DoC factors of system components with dimension (N x 1) for DoM and (M x 1) for DoC.
2- N: Number of DoM factors.
3- M: Number of DoC factors.
4- Φ: Prioritization matrix for components with respect to each DoM/DoC factor. The dimension of this matrix will be (Z x N) for DoM and (Z x M) for DoC.
5- Z: Number of system impacted components.
6- α: Prioritization vector of the components from the perspective of each expert.
7- K: Number of experts.
8- δ: Weighting matrix for the components with respect to experts.
9- X: Number of experts plus 1.
10- DoM(c): Difficulty of Modification value for component (c).
11- DoC(c): Difficulty of Creation value for component (c).
12- XoM(c,g): eXtent of Modification for component (c) and goal (g).
13- MSize(c,g): Modified SLOC result from implementing goal (g) into component (c).
14- Size(c) : Total component size.
15- T: Number of components impacted when implementing goal (g).
16- L: Number of new components impacted when implementing goal (g).

17- R: Number of existing components impacted when implementing goal (g).
18- Impact(g): Total impact of implementing goal (g) in the existing system components.

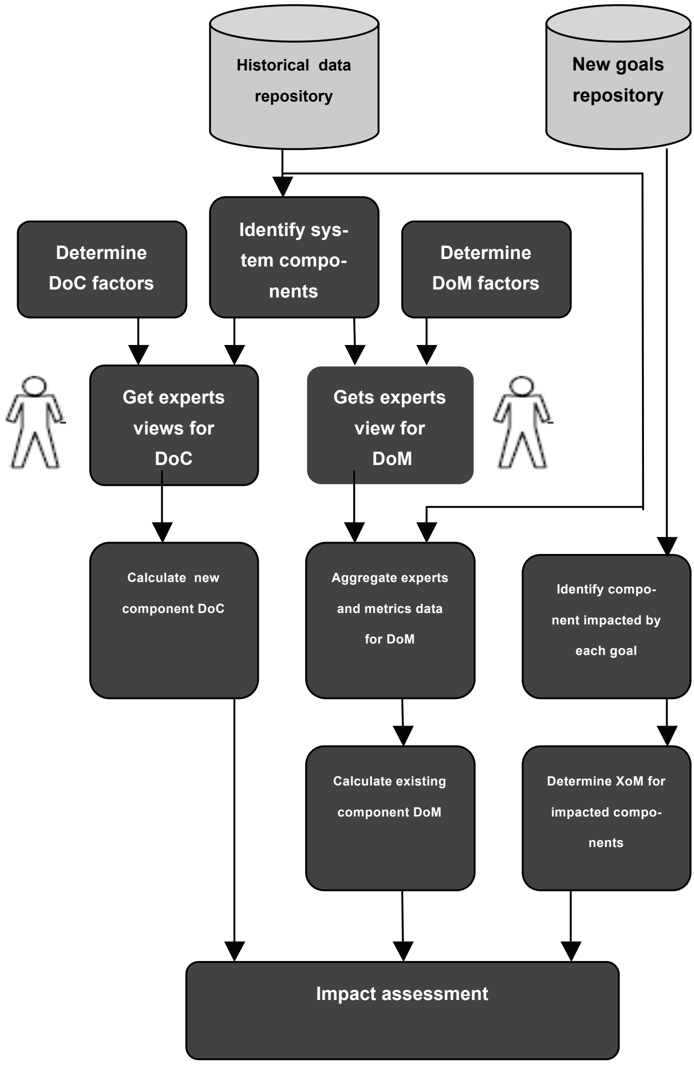

Figure 3 HBMIA architecture

- Step 1: Prioritize DoM/DoC factors. This prioritization will show to which extent each factor will impact the DoM/DoC relative to the other factors. This relative importance of the factors will identify the effect of each factor on DoM/DoC from the point of view of each expert and also based on the maturity of these factors data within the system historical metrics. This will result in a column vector of β(N)/β(M) with (N) refers to the number of factors affecting (DoM) and (M) refers to number of factors affecting (DoC). The values entered by experts or taken from the historical metrics will take values 1 to 9 such that 1 refers to equal importance, 5 strong importance and 9 extreme importance. The values in between refers to intermediate importance.

- Step 2: Normalize the values of the prioritization vector (β) such that :

$$(0 \le \beta\ (n) \le 1) \text{ and } (0 \le \beta\ (m) \le 1) \tag{1}$$

This will guarantee that:

$$\sum_{N} \beta\ (n) = 1 \quad \text{and} \quad \sum_{M} \beta\ (m) = 1 \tag{2}$$

- Step 3: Prioritize components with respect to each DoM/DoC factor. This prioritization will determine how the DoM/DoC for each component will be affected by each DoM/DoC factor. The result from this step is matrix Φ(Z x N) with (Z) refers to the number of impacted system components. Each entry from this matrix (z,n) will show relative manner in which each factor in column (n) affects the DoM/DoC of the component in row (z). The values entered by experts or taken from the historical metrics will take values 1 to 9 such that 1 refers to equal importance, 5 strong importance and 9 extreme importance. The values in between refers to intermediate importance.
- Step 4: Normalize the values of the prioritization vector (Φ) such that :

$$(0 \le \Phi\ (z,n) \le 1) \text{ and } (0 \le \Phi\ (z,m) \le 1) \tag{3}$$

- Step 5: Calculate the relative priorities of components from the perspective of each experts and with respect to the historical metrics data for each component. This will be calculated using the below formula :

$$\alpha(e,z) = \sum_{M} \Phi\ (z,m) \times \beta\ (m) \tag{4}$$

The result from step will be (K+1) vector with (K) refers to the number of experts who contribute in this assessment plus metric vector result from the historical data for each component.

- Step 6: Prioritize the components with respect to each expert based on his familiarity and experience with each component. Since values for all components from the historical metrics with same importance, then it can be assigned the same weights for the metric vector of the components. The result from this step will be a matrix δ(z,x) with (X) refers to the number of experts plus 1 for the metric vector. This will be used only for (DoM) calculation, but for (DoC) calculation will be using δ(z,k) because there is no historical metric contribution. The dimension of the resultant matrix will be δ(Z x X) for (DoM) and δ(Z x K) for (DoC). The values entered by experts or taken from the historical metrics will take values 1 to 9 such that 1 refers to equal importance, 5 strong importance and 9 extreme importance.

- Step 7: Normalize the values of the prioritization vector (δ) such that :

$$(0 \leq \delta\ (z,x) \leq 1) \text{ and } (0 \leq \delta\ (z,x) \leq 1) \quad (5)$$

This will guarantee that:

$$\sum_{X} \delta\ (z,x) = 1 \quad \text{and} \quad \sum_{K} \delta\ (z,k) = 1 \quad (6)$$

- Step 8: Prioritize the experts with respect to each other. Each expert along with the historical metric data will be assigned a relative weight to reflect the importance and maturity of each with respect to the organization. The result from this step is a column vector λ (X) for (DoM) and λ (K) for (DoC). The values entered by organization will take values 1 to 9 such that 1 refers to equal importance, 5 strong importance and 9 extreme importance.
- Step 9: Normalize the values of the prioritization vector (δ) such that :

$$(0 \leq \lambda\ (x) \leq 1) \text{ and } (0 \leq \lambda\ (k) \leq 1) \quad (7)$$

This will guarantee that:

$$\sum_{K} \lambda\ (k) = 1 \quad \text{and} \quad \sum_{X} \lambda\ (x) = 1 \quad (8)$$

- Step 10: Calculate the DoM/DoC values for each component. This step will aggregate the different contributions from the experts and historical data based on the assigned weights described in the previous steps. This will be calculated using the below formulas:

$$DoM(c) = \sum_{X} \delta\ (z,x)\ X\ \alpha(e,z)\ X\ \lambda\ (x) \quad (9)$$

$$DoC(c) = \sum_{K} \delta\ (z,k)\ X\ \alpha(e,z)\ X\ \lambda\ (k) \quad (10)$$

The result from the above formulas will satisfy the inequalities:

$$(0 \leq DoM(C) \leq 1) \text{ and } (0 \leq DoC(C) \leq 1) \quad (11)$$

- Step 11: Calculate the eXtent of Modification (XoM) for each component, goal pair such that XoM(c,g) will reflect to which extent component (c) will be impacted/modified by implementing goal (g). This calculation will be done only for existing components which subject to modification. XoM for new components will be equal to 1 based on the above definition.

$$XoM(c,g) = MSize(c,g) / Size(c) \quad (12)$$

- Step 12: Identify the list of component impacted by implementing each new goal. The result from this step will be a list of component (T) impacted by implementing goal (g). This step called Goals-Driven Impact Analysis (GDIA).
- Step 13: Calculate the total impact of implementing goal (g) in existing system. This impact will be affected by both existing components and new components that impacted by implementing goal (g) in the system.

$$Impact(g) = \sum_{L} (1 + XoM(c,g)) * DoM(c) + \sum_{R} 2 * DoC(c) \quad (13)$$

The value of 2 in the above equation results from substituting of 1 for XoM(c,g) for the new components. Taken into consideration that the total number of existing components (L) plus the total number of new components (R) impacted while implementing goal (g) will equal to the total number of components (T) which impacted by implementing goal (g).

5- CASE STUDY

The (HBMIA) will be evaluated for the components of the (KC1) case study from the Metric Data Program [22] The Metrics Data Program is a database that contains data about problems, products and metrics of a number of software projects. The main objective of the program is to gather, validate, arrange, save and provide software metrics data for the software engineering community. The case study (KC1) is a software component of a data processing unit within a large ground system. The system is made up of 43 (KSLOC) of C++ code. The error data for this code has been collected since the beginning of the project over five years of development and maintenance. The data from (KC1) is analyzed to map it to the class level, so each component refers to a class within KC1.

In order to show the practical benefit from the (HBMIA) we will consider 10 new goals G1 to G10 scheduled for implementation as part of the future product maintenance. In this case study, there are two experts (expert_1, expert_2) along with the system historical data metrics data. The main objective of these hypothetical goals and expert's data is to be able to evaluate our proposed (HBMIA) methodology against existing ones taking into consideration the real data provided from the existing system of (KC1). Our scope through this case study using (KC1) component is to show the impact of the system history metrics on evaluating the new goals impact on the existing system. Thus we proposed 10 new goals along with expert's data to be used through the evaluation. Given the components affected by each new goal from table 1 to table 13 taken into consideration that CM1, CM2, CM3 are existing system components while CM4, CM5 are new components, we will be able to calculate the goals impacts by following the (HBMIA) algorithm steps mentioned in the previous section. These impacts values will identify which goals to be selected for implementation based on the available product resources. The data given in table 1 indicate both G1, G3 will result in creation of two new components CM4, CM5 respectively along with modification of other existing components.

Table 1 Components impacted by new goals

Components/ Goals	CM1	CM2	CM3	CM4	CM5
G1	X				X
G2			X		
G3	X	X		X	
G4	X				
G5	X	X	X		
G6	X		X		
G7		X			
G8			X		
G9	X		X		
G10		X	X		

Table 2 DoM factor historical metric values

Factors Compo-nents	Size	Compl.	Health	Un-derstd.	Function.	Weight
CM1	0.13	1	0.06	1	0.07	1
CM2	0.02	0.13	0.07	0.41	0.05	1
CM3	0.06	0.35	0.09	0.25	0.08	1

Table 3 DoM factor values for expert_1

Factors Components	Size	Compl.	Health	Understd	Function	Weight
CM1	0.3	0.6	0.44	0.27	0.23	3
CM2	0.5	0.3	0.25	0.54	0.3	4
CM3	0.2	0.1	0.31	0.18	0.46	6

Table 4 DoM factor values for expert_2

Factors Components	Size	Compl.	Health	Understd	Function	Weight
CM1	0.2	0.43	0.26	0.44	0.27	5
CM2	0.7	0.21	0.6	0.37	0.37	7
CM3	0.1	0.36	0.14	0.12	0.12	3

Table 5 DoC factor values for expert_1

Factors Components	Size	Compl.	Critical	Understd	Depend.	Weight
CM4	0.71	0.7	0.28	0.6	0.6	1
CM5	0.29	0.3	0.72	0.4	0.4	2

Table 6 DoC factor values for expert_2

Factors Components	Size	Compl.	Critical	Understd	Depend.	Weight
CM4	0.75	0.44	0.53	0.75	0.6	3
CM5	0.25	0.56	0.47	0.25	0.4	2

Table 7 DoM factor historical metric weights

Factor	Size	Complexity	Health	Understandability	Functionality
Weight	1	1	1	1	1

Table 8 DoM factor weights for expert_1

Factor	Size	Complexity	Health	Understandability	Functionality
Weight	3	5	2	7	1

Table 9 DoM factor weights for expert_2

Factor	Size	Complexity	Health	Understandability	Functionality
Weight	5	2	6	3	4

Table 10 DoC factor weights for expert_1

Factor	Size	Complexity	Critical	Understandability	Dependability
Weight	2	6	4	9	3

Table 11 DoC factor weights for expert_2

Factor	Size	Complexity	Critical	Understandability	Dependability
Weight	7	1	3	7	4

Table 12 Relative expert's weights

Expert	Relative Weight
Historical Metric	9
Expert_1	3
Expert_2	5

Table 13 XoM data for modified components

Component	Volume(SLOC)	Modified size (SLOC)
CM1	2789	2430
CM2	401	332
CM3	1621	1123

Following the steps detailed in the previous section, we will be able to calculate the total goal impact on the system components as follows in the below steps:

- Step 1: Calculate the DoM/DoC for all the system components using formulas 1 to 11. This will yield to the following results shown in table 14.

Table 14 DoM/DoC component values

Component	DoM/DoC
CM1	0.09
CM2	0.11
CM3	0.05
CM4	0.11
CM5	0.08

- Step 2: Calculate the total goal impact based on the goal assignment shown in table 1, (XoM) data in table 13 and DoM/DoC calculated values shown in table 14 using formulas 12, 13. This yields to the following results shown in table 15.

Table 15 Total goals impact values

Goal	Total impact
G1	0.33
G2	0.08
G3	0.58
G4	0.17
G5	0.45
G6	0.25
G7	0.20
G8	0.08
G9	0.25
G10	0.28

As indicated from table 15, G3 is the goal with the greatest impact on the system components and both G2, G8 are the goals with the lowest impact on the system components. The goals proposed for implementation should be selected based on their impacts on the system components such that goals with lower impacts will be good candidates for implementation because it will consume lower resources and costs [40]. Thus calculating the goal impact on system components can be used for better planning activities such as resource allocation and producing better estimates [35]. This will enable project manger of the software product to get better control over the product resource and achieve product success [25].

6- EVALUATION OF HBMIA VERSUS SIMILAR METHODOLOGY

In order to show the strength of the new proposed technique (HBMIA) compared to other existing and similar approach proposed by Saliu [34], an empirical evaluation should be conducted. The main drawback of methodology proposed by Saliu is that it depends mainly on experts experience with the system components in deriving the information about these components even when there is no evidence about their values. It also neglects the components history of the system which contains valuable indicators about future trends of the system components [21]. Saliu's proposed methodology also takes the effect of existing components only while calculating the total goal impact and neglecting

the effect of new components result from implementing the new goals.
On the other hand, we tried to cover these drawbacks in our proposed (HBMIA) methodology. The new (HBMIA) combines the strengths of both metric-based and expert-based approaches while collecting DoM/DoC factor values. It balances between historical and experts contributions through weights assigned within the organization based on the importance and maturity of the experts and historical data. (HBMIA) also gets the effect of the new components while calculating the total goal impacts. This will provide a better quality for the calculated goal impact especially when the new goals result in creation of new components.
In order to be able to compare the two methodologies, we will calculate the (DoM) values for the existing components (CM1, CM2, and CM3) without taking the effect of the historical metric data into consideration. This will result in the data shown in table 16.

Table 16 DoM/DoC component values (Saliu's methodology)

Component	DoM
CM1	0.15
CM2	0.22
CM3	0.08

Calculate the total goal impact based on the goal assignment shown in table 1, (XoM) data in table 13 and (DoM) calculated values shown in table 14 using formulas 12, 13. This will yield to the following results shown in table 17.

Table 17 Total goals impact values (Saliu's methodology)

Goal	Total impact
G1	0.28
G2	0.13
G3	0.68
G4	0.28
G5	0.80
G6	0.41
G7	0.40
G8	0.13
G9	0.41
G10	0.53

To evaluate the (HBMIA) against Salui's methodology, we will compare the results from the two methodologies as follows:

1. Comparing the data shown in tables 14, 16 for both methodologies, we conclude that (DoM) values calculated by (HBMIA) is around 55% of the corresponding values calculated by Saliu's methodology. This indicates that taking historical data into account while calculating the (DoM) for the existing components will result in higher quality values which will affect the goals impact calculation. The higher quality gained from (HBMIA) results because of high maturity historical data used from (KC1) test case reflected also from the relative weight 9 given to historical data in table 12.

The metrics data is justifiable from system history over the past system releases which differs from expert's data that has no justification and it depends only on the experts experience without any evidence on its quality. The importance of using the hybrid-based approach through (HBMIA) methodology is the ability to balance between the expert-based and metric-based approaches based on the organization maturity. This is not the only advantage for using the hybrid-based approach but it also overcomes the bias effect of expert and provides a justifiable data based on the system metrics as indicated in table 2. The data shown in tables 2, 3 and 4 indicates that DoM values for components CM1, CM2 and CM3 calculated from (HBMIA) reflect accurate results based on the relative weights detailed in table 12. The main advantage from our (HBMIA) methodology arises for cases where the existence of experts is hard and too expensive to the organization. Thus (HBMIA) provides higher flexibility to accommodate with the business case to better provide higher accuracy based on the maturity and availability of the experts.

2. Comparing the data shown in tables 15 and 17 we conclude that G3 has the greatest impact from HBMIA while G5 has the greatest impact from Saliu's methodology. To evaluate the correctness behind these results, we need to identify effect of each goal on the system components as indicated in table 18.

Table 18 Component impact from HBMIA and Saliu's methodologies

Component name	HBMIA impact	Saliu's impact
CM1	0.17	0.28
CM2	0.2	0.4
CM3	0.08	0.13
CM4	0.22	Not supported
CM5	0.16	Not supported

G5 results in modification of 3 existing system components (CM1, CM2, CM3) while G3 results in creation of (CM4) and modification of (CM1, CM2). Since creation of new components has a bigger impact on the system than the modification of existing components [8], the results from (HBMIA) tends to be more accurate than Saliu's methodology results. This can be also shown from G1 which results in creation of (CM5) and modification of (CM1). The impact value calculated by (HBMIA) is much higher than that calculated by Saliu's which is reasonable because Saliu's impact result from the modification of (CM1) only and neglects the impact of new component creation.

7. CONCLUSION

In this paper, we have presented a new methodology for calculating the impact of implementing new goals on existing system called (HBMIA). Our contribution in this paper is two fold; first, the new (HBMIA) combines the strengths of both metric-based and expert-based approaches while collecting DoM/DoC factor values. It balances between historical and experts contributions. Second, (HBMIA) gets the effect of the new components while calculating the total goal impacts. This will provide a better quality for the calculated goal impact especially when the new goals result in creation of new components. Thus (HBMIA) will overcome the drawbacks of existing similar approaches which depend only on the expert's data which has no evidence on its correctness and neglect the historical data which has good indicators of the future trends of the system components.

Since only a few studies have been performed to evaluate the efficiency and suitability of (HBMIA), there is a need to do further studies for some issues that can affect the algorithm efficiency. Further evaluation for the different factors that influence both (DoM) and (DoC) is required. Further analysis for the type of modification on the (DoM) assessment is required. Analyze how the software developer productivity will impact the assessment of both (DoM) and (DoC).

REFERENCES

[1] K.K. Aggarwal, Y. Singh, and J.K. Chhabra, "An Integrated Measure of Software Maintainability", Proc. Annual Reliability and Maintainability Symposium, IEEE, 2002.

[2] A.J. Albrecht, "Measuring application development productivity". Proc. IBM Applications Development Sysprosium (Monterey,Calif.), pp. 83- 92, 1979.

[3] R.S. Arnold, S.A. Bohner, "Impact Analysis – Toward a Framework for Comparison", Proc. Conference on Software Maintenance, Montreal, Canada, IEEE Computer Society Press, Los Alamitos, CA, pp. 292-301, 1993.

[4] L.J. Artur, Software Evolution: The Software Maintenance Challenge, John Wiley & Sons, New York, NY, 1988.

[5] D. Ash, J. Alderete, L. Yao, P.W. Oman, and B. Lowther, "Using software maintainability models to track code health", Proc. International Conference on Software Maintenance, IEEE, 1994.

[6] P. Bengtsson, "Towards Maintainability Metrics on Software Architecture: An Adaptation of Object-Oriented Metrics". Proc. First Nordic Workshop on Software Architecture. Research Report ISSN: 1103-1581, Blekinge Institute of Technology, Ronneby, Sweden, pp. 87-91, 1998.

[7] P. Bengtsson, "Architecture-Level Modifiability Analysis", Ph.D. disseration, Blekinge Institute of Technology, Sweden, 2002.

[8] B. Kitchenham and E. Mendes, “Software productivity measurement using multiple size measures,” IEEE Trans. On Software Engineering, vol. 30, pp. 1023-1035, 2004.

[9] S.A. Bohner, R.S. Arnold, Software Change Impact Analysis, IEEE Computer Soc. Press, Los Alamitos, CA, 1996.

[10] S.D. Conte, H.E. Dunsmore and V.Y. Shen, Software Engineering Metrics and Models, Benjamin/Cummings Publishing Company, 1986.

[11] N.E. Fenton and S.L. Pfleeger, Software Metrics -A Rigorous & Practical Approach, PWS Publishing Company, 1997.

[12] R.B. Grady, “Successfully Applying Software Metrics”, IEEE Computer, vol. 27, issue 9, 1994.

[13] IEEE Standard for Software Maintenance, Std 1219-1998, Software Engineering Standard, vol. 2, 1999.

[14] ISO/IEC-14764 Information technology – Software Maintenance, Standard, First edition, Nov 11, 1999.

[15] M.M. Lehman, “On understanding Laws, evolution and conversation in the large program lifecycle”. Journal of Software & Systems, vol. 1, pp. 213 – 221, 1980.

[16] M. Lindvall and K. Sandahl, “How Well do Experienced Software Developers Predict Software Change?”, Journal of Systems and Software, vol. 1, pp. 19-27,1998.

[17] B. Lientz, E. Swanson, Software Maintenance Management, Reading, Mass., Addison-Westley, pp. 214, 1980.

[18] L.C. Briand, V.R. Basili, Y.M. Kim, and D.R. Squier, “A Change Analysis Process to Characterize Software Maintenance Projects”, Proc. International Conference on Software Maintenance, Victoria, Canada, 1994.

[19] J. Martin and G. McClure, Software Maintenance: The Problem and its Solutions, Prentice Hall, 1983.

[20] K.D. Maxwell, L. Van Wassenhove and S. Dutta, “Software Development Productivity of European Space, Military, and Industrial Applications”, IEEE Transactions on Software Engineering, vol. 22, pp. 706-718, 1996.

[21] Metrics That Matter, Proc. 27th Annual NASA Goddard/IEEE Software Engineering Workshop, 2002.

[22] Metrics Data Program, NASA IV&V Facility http://mdp.ivv.nasa.gov/. 2008.

[23] S. Muthanna, K. Ponnambalam, K. Kontogiannis and B. Stacey, “A Maintainability Model for Industrial Software Systems Using Design Level Metrics”, Proc. seventh Working Conference on Reverse Engineering (WCRE’00), Brisbane, Australia, 2000.

[24] NASA Technical Std. NASA-STD-8719.13A, Software Safety, http://satc.gsfc.nasa.gov/assure/nss8719_13.html. 1997.

[25] J. M. Nicholas, Project Management for Business and Technology - Principles and Practice, 2nd Edition, Prentice Hall, Upper Saddle River, NJ, 2001.

[26] P. Oman and J. Hagemeister, “Metrics for Assessing a Software System’s Maintainability”, Proc. Conference on Software Maintenance, IEEE, 1992.

[27] P. Oman, , J. Hagemeister, , and D. Ash, , A Definition and Taxonomy for Software Maintainability, report SETL Report 91-08-TR, University of Idaho, 1991.

[28] J.P. Queille, , J.F. Voidrot, , N. Wilde, , M. Munro, “The Impact Analysis Task in Software Maintenance: A Model and a Case Study”, Proc. International Conference on Software Maintenance, Victoria, Canada, IEEE Computer Society Press, Los Alamitos, CA, pp. 234-242, 1994.

[29] T.M Pigoski, Practical Software Maintenance: Best Practices for Managing your Software Investment. Wiley, 1997

[30] S. Robertson, and J. Robertson, Mastering the Requirements Process, Addison-Wesley, Harlow, England, 1999

[31] J.F. Ramil and M.M. Lehman, “Metrics of Software Evolution as Effort Predictors - A Case Study”, Proc. International Conference on Software Maintenance, IEEE, 2000

[32] X. Ren, , F. Shah, , F. Tip, B. Ryder, and O. Chesley, Chianti, “A Tool for Change Impact Analysis of Java Programs”, Proc. Conference on Object-Oriented Programming Systems, Languages, and Applications, October 26-28, Vancouver, Canada, pp. 432-448, 2001.

[33] O. Saliu, G. Ruhe, “Software release planning for evolving systems”. Innovations in Systems and Software Engineering – A NASA Journal vol 1, 189–204, 2005.

[34] T. Scott, D. Farley, Slashing Software Maintenance Costs, Business SoftwareReview, Indianapolis, 1988.

[35] SEI Software Technology Review, Cyclomatic Complexity, http://www.sei.cmu.edu/, 2002.

[36] C. Szyperski, Component Software - Beyond ObjectOriented Programming, Addison-Wesley, 1998.

[37] S.R. Chidamber, C.F. Kemerer, “Towards a metrics suite for object oriented design”, Proc. International workshop on Principles of software evolution, 1991

[38] R.J. Turver, and M. Munro, “An Early Impact Analysis Technique for Software Maintenance”. Journal of Software Maintenance Research and Practice, vol 6, pp. 35-52, 1994.

[39] K. Wiegers, Software Requirements Second Edition, Microsoft Press, 2003.

[40] C. Wohlin, and A. Aurum, “Criteria for Selecting Software Requirements to Create Product Value: An Industrial Empirical Study’, in Value-based Software Engineering”, Springer Verlag, Germany, pp. 179-200, 2006.