Scientific Journal of Riga Technical University
Computer Science. Applied Computer Systems
2011
Volume 46

A Method for Documenting Modifications in ERP Systems

Martins Ebersteins, Janis Grabis, *Institute of Information Technology, Riga Technical University*

***Abstract* - Modern Enterprise Resource Planning (ERP) systems are highly complex. This complexity is further increased by introducing modifications into the standard functionality. Maintenance and evolution of ERP systems becomes unmanageable if modifications are not properly documented. This paper elaborates a method for documenting and analyzing modifications made in ERP systems. These modifications are classified according to their impact on the standard functionality and a set of parameters for defining each modification is established. A template for describing modifications is also developed. The modifications are stored in the register of modifications and spreadsheet tools are used for analyzing these modifications. The analysis example is provided. It demonstrates identification of relationships between modifications and impact of modifications on elements of the standard system.**

***Keywords* - documentation, ERP, modification.**

I. INTRODUCTION

An Enterprise Resource Planning (ERP) system is a standard software system, which provides functionality to integrate and to automate the business practices associated with operations or production aspects of a company. The integration is based on a common data model for all system components and extents to more than one enterprise sectors [2].

Because ERP systems in the last decades have enormously progressed in their functionality, they have become so complex that if in the past one consultant could fully know almost all parts of the standard ERP system, then at present one consultant can barely manage functionality of few standard ERP modules and in the case of the largest ERP systems it could be just one module or part of it. If the standard ERP additionally has been modified by introducing enterprise specific functionality, it is almost impossible for consultants, project managers, programmers and other employees to precisely know the functionality of the implemented ERP without proper documentation [3,12].

The modification increases complexity [6,8] and causes dependence upon knowledge of particular employees and partner companies [3,11], complex migration to newer ERP releases [11,12], painful further modifications [11], hard problem tracking (errors, performance and stability issues etc.), high time usage to comprehend modifications and relations between them and other problems. That all usually leads to increased costs and lost time [11]. The additional future costs and reaction time could be decreased by proper and effective modification documentation method [3].

A universal method for modification documentation and analysis is highly required as at present there are no defined standards for doing it. It is possible to find some general suggestions for documenting changes [7], standards for documenting particular ERP systems modifications [1,10] or difficult to use maintenance methods borrowed from development of custom software [4], but there are no common standards or even research cases, providing information on how modifications should be documented and how to operate and use this documentation.

The objective of this paper is to elaborate a formal method for identification and documentation of changes made in the ERP system and for analyzing relationships among these changes. The formal definition of modifications would provide all necessary information about the implemented ERP modifications in such a manner that a 3rd party ERP partner would be able to understand the exact modifications that have been made in the standard ERP; how they affect the standard ERP functionality; how to manage those modifications; how to make those modifications work with newer ERP releases etc., without reading the source code of the modifications.

The method includes classification of modifications, calculation of modification value, definition of modification parameters and analysis of modifications. Modifications are identified using their value, which allows defining modifications, using a template-based model. This model represents only the modified elements of the ERP system. A modification description template is created for each modification type. Combining the heading of modifications description with one or several modification specific type descriptions creates a full description of modifications. For analysis purposes a method for registering, sorting, analyzing and identifying modifications is developed.

Using the proposed methods for describing modifications should decrease effort needed for further evolution and maintenance of ERP systems as well as should improve reaction time of the ERP partner. The method is geared for middle-tier ERP systems and medium size consulting companies, which do not extensively rely on sophisticated software engineering methods.

This paper is divided into 5 sections. Section 2 identifies types of modifications. Definition of modifications is elaborated in Section 3. A sample of analysis of modifications is provided in Section 4. Section 5 concludes the study.

II. TYPES OF MODIFICATIONS

ERP systems are software packages composed of several modules, such as human resources, sales, finance and production, providing cross-organization integration of data through imbedded business processes. These software

packages can be customized to meet the specific needs of each organization. [5]

A typical ERP system consists of three layers – client layer, business logic (application) layer and database layer. The application layer consists of three domains – report domain, journal domain and function domain.

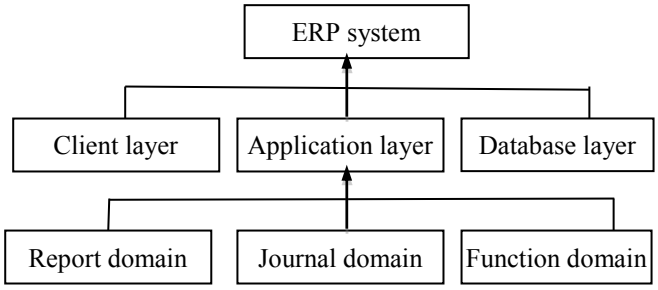

Fig. 1. Main layers in ERP systems

The term "Client layer" in this case represents the GUI, report domain represents the standard ERP system reports, journal domain represents the standard ERP system journals used for creating and editing data and function domain represents the standard ERP system functions, that automatically updates data in journals. In this paper only those modifications affecting the client layer, report domain, journal domain and function domain will be described.

Besides the client and application layers, modifications also can affect other resources:

- Standard ERP system;
- WEB services;
- Locally installed software;
- Remotely accessed software;
- Hardware devices.

Modifications can also affect other modifications, that are already part of application or (and) client layers.

In this paper we refer to the term "modification" as an implementation of a single change request – it could be, for example, one project in Microsoft Dynamics AX, one package in SAP, one *.hal file in HansaWorld. The modification stands for a single implementation, made for some exact purpose. Several implementations realized in one project (for example, new report for sales analysis and a connection with POS hardware devices) should be understood as two separate modifications and should be documented in that way. The term "modification" will not be applied to any changes made by configuring or by using high level modification tools available in ERP systems, for example, a document form creation tool, a report generation tool etc. In other words, the term applies only to changes made to the standard ERP by using some programming language and by changing the standard ERP's functionality in a way what was not directly provided by an ERP developer.

Each single modification *m* affecting only one resource is defined by a resource *R*, type of modification *T*, a set of modification parameters *P* and modification value *v (1)*:

$$m = \{R, T, P, v\} \quad (1)$$

The resource *R* affected by the modification, type *T* of modification and modification value *v* is determined, according to Table I.

TABLE I

MODIFICATION CLASSIFICATION VALUES AND LETTERS

Resource	Use	Modify	Read data / receive data	Write data / send data	Create new
Standard ERP reports (including exports and document forms)	1 a	2 b			*
Standard ERP journals (including configurations)		4 c	8 d	16 e	*
Standard ERP functions (including imports)	32 f	64 g			*
Standard ERP GUI and accessibility		128 h			
ERP elements					256 i
Other modifications	512 j	1024 k			
WEB services			2048 l	4096 m	
Local software			8192 n	16384 o	
Remote software			32768 p	65536 q	
Hardware			131072 r	262144 s	

"*" mark in the table means that it is possible to create new reports, journals and functions, but all those elements are described in letter "i" or value 256 description, because the same template is used for describing all new elements. The set of modification parameters defines the modification and is described in Section 3.

An actual modification *M* usually consists of multiple single modifications and it is defined as *(2)*:

$$M = \{m, v_M\} \tag{2}$$

where *m* is a vector of single modifications comprising the modification *M* and v_M is calculated using *(3)*:

$$v_M = \sum_{i=1}^{19} a_i v_i = \sum_{i=1}^{19} a_i 2^{i-1} \tag{3}$$

where $a_i \in (0,1)$ indicates whether resources of type *i* have been modified.

This is why 2^n values are used, to determine the type of a modification – as there are 19 used cells and in every cell it is possible to use two values – "yes" or "no", total count of all different possible cases is 524287 ($2^{19} - 1$). The exact value should be added to the modifications total value, if the modification uses the exact resource in the exact type. By using the 2^n values it is possible to precisely determine everything in resource and modification types comprising the modification. Knowing, for example, that a modification value is 85, the only way, how this sum can be obtained, is by summing 64, 16, 4 and 1. That means that this modification uses standard ERP reports, writes information to some standard ERP journals and modifies some standard ERP journals and also modifies some standard ERP functions. The numeric value allows storing information about total 524287 possible modification types in a single integer field, instead of using string type values or even tables.

To determine resources and modification types comprising the modification, the (4) algorithm should be used.

if vM -218>0 then ms=1 and vM = vM -218, else ms=0;

if vM -217>0 then mr=1 and vM = vM -217, else mr=0;

if vM -216>0 then mq=1 and vM = vM -216, else mq=0;

...

if vM -20>0 then ma=1 else ma=0, (4)

where "v_M" is total calculated value of the modification, $m_a \ldots m_s$ identifies the used elements, described in table 1.1, if $m_x=1$ then modification uses this exact element.

The letters are used to simplify in some cases the recognition of exact modification type and they are fully inter-exchangeable with numbers.

III. Definition of Modifications

A. Single Modification Definition Template

Each single modification definition consists of a header and details. As usually modifications consist of a group of single modifications, the header will allow associating all single modifications, but the details part will describe each single modification.

The header is defined as it is shown in the Table II.

TABLE II
Data Fields Used in the Header

#	Fields name	Description
1	Modification type	The modifications letter
2	Modification value	The modifications value

The modification type letter allows easy locating of necessary single modification in a combined modification description. The single modification value allows calculating the total modification value (M). Additionally to the header, each single modification description contains a set of parameters used for the description.

B. Modification Parameters

The details section of the modification description is defined using a set of parameters. Each modification type has its own set of parameters. Parameters used to define modifications are introduced in Table III.

Table IV defines the set of parameters used to define each type of modifications. For example, the modification of type "a" is defined using N, P, X, DR, F and D parameters.

TABLE III

MODIFICATION PARAMETERS

Variable	Name	Description
A	Functionality after changes	Describes how the affected element works after changes
AC	Accessibility permissions	Describes accessibility permissions for different user groups, if differs
AP	Appearance	Describes newly created elements GUI or reports results sample
B	Functionality before changes	Describes how the affected element worked before changes
C	Connection	Describes how software connects to some 3rd party elements
CR	Records chosen	Description of rules used for journal record choosing
D	Description	Short description of modification purposes
DC	Data converting	Description how data is converted to required format
DEV	Developer	Indicates hardware or software developer
DF	Data fields	Describes used or created data fields
DR	Data received	Describes data that is received from 3rd parties
DS	Data sent	Described data that is sent to 3rd parties
DST	Data destination	Indicates journal or function to which data is delivered
F	Internal functions	Indicates internal functions contained in different elements and that are used or modified by modifications
G	Globalization	Indicates if changes to data definition are global for the field type in the whole ERP or just in a single journal
N	Name	Indicates internal or external name of the affected element
P	Path	Describes, how element can be accessed in the ERP
REL	Release	Release or model of 3rd party elements
S	Sample	GUI or report samples
SR	Special pickup rules	Special data pickup rules used for data gathering, validating or choosing
SRC	Data source	Functions or journal internal name
ST	Standard	Standards used for data exchange between ERP and 3rd party elements
T	Type	Indicates element type – report, journal or function
X	Parameters passed	Describes parameters passed to elements
Y	Parameters received	Describes parameters received from elements

TABLE IV

SET OF PARAMETERS USED ACCORDING TO THE TYPE OF MODIFICATION

Modification type	Set
a	M=(N, P, X, DR, F, D)
b	M=(N, P, B, A, F, D, S)
c	M=(N, P, B, A, G, D, S)
d	M=(N, P, CR, DF_1, DF_2, SR)
e	M=(N, P, CR, DF, SRC, D, DC)
f	M=(N, P, D, F, X, Y)
g	M=(N, P, F, B, A, D)
h	M=(N, P, B, A, D, S)
i	M=(T, N, P, AP, DF, AC, D, S)
j	M=(N, T, P, F, D, X, Y)
k	M=(N, T, P, D, B, A, S)
l	M=(N, D_1, C, ST, X, DR, D_2)
m	M=(N, D_1, C, ST, X, Y, DS, D_2)
n	M=(N_1, DEV_1, REL_1, P_1, C_1, ST, N_2, DEV_2, REL_2, P_2, D_1, C_2, DR, D_2, X, DST)
o	M=(N_1, DEV_1, REL_1, P_1, C_1, ST, N_2, DEV_2, REL_2, P_2, D_1, C_2, DS, D_2, X, Y, SRC)
p	M=(N_1, DEV_1, REL_1, C_1, ST, N_2, DEV_2, REL_2, P_1, D_1, C_2, DR, D_2, X, DST)
q	M=(N_1, DEV_1, REL_1, C_1, ST, N_2, DEV_2, REL_2, P_1, D_1, C_2, DS, D_2, X, Y, SRC)
r	M=(N_1, DEV_1, REL_1, C, ST, N_2, DEV_2, REL_2, N_3, DEV_3, REL_3, Y_1, Y_2, X_1, X_2, DR, DST, D)
s	M=(N_1, DEV_1, REL_1, C, ST, N_2, DEV_2, REL_2, N_3, DEV_3, REL_3, Y_1, Y_2, X_1, X_2, DS, SRC, D)

C. Combined Modification Definition Template

The complete modification description consists of the header (Table V) and description of every single modification comprising the complete modification.

TABLE V

DATA FIELDS USED IN THE HEADING

#	Fields name	Description
1	Modification value	The calculated modifications value (M).
2	Modification name	Modifications internal or file name.
3	Description	Short description of modification purpose

Sorting modifications by the calculated modifications value (M) would allow keeping similar modifications together and also quickly finding the necessary modification, if it is known, what elements that modification uses, because in this case it would be possible to calculate the calculated modifications value (M) and to search by it for the necessary modification.

As usually there will be at least some modifications with the same value, it is also necessary to use the modification file name or system internal name as a second identification field. As this name is unique, there will never be two modifications with equal names, what mean that this field can be used as the primary search key.

If it is possible, a short description of modifications should be written, more precisely indicating modifications type and modifications purpose. Information for what purposes the modification is made and how the modification can be accessed or in what cases the modification is executed also should be included.

After the header part, descriptions of every single modification are included.

D. Definition Examples

An example of modification description is shown in Fig. 2. The sample header contains value 10, which is calculated as a sum of included single modifications (2+8), ArtInStockVcREP.mod is the modifications file name, a short description of modification purposes is in the next line followed by descriptions of modification of Type B and a D.

A. Register of Modifications

All modifications are registered in a single register, for example, spreadsheet. An example of spreadsheet is given in Fig. 3. One modification description usually is registered in several lines, displaying in the first column modification total calculated value (M), in the second column – modification internal or file name, in the following two columns single modification values and letters are registered, in the next column internal names or addresses of used elements (such as journals, reports, other modifications, WEB – services etc.) are registered, and in the last column the exact used fields or functions (such as some exact journal data fields, internal functions etc.) are registered.

10
ArtInStockVcREP.mod
Adds to “Items in stock” report column “Item group”.
B Type
2
N: ArtInStockVcREP
P: Stock module, Report, “Items in stock”
B: -
A: Label “Item group”, 25 from up, 200 from left, “Arial”, size 10
B: -
A: Column “ItemGroup” 50 from up, 300 long, 200 from left 50 wide, “Arial”, size 10, centered, wrapping – off, cutting – on
SRC: StockVc
DF:

Name	Data type	Data length	Data description	Separator
ItemGroup	M4Code	10	Code String	<TAB>

D Type
8
N: StockVc
P: Stock module, Stock journal
CR: matches [ItemCode] received
DF_C: Reads [ItemGroup]
DF_R:

Name	Data type	Data length	Data description	Separator
ItemCode	M4Code	20	Code String	<TAB>
ItemGroup	M4Code	10	Code String	<TAB>

Fig. 2. ArtInStockVcREP.mod Description

Mod. value	Mod. name	Part value	Part letter	Part name / address	Fields / functions used
265	DailySales.mod	256	i	MinPerLocRep	CustGroupName
265	DailySales.mod	256	i	MinPerLocRep	ItemInStock
265	DailySales.mod	256	i	MinPerLocRep	ItemMinLevel
265	DailySales.mod	256	i	MinPerLocRep	ArtCode
265	DailySales.mod	8	d	PUVc	RegDate
265	DailySales.mod	8	d	PUVc	Objects
265	DailySales.mod	8	d	PUVc	Location
265	DailySales.mod	8	d	PUVc	ArtCode
265	DailySales.mod	8	d	PUVc	Quant
265	DailySales.mod	8	d	SOVc	RegDate
265	DailySales.mod	8	d	SOVc	ArtCode
265	DailySales.mod	8	d	SOVc	Location
265	DailySales.mod	8	d	SOVc	BasePrice
265	DailySales.mod	1	a	InStockMovmVcREP	-
152088	Supply.mod	8	d	PUVc	RegDate
152088	Supply.mod	8	d	PUVc	Location
152088	Supply.mod	16	e	PUVc	RegDate
152088	Supply.mod	16	e	PUVc	Location
152088	Supply.mod	512	j	CClassVcNewJournal.mod	RecvArtMinBestBefDate
152088	Supply.mod	4096	m	http://www.bank.lv/excel/valkurlv.php	-
152088	Supply.mod	131072	r	USB Desktop Reader UHF	SntItVc
152088	Supply.mod	16384	o	VIDEDSPVN1	ExportVATtoEDSREP

Fig. 3. Spreadsheet Example

IV. Analysis of Modifications

Analysis of modifications can be done in at least 2 different scenarios. The main tool for modification analysis is the modification register, for example, spreadsheet and the required information tracking can be done by using pivot tables.

In the first case, scenario modification analysis is done by locating a required modification and acquiring information about elements affected by this modification. This is useful when the modification is not functioning properly.

The second scenario allows identifying all those modifications that use particular elements of the ERP system. This is useful, for example, knowing that in a new ERP systems release some standard data fields, journals or internal functions are modified. In this case it is possible to locate all those modifications that use the affected elements.

These analysis scenarios are explored using examples from modification of the HansaWorld ERP system, where two modifications, namely, DailySales.mod and Supply.mod have been implemented and documented.

The DailySales.mod modification creates new ERP report with internal name MinPerLocREP. Report reads data from standard ERP journals PUVc and SOVc and also uses a standard ERP report with internal name InStockMovmVcREP.

The Supply.mod modification reads and writes data to the standard ERP journal PUVc, uses DailySales.mod modification, receives currency exchange rates from http://www.bank.lv/excel/valkurlv.php, sends data to VIDEDSPVN1 local software and reads data from USB Desktop Reader UHF hardware device.

The first analysis example (Fig. 4.) allows reviewing all modifications, displaying all implemented modifications and all single modifications included in these modifications as well as elements of the ERP system affected by these modifications. To create a pivot table, column label fields used are "Fields / functions used" field, row labels are "Mod. value", "Mod. name", "Type value", "Type letter" and "Element name" (in this exact order), while "count of "Fields / functions used" is used for values.

It is possible to clearly see in the pivot table that there are in total 2 modifications – DailySales.mod and Supply.mod, with their total calculated values 265 and 150040.

The DailySales.mod modification uses standard ERP's report InStockMovmVcREP report, reads data from two journals – PUVc and SOVc, and also creates new report MinPerLocREP. From the PUVc journal the modification reads data from fields ArtCode, Lucation, Objects, Quant and RegDate, while from the SOVc journal the modification reads data from fields ArtCode, BasePrice, Location and RegDate. The newly created report MinPerLocREP includes data from standard ERP's fields ArtCode. CustGroupName, ItemInStock and ItemMinLevel. The column GrandTotal displays that 5 and 4 fields from journals PUVc and SOVc, respectively, are used, while the MinPerLocREP report uses 4 standard ERPs data fields in total.

The Supply.mod modification reads and writes data to the PUVc journal, uses DailySales.mod modification, receives data from the http://www.bank.lv/excel/valkurlv.php WEB service, sends data to the VIDEDSPVN1 local software and receives data from the USB Desktop Reader UHF hardware device. The pivot table also provides data that the modification reads and writes data to the PUVc journal Location and RegDate fields, that from the modification DailySales.mod report MinPerLocREP is being used, that data to the VIDEDSPVN1 local software is being sent, using the ExportVATToEDSREP report and that the data from the USB Desktop Reader UHF hardware device is received, using the SntItVc function. The Grand Total column provides data that

two fields from the journal PUVc are being used, while the rest Supply.mod elements use only single elements.

The pivot table also provides information that modifications use the ArtCode field three times, BasePrice, CustGroupName, ExportVATtoEDSREP, ItemsInStock, ItemMinLevel, MinPerLocREP, Objects, Quant and SntItVc fields or functions are used once and Location and RegDate are used four times.

The second example, provided in Fig. 5., allows tracking required modifications by element name, and displaying elements affected by single modifications. This example uses the same modifications as in the first example, displaying the data in a differently formatted pivot table - the column label still remains "Fields / functions used" field, while row labels are sorted in different order - "Element name", "Mod. value", "Mod. name", "Type value" and "Type letter", and values are determined using - "count of "Fields / functions used".

The example shows that the DailySales.mod modification is used by the Supply.mod modification and that exactly the MinPerLocREP report, created by the Supply.mod modification, is being used.

http://www.bank.lv/excel/valkurlv.php WEB-service provides data to the Supply.mod modification. The third pivot table row indicates that the DailySales.mod uses the InStockMovmVcREP report. MinPerLocREP is being created by the DailySales.mod modification, using the ArtCode, CustGroupName, ItemInStock and ItemMinLevel fields. The standard ERP's PUVc journal is being used by both modifications – the DailySales.mod and Supply.mod. The DailySales.mod modification reads data from the PUVc journal fields ArtCode, Location, Objects, Quant and RegDate, while the Supply.mod modification reads and writes data to the Location and RegDate fields. The standard ERP's SOVc journal is being used by the DailySales.mod modification, reading data from its ArtCode, BasePrice, Location and RegDate fields. The example pivot table also provides data that the Supply.mod modification receives data from the USB Desktop Reader UHF hardware device with SntItVc function and that the Supply.mod modification sends data to the VIDEDSPVN1 local software, using function ExportVATtoEDSREP function.

The Grand Total column displays data on how many functions or fields use the particular element, while the Grand Total row indicates how many times the particular field or function is being used.

Count of Field																
Mod. value	Mod. name	Type value	Type letter	Element name / address	-	ArtCode	BasePrice	CustGroupName	ExportVATtoEDSREP	ItemInStock	ItemMinLevel	Location	MinPerLocREP	Objects	Quant	RegDate	SntItVc	Grand Total
265	DailySales.mod	1	a	InStockMovmVcREP	1													1
		8	d	PUVc		1						1		1	1	1		5
				SOVc		1	1					1				1		4
		256	i	MinPerLocREP		1		1		1	1							4
150040	Supply.mod	8	d	PUVc								1				1		2
		16	e	PUVc								1				1		2
		512	j	DailySales.mod									1					1
		2048	l	http://www.bank.lv/excel/valkurlv.php	1													1
		16384	o	VIDEDSPVN1					1									1
		131072	r	USB Desktop Reader UHF													1	1
Grand Total					2	3	1	1	1	1	1	4	1	1	1	4	1	22

Fig. 4. Modification Review Pivot Table

Count of Fields / functions used																		
Element name / address	Mod. value	Mod. name	Type value	Type letter	-	ArtCode	BasePrice	CustGroupName	ExportVATtoEDSREP	ItemInStock	ItemMinLevel	Location	MinPerLocREP	Objects	Quant	RegDate	SntItVc	Grand Total
DailySales.mod	150040	Supply.mod	512	j									1					1
http://www.bank.lv/excel/valkurlv.php	150040	Supply.mod	2048	l	1													1
InStockMovmVcREP	265	DailySales.mod	1	a	1													1
MinPerLocREP	265	DailySales.mod	256	i		1		1		1	1							4
PUVc	265	DailySales.mod	8	d		1						1		1	1	1		5
	150040	Supply.mod	8	d								1				1		2
			16	e								1				1		2
SOVc	265	DailySales.mod	8	d		1	1					1				1		4
USB Desktop Reader UHF	150040	Supply.mod	131072	r													1	1
VIDEDSPVN1	150040	Supply.mod	16384	o					1									1
Grand Total					2	3	1	1	1	1	1	4	1	1	1	4	1	22

Fig. 5. - Modification Tracking Pivot Table

V. CONCLUSION

The main cause of ERP complexity are modifications, because after modifying the standard ERP without proper documentation or access to the source code a 3rd party expert cannot know how even the standard ERP parts work at present. Also it becomes impossible for the ERP partner to successfully manage large number of modifications without proper documentation because it becomes too hard to keep them all in mind and remember all the changes that have been made.

The formal method for documenting and analyzing modifications made in ERP systems has been elaborated in the paper. Nineteen types of modifications have been identified and a set of parameters characteristic of each type has been identified. These parameters are used to describe modifications. All modifications are stored in the register of modifications. For medium level ERP systems, it is suggested to store these modifications in a spreadsheet and to use typical spreadsheet analysis tools such as pivot tables as the main tool for analyzing modifications. That makes the proposed method accessible to small and medium size ERP implementation partner companies. Analysis of modifications allows identifying all modifications made into the ERP system, relationships between modifications and elements of the ERP system affected by modifications. This information can be identified without the need to analyze the source code of the ERP system. Obviously, description of modifications is time consuming though experience shows that otherwise modifications become unmanageable.

The main direction of future research is to tie together documentation and analysis of modifications with effort estimation in order to quantify impact of modification of efforts needed for implementation and maintenance of ERP systems.

REFERENCES

1. Aloini, D., Dulmin, R., Mininno, V.: Risk Management in ERP Project Introduction: Review of The Literature. J. Inf. Man. vol. 44, iss. 6: pp. 547 - 567 (2007);
2. Brehm, N., Gómez, J. M.: Web Service-Based Specification and Implementation of Functional Components in Federated ERP-Systems. Oldenburg, Germany (2007);
3. Crane, D.: Writing Cost-Effective Documentation for Software Systems, Indigo Byte Systems (2007);
4. Dor, N., Lev-Ami, T., Litvak, S., Sagiv, M., Weiss, D.: Customization Change Impact Analysis For ERP Professionals Via Program Slicing. ISSTA '08 Proceedings of The 2008 International Symposium on Software Testing And Analysis – ACM, New York, USA: 2008, pp. 97 – 108;
5. Esteves-Sousa, J., Pastor-Collado, J.: An ERP Life-cycle-based Research Agenda. In: 1st International Workshop on Enterprise Management Resource and Planning: Methods Tools and Architectures – EMRPS'99 – Venice, Italy: 1999, pp. 359-371;
6. Govers, M.J.G.: Summary PhD study: setting out for modern bureaucracies with ERP systems? Archypel, Netherlands (2003);
7. Light, B.: The Maintenance Implications of The Customization of ERP Software. . J. Softw. Maint. Evol.: Res. Pract. 13: pp. 415–429 (2001);
8. Luo, W., Strong, D. M.: A Framework for Evaluating ERP Implementation Choices. IEEE Transactions on Engineering Management, Vol. 51, No. 3, August 2004, pp. 322 – 323;
9. Outsourcing Product Documentation Reduces Project Costs, Anthrobytes Consulting, http://www.anthrobytes.com/cost_savings.pdf;
10. SAP Enterprise Support – ASAP to Run SAP, 2nd, SAP Press, 2010;
11. See Pui, Ng. C.: A Decision Framework For Enterprise Resource Planning Maintenance and Upgrade: A Client Perspective. J. Softw. Maint. Evol.: Res. Pract. 13: pp. 431–468 (2001);
12. Yachin, D.: SAP ERP Life-Cycle Management Overcoming the Downside of Upgrading. IDC (2009).

Mārtiņš Ebersteins is a PhD student at the Faculty of Computer Science and Information Technology, Riga Technical University, Latvia. He obtained his master's degree in Information Technology from Riga Technical University in 2010 and a bachelor's degree in Entrepreneurship from Banking Institution of Higher Education in 2009. He has more than 6-year work experience with ERP systems as a project manager and consultant. His research interests are in ERP system implementation, maintenance and ERP systems particularly in manufacturing and retail businesses.

Jānis Grabis is a Professor at the Faculty of Computer Science and Information Technology, Riga Technical University, Latvia. He obtained his PhD from Riga Technical University in 2001 and worked as a Research Associate at the College of Engineering and Computer Science, University of Michigan-Dearborn. He has publications in major academic journals including OMEGA, European Journal of Operational Management, International Journal of Production Research, Human Systems Management and others. He has been a guest-editor for two top academic journals and member of the program committee of several academic conferences. His research interests are in enterprise applications, supply chain integration and project management.