Measurement of the Standard Proximity of Adapted Standard Business Software

In order to optimize the economical use of SAP software systems and to increase the cost effectiveness of the SAP investment, the available SAP standard system should be implemented in the best possible way. This article describes a new technical measurement methodology based on key performance indicators allowing for measuring the standard proximity in SAP systems thereby creating transparency regarding the use of the available standard and non-standard functions. The methodology is evaluated in the context of a case study within the BMW Group. From the results of the measurement, actions are derived which foster the standard utilization and thus the efficiency of an SAP system.

DOI 10.1007/s12599-009-0045-4

The Authors

Dr. Eva Peggy Sekatzek

BMW Group Knorrstr. 119 80807 Munich Germany peggy.sekatzek@bmw.de

Prof. Dr. Helmut Krcmar

Technische Universität München Chair of Information Management Boltzmannstr. 3 85748 Garching Germany krcmar@in.tum.de

Received: 2008-06-23 Accepted: 2009-03-02 Accepted after two revisions by the editors of the special focus.

This article is also available in German in print and via http://www.wirtschaftsinformatik.de: Sekatzek EP, Krcmar H (2009) Messung der Standardnähe von betrieblicher Standardsoftware. WIRTSCHAFTSINFORMATIK. doi: 10.1007/11576-009-0165-y.

1 Introduction

In the past, the successful implementation of standard business software systems

provoked interest in both theory and practice (Chen 2001, p. 247). Although numerous success stories have been published, smooth implementation involves significant risks. First, higher costs are incurred by the introduction of standard business software systems. Secondly, the implementation has consequences on the enterprise (Sedera et al. 2003, p. 1075), including adaption to the new system processes, and activity profiles have to be adapted to the new software during its introduction. Further, the organizational structure of the enterprise has to change in response to the introduction of the new business software system (Hong and Kim 2002, pp. 27 f).

In some cases, however, companies take a different approach and adapt the software to meet their requirements, as opposed to adapting the processes (Beatty and Williams 2006, p. 108). Such changes are driven by the constantly changing environmental conditions a company is forced to deal with. In order to stay flexible, the enterprise must adapt standard business software to the changing requirements (Beatty and Williams 2006, p. 108). A further reason for system adaptation is the necessity to implement interfaces for integrating new software into the existing system landscape.

There are two kinds of adaptation to program code which generate increased maintenance efforts and thus additional costs. These are modification and extension (Bohr 2005; Markus et al. 2000, pp. 245 ff; Krcmar 2004). A substantial part of the resulting programming typically becomes obsolete over its lifetime due to changing requirements (Pütter 2006). When over time the changed requirements diverge considerably from the requirements defined at the beginning of the standard business software introduction phase, the individual developments and modifications are abandoned by their users. However, these unused programs remain within the software and are maintained and tested through each change or upgrade which generates unnecessary costs (Pütter 2006).

In order to identify and leverage cost saving potentials related to this in IT landscapes, enterprises should evaluate their standard business software on a regular basis. This article presents a new technical measurement methodology based on key performance indicators (KPIs) which increases transparency regarding the usage of available standard and non-standard components in standard business software. This methodology facilitates the measurement of the standard proximity of standard business software on a technical level. Although this article concentrates on SAP as exemplary standard software, practical hints are provided on how the measurement methodology can be applied to other standard software products. The methodology was evaluated in the SAP application landscape of the BMW Group in the context of a case study.



We shape the future

Deutsche Telekom Laboratories is Deutsche Telekom's research and development institute based in Berlin. It is simultaneously a scientific institute organized under private law and associated with the Technische Universität (TU) Berlin. At Deutsche Telekom Laboratories, scientists from across the globe work together with experts from the Group to develop new services and solutions for Deutsche Telekom's customers. Establishing new companies (spin-offs) is another method for making use of research output.

Cooperation with the TU Berlin, other universities and industry partners creates a bridge between business and science in order to turn ideas into marketable innovations as quickly as possible. As part of this, Deutsche Telekom Laboratories focuses on five fields of innovation (5 i):

- Intuitive Usability of services and devices
- Integrated Service Components
- Intelligent Access
- Infrastructure for IT and telecommunications
- Inherent Security

The business and information systems engineering offers useful interdisciplinary approaches for all these areas of innovation. Subject matter includes, for example, modeling, methods and tools for process innovations, agile architectures for information and communication technologies (ICT), technology-oriented management approaches and techno-economic assessments. The aim is to safeguard the economic sustainability of innovations for the Group.

Deutsche Telekom Laboratories is divided into two areas: The Innovation Development Laboratory focuses on market-centric research and development within a timeframe of up to three years. The basic and technology research of the Strategic Research Laboratory has a long-term focus. Common goal: Deutsche Telekom Laboratories is looking to become one of the world's leading research and development institutions in the field of new ICT.

An institute was set up together with Ben-Gurion University in Beer Sheva, Israel, in 2006. Since 2008, Deutsche Telekom Laboratories has also been represented in Darmstadt. Another project office was opened in the Silicon Valley, United States, in January 2009.

Contact:

Deutsche Telekom Laboratories Ernst-Reuter-Platz 7 10587 Berlin, Germany E-mail: wi.laboratories@telekom.de www.laboratories.telekom.com

Deutsche Telekom Laboratories

An-Institut der Technischen Universität Berlin



Time line

Fig. 1 Adaptation of an SAP system

2 Related methods and tools

2.1 Reverse Business Engineer (RBE)

The Reverse Business Engineer was developed by the SAP AG in cooperation with IBIS Professor Thome AG. This tool was developed to automate the reverse business engineering process. SAP systems are analyzed in order to determine a retrograde model of the actively used system parts. The method helps enterprises to analyze and document their SAP systems (Hufgard and Wenzel-Däfler 1999).

2.2 SAP Custom Development Optimization Package (CDOP)

The SAP Custom Development Optimization Package was developed to achieve the following goals (SAP AG 2007a, pp. 32 ff):

- Create transparency with regard to custom developments and modifications;
- Pinpoint the dependencies of new release versions on custom developments and modifications;
- Contribute to the acceleration of release upgrades.

These two tools, the Reverse Business Engineer and the SAP CDOP, contain analysis tools which help to identify empty databases, unreferenced objects and unused custom developments. The dependencies of custom developments on new release versions can be pinpointed with the help of synchronization functionality. The main advantage of both tools is that they provide numerous analysis functions to support release upgrades (SAP AG 2007a, p. 7). Both tools focus on analyzing the entire data of an SAP system and generate a large amount of data.

Both tools have disadvantages. For example, neither tool can support the archiving of data and consequently the tracking of changes during the course of time¹. More importantly, a major drawback of both tools is their lack of single key performance indicators.

Single indicators providing top management with summarized information about a system's status is essential (Dietrich and Schirra 2004; Geiß 1986, pp. 42– 47; Siegwart 1998). In particular, in the field of IT controlling there is a demand for (Siegwart 1998):

- tracking the system status over time,
- comparing several systems within one company,
- comparing the systems of different companies.

Management teams are usually unable to directly interpret the huge amount of information provided by commonly known upgrade tools. Practical experience gained in the course of this research shows that there is an unsatisfied demand for aggregated key figures that improve transparency regarding the standard proximity of standard business software landscapes.

The methodology presented here differs from the classification method of RBE and CDOP in a number of ways. In RBE and CDOP only two categories of development objects exist: standard and custom development (non-standard). This classification is performed based on the object names. Practical experience shows that there is a need for a more detailed classification method for the following reasons:

- If a standard object is modified in RBE and CDOP, its name is not affected. Consequently, the object will not be identified as "non-standard", but misleadingly as "standard".
- If a standard transaction calls such a modified object, the CPU workload would misleadingly be added to the class "standard", whereas it actually belongs to the class "modification".
- A transaction in the customer name space which calls an SAP standard program would misleadingly be rated as custom development, although it should in fact be classified as standard.
- It is essential to differentiate between modifications and custom developments. Modifications are even more critical than custom developments due to the fact that they change the standard functionality of the software systems.

In summary, basing the calculation of the standard proximity on the object names alone cannot provide adequate information about the volumes of standard functionality, modifications and custom developments in a system due to the occurrence of special cases. More importantly, the indication of standard usage would degrade to highly inaccurate values over time.

The central contribution of the methodology presented here is to provide aggregated yet expressive key figures, an improved classification methodology, and the possibility to identify modifications. The advantages of the methodology are described in more detail in the following.

3 Measurement methodology of standard proximity

The functionality of an SAP system can be described by its standard proximity. The standard proximity specifies to what degree the functionality is covered by the SAP standard. In a near-standard SAP system, the functional requirements are covered to a large extent by the standard and thus the SAP systems are only slightly adapted. In this context, the business pro-

¹ The evaluation of RBE took place in 2006/2007, evaluation of CDOP in 2008.

Self-development

cesses are also implemented via standard functionality. If functional requirements can only be partly covered by the standard, the system must be adapted whereby the standard proximity of the system decreases (SAP AG 2001, p. 8).

Such adaptations are advantageous if they optimize the implemented process. In practice however, it is frequently the case that adoptions are hardly used. This then makes it necessary to identify unused adaptations to leverage the advantages of near-standard over non-standard SAP systems. The advantages of near-standard over non-standard SAP systems according to Appelrath and Ritter 2000; Light et al. 2001; Markus et al. 2000 are:

- Reduction of total operating costs of SAP systems,
- Potential consolidation of multiple SAP systems,
- Long-term securing of operating ability,
- Long-term management of the investment,
- Optimized manufacturer support and improved utilization of standard processes.

Accordingly, it makes sense to measure the standard proximity and to define suitable actions for increasing it.

Periodic measurement becomes essential because the adaptation of an SAP system typically starts with either its distribution or its introduction and continues as long as the SAP system is in use and functional requirements change. The standard proximity varies with each adaption of the system's functionality (**Fig. 1**). Measurement of the success of the actions on a regular basis is necessary to achieve a seamless embedding of the system company-wide and to achieve a sustainable improvement of the total operating cost.

The overall functionality of an SAP system can be divided into the three classes standard, modification and custom development which are based on the degree of adaptation (**Fig. 2**) (Appelrath and Ritter 2000, pp. 64–67; Stahlknecht and Hasenkamp 2005, p. 289). At the time of purchase, the entire functionality of the system is within the standard class and therefore marks the baseline for measuring the standard proximity. The standard proximity then decreases with every modification or custom development.

So far only the provided functionality has been considered. In practice there is a difference between implemented and used



Standard



functionality. Within most standard business software systems less functionality is actually needed than provided by the software supplier (**Fig. 2**).

It is important to analyze the standard proximity from two perspectives. The first point of view describes the standard proximity related to the entire functionality, i. e. the degree of coverage of the functional requirements by SAP standard (standard coverage degree). The productive part of the functionality is described by the second point of view; the degree to which the SAP standard is really used (standard usage degree).

Both aspects are necessary in order to draw a realistic picture of the usage of the system. Measuring only the standard coverage would not result in a holistic picture because information about the used functionality would be missing. A statement whether the modifications of a system are used and thus meaningful cannot be based on the standard coverage degree but rather on the standard usage degree. In cases where the standard degree of usage is high, it can be assumed that the modifications in a system are justifiable. However, if the modifications are not used they serve only to increase the complexity of the system.

Contrastingly, an isolated view of the standard usage would not permit a conclusion about the used functionality. Two possible scenarios are:

Example 1

The functionality of system A is implemented by five applications, four of which are SAP standard and one is self-developed. The standard coverage degree in this system is 80 %. The custom development was called four times in the observed period while only one standard application was used. If we assume that each application needs the same computing time for its execution, the standard usage of the system is only 20 %.

Example 2

In system B, eight standard and two custom-developed applications are used.

If each application needs an equal amount of computing time for execution, the standard usage degree is 80 %. The system contains another five custom-developed applications as well as a further five modified standard applications. The result is a standard coverage of only 40 %.

Modification

Used functionality

While unused standard functionality itself has no negative effects on maintenance and operating cost, unused modifications are cost drivers (Markus et al. 2000, p. 259). There are several reasons for this. First, the implementation of a modification incurs costs and each modification generates subsequent costs over the remaining life cycle of an SAP system (Mabert et al. 2003, p. 238; Markus et al. 2000, pp. 245 ff). During each system upgrade all modifications have to be transferred to and tested thoroughly in the new release environment (Mabert et al. 2003, pp. 238 ff). Customizing, which is the third possible option to adapt a standard software system, belongs to the "standard" category due to the fact that it has no negative effects on system upgrades.

One reason for unused custom developments and modifications are changed requirements. Because of the long lifecycle of SAP systems, the modifications are often implemented several times in parallel within a system over its lifecycle. A lack of communication between IT and business divisions is a further reason why expensive modifications are abandoned.

Moreover, a lack of understanding and time and budget pressures make it difficult for system and maintenance personnel to identify and remove unused modifications during a release change. Unused functionality thus remains within the system causing the same problems during the next release change.

3.1 Classification methodology

The subject of the measurement is the standard proximity and the key performance indicators for standardization based on the development objects in the repository of an SAP system. The following description refers to SAP ERP systems; however, the measurement methodology



Fig. 3 Simplified assumption for the solution space



Fig. 4 Linkage of basic and detail data

can be easily adapted to other standard software products (see chapter 5).

SAP R/3 is divided into different modules each of which covers part of a business process. The highest level module provides groups such as accounting, logistics and personnel management (Chen 2001, p. 374; Loos 2000, p. 685; Lebedinski et al. 2006, pp. 15 ff). The individual modules are further divided within these module groups. The module group accounting, for example, contains the modules finance, controlling and project system. The modules themselves are again divided into individual sub modules (Kösegi and Nerding 2005, p. 34).

A module consists of multiple development objects, e. g. programs, functional modules, tables, screen masks, and data types. The development objects are grouped into packages (Ackermann 2000, p. 2). A package is a construct which contains individual development objects and/or other packages. Each development object is part of exactly one package and a package can be contained in one other package at the most. A main package is a package which is contained in no other package (i. e. a package without a parent package). Each package can appear in the role of the offerer (= server) as well as of the user (= client) of a service. These three characteristics of packages provide the possibility to divide and encapsulate R/3 into distinct technical units which reduces dependencies and increases the comprehensibility of the system (Ackermann 2000, p. 2).

The measurement procedure is based on three principles (**Fig. 3**):

- The capacity of an SAP system can be structured into modules and packages.
- All development objects are administered in a central repository.

 All development objects can call other development objects.

The standardization degrees are calculated based on the development objects and their characteristics. Constraints are established to restrict the amount of different possible scenarios in view of the multitude of different development object types.

Not each development object is suitable for the calculation of the key performance indicators. For instance, those that are only used in the context of a closed application domain do not affect the standardization degree of the SAP system. Examples of such development objects are customizing objects. On the other hand, instances of many specialized development object types exist only in small amounts. Although these would only marginally affect the calculation results, the effort needed to analyze them would be better spent on the analysis of development object types which occur with a higher frequency.

Tab. 1 shows an excerpt of the repository of a SAP reference system (IDES system) which lists the 20 most frequent development object types. To create this list the SAP repository, which contains all development objects, was arranged by the different types. **Tab. 1** also shows the frequency of each development object type. In summary, the reference system registers approximately 1.3 million development objects and 274 different development object types.

The listed development object types represent nearly 90 % of all objects in the repository. The list contains all classical development objects such as data elements, tables and programs which affect the standardization degree. In addition, there is a multitude of types which are less or not at all significant, e. g. text elements or customizing objects.

To calculate the standardization degrees, each development object was assigned to one of the following classes: standard, modification or custom development. This classification is done by a novel method described in the following.

The name of a development object is part of a name space based on the creator of the object. Consequently, the standard SAP objects lie in the SAP name space and the internally developed objects lie in the so-called customer name area (Kösegi and Nerding 2005, p. 237). The allocation to a name area suggests one of the classes standard or custom development

relative

19.1 %

32.1 %

44.5 %

50.8 %

55.0 %

58.9 %

61.9 %

64.9 %

67.6 %

70.2 %

72.6 %

74.9 %

77.2 %

79.5 %

81.8 %

83.6%

85.4 %

86.8 %

87.7 %

88.4 %

89.1 %

Cumulated frequency

olute

1,135,154

1,146,933

1,156,284

1,165,363

provided there are no modifications and the name space concept is kept. The novelty of the methodology presented here is that for the clear identification of opment object in an SAP system nical name of a development obj beside the name area. This is mo the realization that the name a is not sufficient for an unambig tinction between standard and tion because development obje SAP name space can be modifie changing their name. Thus, wit sification based on the name an these objects would be mistake to the standard as opposed to th cation class (see chapter 2).

SAP systems have reserved na for customer objects and SAP ol customer developments, the ini Y and Z are pre-determined if space concept does not apply (K Nerding 2005, p. 237). The stand tionality of SAP standard con number of objects which fall int tomer name range. The technical name offers a possibility to assign customer developments to the class custom development. The field development object type describes the kind of a development object. The classification can be anticipated by this characteristic in some cases as shown in the following two examples:

- Function group with customer includes SAP part: This is an expandability planned in the standard within the SAP extension concept and is classified as standard.
- Function group with customer includes customer part: This is the arrangement by the customer and thus a custom development.
- In the classification approach presented here, the author and the last editor are noted for each development object in order to differentiate customer objects and modifications from standard objects. To this end, three rules can be formulated to accomplish the classification based on the change history of the development object:
- If the author and last editor are of type SAP standard developer, the development object is classified as standard.
- If the author is of type SAP standard developer, but not the last editor, the development object is classified as modification.
- If neither the author nor the last editor is of type SAP standard developer, the

		absolute	relative	absolute
of a devel- in the tech- ect is used otivated by area alone guous dis- modifica- ects in the ed without h the clas- rea alone, nly added ne modifi-	Data element	249,655	19.1 %	249,655
	Table	169,853	13.0 %	419,508
	Program	162,276	12.4 %	581,784
	Transaction	82,672	6.3 %	664,456
	Chapter of a book structure	54,534	4.2 %	718,990
	Domain	50,847	3.9 %	769,837
	Table type	38,691	3.0 %	808,528
	Customizing IMG activity	38,641	3.0 %	847,169
	Customizing attribute	35,464	2.7 %	882,633
	Customizing transactions	33,679	2.6 %	916,312
	Function groups	31,871	2.4 %	948,183
me ranges bjects. For tial letters the name Gosegi and lard func- mprises a to the cus-	Definition of care- and transport-objects	30,639	2.3 %	978,822
	Classes (ABAP objects)	30,434	2.3 %	1,009,256
	Views	30,419	2.3 %	1,039,675
	Generic structure clipboard: definition of a structure	30,199	2.3 %	1,069,874
	Documentation	23,797	1.8 %	1,093,671
	Generic text	23,045	1.8 %	1,116,716

Frequency

18,438

11,779

9,351

9,079

1.4 %

0.9 %

0.7 %

0.7 %

Development object types

development object is classified as custom development.

Information object of MIME repository

Test case

Search item

Solution map: process

A special case of this classification scheme of development objects are add-on components from third party suppliers which are always classified as custom development.

3.2 Technical implementation of the classification methodology

This section gives an overview of the necessary system tables which contain the characteristics described above and explains how the data of the system tables are related to each other. The analysis and classification of a development object is usually based on two system tables. The basic data of a development object can be found in the table repository objects catalogue (technical name: TADIR) and comprises three attributes:

- Transport and development object type, data fields PGMID and OBJECT;
- Technical name, data field OBJ_ NAME;
- Package (or development class), data field DEVCLASS.

Detail data is stored in development object type specific system tables which make a linkage of these system tables necessary in order to analyze the described characteristics. A linkage of the two system tables is accomplished by a copy of the development object type to the system table names. The navigation to the detail data record of the development object is performed via the technical name as well as further typespecific attributes, if necessary. In some cases, special name concepts are used for the technical name which makes a direct linkage from basic to detail data more difficult. This is illustrated in Fig. 4.

The measurement of the standard proximity and its KPIs follows the simplified flow chart in Fig. 5.

The application statistics are logged as part of the system workload and contain the number of calls of each application in the system. The system load can be displayed by calling transaction ST03. Basis of the system load monitor is the system table MONI in which the data are stored in aggregated form. The component SAPWL_GET_SUMMARY_STATISTIC



Fig. 5 Flow-chart of system measurement







Fig. 7 Result levels of detail

provides an RFC interface for the extraction of this data.

Dialogue-based entries are then extracted from the application statistics and form the basis of the measurement. The module usage which determines the used capacity of the SAP system (a requirement for the calculation of the standard coverage degree), can be calculated this way. Furthermore, the data is needed for the computation of KPIs of the standard usage degree as well as for the identification of unused modifications and custom developments.

Based on data excerpts of the application statistics, the module usage can be determined in a bottom up fashion in order to identify the used parts of an SAP system. Subsequently, all development objects belonging to these modules can be determined by the reverse analysis (top down) (**Fig. 6**).

Both procedures (bottom up and top down) use the package usage list as an intermediate step. The entries of the application statistics can be assigned to development objects in the repository either directly or indirectly. Afterwards the packages which contain the development objects can be identified (in this case the ABAP programs). For each package the associated software component can be identified in the SAP system. The software component groups all development objects which belong to a sub module into packages.

From an economical point of view, the result of the measurement of the module usage is an exact determination of the actually used functionality within an SAP system. From a technical perspective, the result is a list of all relevant development objects. The extracted data of the development objects is completed by their basic data and extended by their detail data. With this level of detail, the KPIs can be calculated.

3.3 Formulas for the key performance indicators

As explained, the development objects are divided into the three classes: standard, modification and custom development. Let Kx, with $x \in \{$ standard, modification, custom development $\}$ be one of these classes. Then the standard coverage degree (SCD) is calculated as follows:

$$\frac{\text{STAG}}{|K_{\text{Standard}}|} = \frac{|K_{\text{Standard}}|}{|K_{\text{Standard}} \cup K_{\text{Medification}} \cup K_{\text{Self-Development}}|} [\%]$$

The formula calculates the standard coverage degree based on the cardinality of the individual classes without an evaluation of the individual development objects.

By extraction of the application statistics, the used ABAP programs can be determined.

Let $Kx^* \subset Kx$ be the amount of called ABAP programs in the observed period and $p \in Kx^*$ a called program. The function t(p) returns the accumulated CPU time for the called program in time units

$$T_x = \sum_{i=1}^{n} t(p_i)$$
 [ZE] for $n = |K_x^*|$

is the accumulated CPU time of the entire class Kx*. After the accumulated CPU times of the three classes are computed, the standard usage degree (SUD) can be calculated in accordance with the formula:

 $\begin{array}{ll} {\rm STNG} &= \\ {T_{\rm Standard}} \\ {T_{\rm Standard} + T_{\rm Modification} + T_{\rm Self-Development}} \left[\% \right] \end{array}$

4 Case study: measurement results and actions

The methodology presented here has been rolled out within the BMW Group in order to evaluate its practicability. The two key figures SCD and SUD are part of a larger KPI system for increasing the control and transparency of SAP application landscapes. This system was established in the BMW Group between June 2005 and November 2006 (for further information see Sekatzek and Krcmar 2007, pp. 271–288; Sekatzek 2008) and comprises several technical as well as economical key performance indicators. The two key figures SCD and SUD are examples of technical key performance indicators. The economical influence of standard proximity, which is described in detail in the complete KPI system (Sekatzek and Krcmar 2007, pp. 271-288; Sekatzek 2008), was not discussed in this article.

As a manufacturer of luxury cars the BMW Group needs efficient and effective support of its business processes which can only be achieved by using adequate standard business software products. Currently, the BMW Group's SAP system landscape comprises about 200 SAP sys**Fig. 8** Overview of measurement results and implications



SAP user

. Interface

IT

SAP user

SAP key user

(per SAP modul)

Central key user

(per SAP modul)

Support teams

(per SAP modul)

Decision committee

SAP user

tems, 50 of which have to do with automobile production. This is one reason why key performance indicators are essential for steering and controlling the systems landscape.

For purposes of evaluation, the methodology was implemented in ABAP. **Fig. 7** shows the measurement results divided into three levels of detail. On the first and third levels, the key performance indicators are depicted as top level key figures:

- SCD of the system
- SUD of the system

On the second level, the key figures are split into more detailed information:

- Names of modules which have been used in the last period,
- Number of modifications per module,
- Number of custom developments per module,
- SCD and SUD per module.

The information on the third level is the most detailed:

- Names of modified objects,
- Names of custom developments,
- Location of these objects (module name),
- Object type,
- Usage of the object within the last period.

Fig. 8 shows the technical implication derived from the measurement results of each level. Some economical questions are also depicted to give an indication of further fields of research (see chapter 6).

During the case study, different use case scenarios for the measurement results emerged which were used by stakeholder groups. Some of these were:

1. Level 1 – Comparison:

- A comparison between an old and a new SAP system was performed in order to show if a higher functional usage and a higher standard proximity could be reached by the new implemented system (see Fig. 7).
- A comparison of each SAP system's status over time was performed in order to



identify in which systems the complexity increases.

 A comparison between several systems within a template was done in order to identify systems which differ from the template and systems in which functionality is used.

This information was used by the SAP system owners and the IT management.

2. Level 3: Identification of unused modifications:

Data on level 3 showed the modifications and custom developments which were used within a period of three months (default value in SAP systems). By repeating the system measurement after this period, a tracking of permanently unused objects was possible. However, experience showed that there might be objects which are rarely used which would have to be treated separately. This information was requested by the application operation teams.

3. Level 2 – Identification of consolidation candidates:

The measurement results gave an overview of the module usage of all SAP systems

Tab. 2 Connection between standard usage and coverage				
	Low standard coverage degree	High standard coverage degree		
Low standard usage degree	 Strongly adapted SAP System with a multiplicity of intensively used adapted objects Actions: Indicates that individual software could be the better option Business process reengineering to adopt processes which are customary within the industry 	 Less adapted but intensively used modifications/custom developments Actions: Consolidate SAP systems to increase system productivity Business process reengineering, to adopt processes which are customary within the industry 		
High standard usage degree	 Strongly adapted SAP system with a multiplicity of unused custom developments Actions: Overwrite unused modifications with SAP standard Reduce unused custom developments from the SAP system 	 Less adapted SAP system, few custom developments Actions: Define actions to maintain the system status over time 		

in the company. A comparison of these module usage profiles showed which SAP systems implement the same functionality and therefore could be consolidated. This information was useful for the IT application architects.

From the results of the Level 1 comparison, four different system types were extracted which are arranged by their SUD along the vertical axis and their SCD along the horizontal axis (**Tab. 2**). In each of the quadrants suitable actions are shown.

Tab. 2 further illustrates the reason why both KPIs are necessary for a holistic interpretation and derivation of feasible actions. On the one hand, a low standard coverage degree indicates that the system contains many adaptations but gives no information about the degree of usage of the modifications and add-ons. Only in combination with the SUD is it clear if the modifications have been used.

In addition to this case study within the BMW Group, an empirical case study within the German automotive industry was performed to evaluate additional actions to support companies in reducing their system complexity by increasing standard proximity (for further details see Sekatzek 2008).

For example, the establishment of a key user organization is one alternative approach to the preservation of standards. To this end, a module-specific SAP key user is assigned to each SAP user, as shown in **Fig. 9**. If a SAP user needs a software modification or an extension, he informs the appropriate key user. Ideally, The SAP key user is a readily available and knows both the business and the SAP processes. He is thus able to estimate the necessity of a modification and is authorized to reject it if necessary. Furthermore he identifies the training needs of single users and forwards their requirements to a central key user who consolidates the requirements of different key users. This is advantageous in that similar requirements can be grouped thereby avoiding a duplication of training exercises. For each module there is typically one central key user. Following consolidation, the key user forwards the requirements to a support team and subsequently then to a committee which decides on either the confirmation or refusal of a modification.

After the introduction of a system introduction there are several possibilities of guaranteeing the standardization of an SAP system on a longer term basis. Through the use of regular audits in which the implemented processes are adjusted to the SAP standard processes, the standard can be optimized. Fig. 10 depicts the procedure of a process improvement audit. The total amount of SAP functionalities is divided into the number of used and unused SAP functions (based on the results shown in Fig. 8). Whereas the business users only know the processes they have used, it is the task of the responsible personnel to point out unused functionalities within the SAP system.

The transfer of modifications and custom developments into the SAP standard is a further possibility of increasing the standardization degree of an SAP system. However, this is clearly a long-term action because alignment with the SAP release functionality is necessary and might be connected with long-term consolidation efforts.

5 Summary and discussion

This article presents a measurement methodology on the standard proximity and the modification degree of an adopted SAP system. Modifications are cost drivers in SAP systems not only because of the effort required for their implementation, but also because additional maintenance costs are generated by the modification and testing of each release upgrade.

Due to the complexity of the subject as well as a lack of existing methods which could fulfill the requirements, a new measurement methodology was developed which enables a more detailed calculation of standard coverage and degree of usage.

The novelty of this approach has two aspects: Firstly, it allows an accurate identification of modifications and custom developments taking special cases into consideration. The categorization of the development objects by means of their change history improves the quality of the classification results in comparison to former classification methods by object names. Consequently, modified objects with SAP standard names can be identified.

Secondly, it provides a variety of aggregated yet expressive key figures to be used by IT controlling and top management to track the change of one system's standard proximity over time as well as to compare different systems.

There is still room for further improvement of the key figures SCD and SUD. The complexity of the process of data collecting makes the communication of the key figures difficult for the occasional user. While the key figure "total number of modifications" can be understood without an additional explanation, the SCD and SUD cannot be interpreted without background information on the method of collection of data. Simplicity alone cannot be the main measure for the usability of a key figure. The key figure "total number of modifications" makes a comparison of different systems impossible due to the lack of the used functionality of a system. With the help of SCD and SUD, the amount of modifications and unused modifications are put into perspective of the used system. This allows the comparison of different systems and the identification of changes over time.

Lastly, the generalizability from SAP to other standard software systems can be explained as follows. The method consists of the classification method of objects and the formula for the calculation of the two key figures. While the formula can be easily generalized for any standard business software, the classification method must



Fig. 10 Process improvement audits

be adjusted if it is to be applied to standard software. The assignability is provided only if the software system consists of object groups grouped into packages and if each object has a change history which tracks the last editor.

6 Future research

Fig. 8 shows several research questions which will be analyzed in future research. By focusing on only one quantifiable technical aspect the standard coverage or usage degree – the next step should be to measure the technical and economical implications of system adoption. Some interesting aspects to be addressed are:

- quantifying the implication of the standard coverage degree on system upgrade cost,
- quantifying the implication of the standard coverage degree on system upgrade duration,
- quantifying the implication of the standard usage degree on system performance,
- (average system response time), and
- quantifying the implication of the standard usage degree on system availability.

There is still plenty of work to be done in this research area. The authors are not aware of many examples of previous research which quantified the above aspects using a valid methodical approach. This is indeed surprising, since these aspects contain a lot of relevant information which could greatly influence the future adaptation habit of companies.

There is a need for research on the time before system introduction, namely on requirements management. For example, how requirements management can support the prioritization of requirements in the requirements phase of standard business software projects. Using a method which reduces the difficulties in communication between IT and business departments, unnecessary modifications and custom developments could be reduced starting at the very early stages of system introduction. The savings could be reinvested into the actually used functionality for further business process improvement.

References

Ackermann J (2000) Das SAP-Paketkonzept – Erfahrungen bei der Modularisierung bestehender Anwendungssysteme. Walldorf

- Appelrath HJ, Ritter J (2000) R/3-Einführung: Methoden und Werkzeuge. Springer, Heidelberg
- Beatty RC, Williams CD (2006) ERP II: best practices for successfully implementing an ERP upgrade. Communications of the ACM 49(3):105– 109
- Bohr D (2005) Benchmarkingstudie: Verschwendung bei SAP-Installationen – Wo liegen versteckte Einsparungspotenziale? West Trax/ Experton Group, Hahnstätten
- Chen IJ (2001) Planning for ERP-systems: analysis and future trends. Business Process Management 7(5):374–386
- Dietrich L, Schirra W (2004) IT im Unternehmen: Erfolgsbeispiele aus der Praxis. Springer, Heidelberg
- Geiß W (1986) Betriebswirtschaftliche Kennzahlen. Theoretische Grundlagen einer problemorientierten Kennzahlenanwendung. Schriften zum Controlling, Vol. 1, OHG, Frankfurt/Main
- Hong KK, Kim YG (2002) The critical success factors for ERP implementation: an organizational fit perspective. Information & Management 40(1):25–40
- Hufgard A (1994) Betriebswirtschaftliche Softwarebibliotheken und Adaption. Vahlen, München
- Hufgard A, Wenzel-Däfler H (1999) Reverse Business Engineering Modelle aus produktiven R/3-Systemen ableiten. In: Scheer AW, Nüttgens M (eds) Electronic Business Engineering,
 4. Internationale Tagung Wirtschaftsinformatik, Physica, Heidelberg
- IBIS (2007) Informationsbroschüre: Reverse Business Engineering. http://www.ibis-thome.de/ fileadmin/website/PDF/deu/RBE_Plus_Informationbroschuere/RBE_Plus_Informationbroschuere.pdf. Accessed 2007-10-31
- Kösegi A, Nerding R (2005) SAP Änderungs- und Transportmanagement, 2nd edn. Galileo, Bonn

Abstract

Eva Peggy Sekatzek, Helmut Krcmar

Measurement of the Standard Proximity of Adapted Standard Business Software

In order to optimize the economical use of SAP software systems and to increase the cost effectiveness of the SAP investment, the available SAP standard system should be implemented in the best possible way. While standard functionality has no negative effects on maintenance and operating cost, modification of the standard has the potential to drive costs (Markus et al. 2000). An integrated methodology and an efficient instrument for the operational and strategic evaluation of this topic are crucial for an efficient application of SAP standard software. This article describes a new technical measurement methodology based on key performance indicators. This measurement methodology allows for measuring the standard proximity (i. e. proximity of used functionality to standard functionality) in SAP systems thereby creating transparency regarding the use of the available standard and non-standard functions. The methodology is evaluated in the context of a case study within the BMW Group. From the results of the measurement, actions are derived which foster the standard utilization and thus the efficiency of an SAP system.

Keywords: Standard business software, SAP, Standard proximity, Standardization degree, Key performance indicator, Efficiency

- Krcmar H (2004) Informationsmanagement, 4th edn. Springer, Heidelberg
- Lebedinski L, Zeilinger S, Grechening T (2006) SAP tutorial. INSO, Wien
- Light B, Holland CP, Wills K (2001) ERP and best of breed: a comparative analysis. Business Process Management 7(3):216–224
- Loos P (2000) Advanced information technology application in ERP systems. In: Chung HM, AM-CIS 2000. Long Beach, 635–639
- Mabert VA, Soni A, Venkataramanan MA (2003) The impact of organization size on enterprise ressource planning (ERP) implementations in the US manufacturing sector. Omega 31(3):235–246
- Markus ML, Axline S, Pertrie D, Tanis C (2000) Learning form adopters' experiences with ERP: problems encountered and success achieved. Journal of Information Technology 15(4):245– 265
- Pütter C (2006) CIO Online: Jedes dritte Programm wird kein einziges Mal aufgerufen. http://www.cio.de/knowledgecenter/ erp/820496/index.html. Accessed 2007-11-20
- SAP AG (2001) Änderungen des SAP-Standards. SAP, Walldorf
- SAP AG (2007a) SAP standard eSOA readiness version 1: clearing or standardization of repository enhancements. SAP Whitepaper, SAP Walldorf SAP AG (2007b) CDOP Masterguide. SAP, Walldorf
- Sedera D, Gable G, Chan T (2003) ERP success: does organisation size matter? In: 7th Pacific Asia conference on information systems (1075– 1088). Adelaide
- Sekatzek P, Krcmar H (2007) Ein Kennzahlensystem zur Optimierung von SAP-Systemen. In: Oberweis A (ed) 8. Internationale Tagung Wirtschaftsinformatik, Vol. 2. Wirtschaftsuniversitätsverlag, Karlsruhe, 271–288
- Sekatzek P (2008) Einsatzentscheidung und -steuerung von SAP Standard Business Software in der deutschen Automobilindustrie. Dissertation, TU München
- Siegwart H (1998) Kennzahlen für die Unternehmensführung, 5th edn. Haupt-Verlag, Bern
- Stahlknecht P, Hasenkamp U (2005) Einführung in die Wirtschaftsinformatik, 11th edn. Springer, Heidelberg