Paradigm Change: Aligning Business and IT with a Business Software Integration Method

Susanne Altendorfer

Industrial Logistics, Montanuniversität Leoben
Franz-Josef-Strasse 18, 8700 Leoben, Austria
susanne.altendorfer@unileoben.ac.at

Abstract. Times change and so do business requirements: Within the years the requirements to do business have changed and got even more complex. Thus the companies' structures have to respond to these new situations with changing the way they are doing business. Traditional structures are no longer appropriate. Therefore a business reengineering is normally not enough. A software reengineering is also necessary to support the business processes in the best way. Business Software is here the key. This paper presents the paradigm change from monolithic standalone software to business software that ideally supports the business processes and thus helps streamlining the processes. It fosters the implementation of the term business software, especially in the technical area and describes the underlying software architecture. Furthermore the advantages of business software are outlined within the automotive industry.

Keywords: business processes, process streamlining, business software, software modeling, business software integration.

1 Introduction and Motivation

Information systems are a crucial factor for companies in all lines of business. Within the years the requirements to do business have changed and got more complex. [18] Thus the companies' structures have to respond to these new situations with changing the way they are doing business. Traditional structures are no longer appropriate. Over the years organizations with traditional structures happened to create business departments as silos: They became big and shore less [9].This leads to blockades and information-silos. Therefore new ways of managing the organizations have to be found. Business Reengineering is one way to optimize the business processes according to the company's requirements responding to customer needs [9]. Especially the information system structure has to adapt to the new situation. A business process reengineering encompassed by a radical architecture change is necessary to gain long-lasting improvements.

The concept of IT as enabler of process change dates already back to [10] and [6] and is still on the forefront of process reengineering. A change in information systems is possible as the technological progress over the years has opened new possibilities to support the organizational reengineering. However, information system aspects have

W. Abramowicz, J. Domingue, K. Węcel (Eds.): BIS 2012 Workshops, LNBIP 127, pp. 104–115, 2012.

often been left out of consideration in reengineering projects. Information systems often have had and still have the status of being a matter of course and therefore their integration is often not thoroughly considered [11]. Thus companies are far too often behind in their information technologies. Software systems are often out-dated and poorly structured. Thus it is impossible to react with such systems to organizational changes. The need for agile software architecture to support business processes becomes evident when the need for more flexibility and reduced costs in the daily business urges companies to restructure the processes. Especially a system adaption is necessary to support the restructuring of the processes otherwise there will always be a break in the organization's structure. It, however, is often not enough to stick to a system adaption.

Although it is obvious that any reorganization project has to start with the company's strategy and should impact the business structure and strategy, it should not stop at this point. Fig. 1. illustrates that the components business strategy, business reengineering and software reengineering are ideally combined like gear wheels. Changes in any of the components influence the other parts, too.

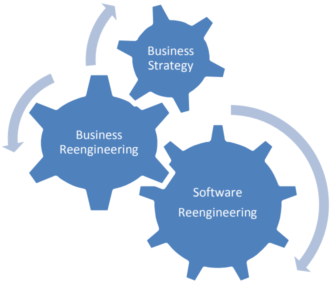

Fig. 1. Reengineering Interactions

Although the business strategy is mainly the impulsive force the other two components support the strategy to be fulfilled when they interact as a closed system. This is often left out of management considerations. The information systems part (illustrated as 'software reengineering') is not considered properly when changing the business strategy and structures, like processes and workflows, although the software influences the processes and the work of the employees to a high impact.

As technologies enable new software architectures, it makes sense to introduce not only standalone software systems but business software to support business processes. In this context Sundblad [19] states that business software is often introduced for exactly one reason: It should support the business and its activities to increase the productivity and efficiency of the business. The advantage of business software lies in the fact that business software, contrary to standalone software, can be integrated in all relevant business processes – like sales, production, after sales - to get a higher scale effect. Exactly this shift to business software means a huge paradigm change for companies all over. It is often not that easy to step into such a radical change project as the results are not easily predictable and case studies on a long time horizon are not available. However, introducing business software would support companies in restructuring their business and making their processes more efficient. The most known and widespread business software is ERP (Enterprise Resource Planning) systems [23]. However, business software is not only restricted to ERP systems though a concentration on those systems is recognizable, due to the fact, that ERP systems are very common no matter from which provider, and the integration cycle is documented and rather similar no matter what's the company's business. Business software, however, can also be implemented for technical software systems. Thus a better process integration and support would be granted. A big hindrance in this area nowadays is that almost no examples of business software integration in the technical area exist.

This paper will therefore deal with this topic and investigates on the paradigm change to business software in the technical area within the automotive industry. Especially in the automotive industry many technical software systems exist, which could be replaced by integrated business software, which could then have a high impact on the workflow of the business processes. It should be noted, that the paper contains a brief outline and short description of a prototypical example of the specific business software to support the business processes in a very special use case, as the work is still in progress. Nevertheless it shows the general potential business software has on a process value chain within any organization.

2 Monolithic Software and Business Software Impacts

Many software systems are used over decades, with just slightly adapting the system. Business needs are not really considered in depth but only to a point where adaptations to existing systems are risky and dangerous. This fact is not a phenomenon within the automotive industry. This is quite a common phenomenon. Raisch [16] describes this phenomenon as "*Ambidexterity*". This means that an organization differs between two cycles of software reengineering: (1) the Exploration and (2) the Exploitation phase. The former signifies a phase where new systems are tested and introduced. This phase, however, requires a high level of readiness to assume risks, as many unknown components are predominant in this phase. The latter concentrates on exploiting existing systems as long as possible. Here just small and almost insignificant adaptations are made at the software systems. This means no risks and no changes to existing workflows. Fig. 2 illustrates how the two phases interchange each other. Normally the exploration phase is significantly longer.

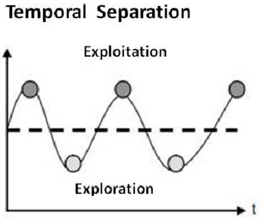

Fig. 2. Temporal Separation Phases in Software Development [16]

Companies wait extremely long, almost too long, before changing a system. Over the years the requirements and the user experience have changed and the technological feasibilities enable a completely different approach to problem solving. This evolution makes it easier for the user to cope with the difficult tasks and complexity could even be reduced. Therefore it is necessary to break up with existing system and to start from scratch in order to change the current situation. Monolithic software architecture is often of great hindrance instead of supporting the business processes. It does not support the business requirements and as standalone software it even makes the processes more complex due to many interfaces needed. Streamlining business processes is therefore hardly possible without a major overhaul of the software architecture that forms the basis for supporting the business processes.

3 Research Domain

The area of test facility automation systems in the automotive industry is an ideal research area for this paradigm change as test facility automation systems are very complex and consist of many software-intensive products. The goal of the research group has been to develop a framework for model-driven generation of automation system configuration parameters, which radically simplifies the configuration and operation of test facility systems.

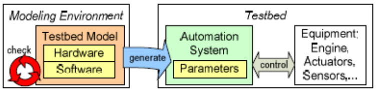

Fig. 3. Model-driven generation of automation system parameters [1]

This framework should then be the basis for business software to be implemented along the whole value chain. As a consequence, the associated business processes should be significantly streamlined.

3.1 Monolithic Architecture with Unstructured Parameters

Test facility systems need to be tailored to customer demands in a straight-forward way, which is neither well supported by the current monolithic software architecture nor does the software support the process chain.[13] The existing software has evolved over the past decades and comprises million lines of code, mainly written in C++ and C, which makes adjustments quite difficult. It requires error-prone editing of parameters (in the order of tens of thousands) in spreadsheet-like tables, as well as the adaptation of configuration files and scripts scattered in the file system. Thus the handling is time-consuming and expensive. Required changes, that would make the software systems flexible to business requirements, are neither possible with the existing software architecture nor is any potential effort for writing code for transforming the data justifiable. A software engineering has to be launched to make the system flexible to current requirements. Therefore it is highly recommended that the new tool supports the whole processes. A lot of information, which is nowadays entered into the tool in the project execution phase already have to be available in an earlier stage, e.g. in the sales phase. Therefore it is useful to rethink the product integration and use the tool as business software along the whole process chain.

3.2 Current Business Process Landscape

The current software is restricted to the project execution process without having any connection to the up- and downstream processes. Fig. 4 illustrates this situation. The handover points from one process to the other are without any tool support. Thus relevant information might be lost throughout the process chain and information needed in the progression of the project is not available as the importance of these data might not be clear in an earlier phase.

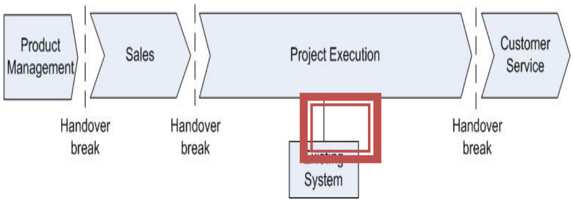

Fig. 4. Position of the current software system

Currently, each process uses different tools, like Excel sheets or proprietary tools. These gaps cause a significant information loss and extra manual conversion and transfer efforts.

4 Business Software Modeling Approach

To show the feasibility of any business software concept both sides the business processes and the software engineering have to be considered. It is crucial that a software system matches the requirements of an organization and that the software enables the business processes to adapt to business requirements. For the described research area the overall complexity of test facility systems demands for business software that applies stepwise refinement. The aim is that a unified method and tool set is established, with which everybody throughout the process chain can interact. The resulting model of the overall test facility should then correspond to the real-world test facility in all its relevant details.

The principal idea is to reduce the complexity through an object-oriented model which enables a graphical 1:1 representation of real-world items of an engine test facility. Thus, this kind of modeling is referred to as deep virtualization of the corresponding real-world test facility. In the beginning a coarse-grained model can be defined which is incrementally refined along the value chain. Models are the main artifacts describing a system under test, and a model at a certain level of abstraction can be transformed into another model at a possibly different level of abstraction: the hypothesis is that a software system that allows a stepwise modeling of a test facility automation system would support the overall business process chain and thus forms the basis for process streamlining and cost cutting.

By examining the examples of how to model an engine and its attached sensors, problems of conventional modeling languages such as UML [15] were identified. Instead the decision was made for what Atkinson and Kühne [3] have called clabject-based modeling, an approach that unifies the notion of classes and objects. Each model element has a compartment for the name, and a combined compartment for the type facet and the instance facet. The dashed arrows between the levels represent the "instance of" relationship. With a uniform representation of type facets and instance facets, our example can be modeled in a natural way. By definition, the clabjects at the top-level only have a type facet, whereas the clabjects at the bottom level only have an instance facet. [2] The advantage hereby is that the domain engineers do not have to concentrate on specific differences between instances, classes and object. They can mainly concentrate on the engineering of the domain elements.

With the clabject-based approach a library of all relevant elements is developed. The element library is then for the software usage relevant. It is filled in the beginning of the value chain (in the product management) and then used throughout the processes. The modeling of the product library, however, is essential for the whole product management and can also enable a better product management concerning a reduction of the complexity of product variants. (A separate research is initiated in the product variants area.)

Thus this approach supports the stepwise refinement, as people from different departments can work with the tool and without having to know the underlying software architecture.

5 Process Reorganisation through Business Software

A typical business process chain, shown in Fig. 5, covers aspects from product management, sales, project execution to the customer service. Each process has its special requirements and associated roles on the tool.

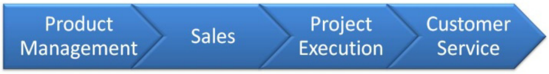

Fig. 5. Schematic Process Chain

5.1 Business Software Integration Method to Structure the Reorganization Project

Integrating business software into the tool landscape is encompassed by a lot of changes to the existing landscape. Introducing a new software modeling environment implicates potential for conflict and opposition which should not be underestimated. When the new software modeling approach should furthermore be operated as business software, it is an even more challenging task. It has to be done very carefully, also considering psycho-social effects of performing a change project within an organisation, as introducing new software especially as business software is a huge change project.

As mentioned in the introduction ERP implementation projects exist manifold [17]. However, for introducing a business software in the automation area there are almost no use-cases available and therefore, within the context of the project, a method has been developed to structure this change process. Fig. 6 shows the main phases of the BSI (Business Software Integration) method on a chronological timeline.

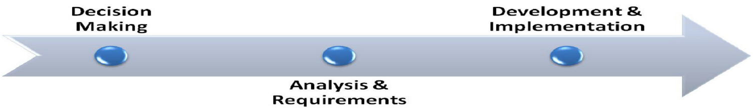

Fig. 6. Business Software Integration Method (BSI-Method)

The BSI method has been developed according to Lewin's [12] three typical phases for change. The three phases are: (1)Unfreezing, (2) Moving and (3) Refreezing.

The introduction of business software is mainly divided into three separate phases, which, nevertheless, interact with one another. Referring to Lewin [12] the different phases correlate as follows: (1) The first phase—decision making—corresponds to the unfreezing phase. Here the management decision to change the status-quo has to be made. (2) The second phase —analysis & requirements —represents to some extend also the unfreezing and in more detail the moving phase. (3) The development and implementation phase corresponds to the refreezing phase. It is split into two phases as the development phase is more complex and needs a separate focus.

Each of these single phases consists of several individual steps. Although the stepsare adapted to the organisation's situation, there are some common steps which should be included in the corresponding phase. The most important steps are highlighted in Fig. 7.

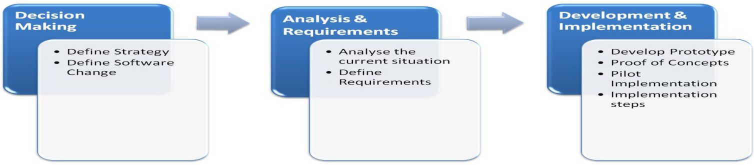

Fig. 7. Phases of the BSI-method with corresponding steps

This method can be seen as manual for any technical business software integration project, which all will mainly have these phases. The steps, however, can differ according to the maturity level of the organisation's processes and the software landscape. Each of these phases is then supported by adequate methods. Although each of these phases are not new, the combination of these phases while introducing a companywide business software has rarely been discussed in literature as no technical business software has up to now been implemented.

5.2 Benefits of the Modeling for the Process Chain

This clabject-based modeling approach can be integrated as business software along the process chain. Fig. 8 demonstrates the core benefit of such a modeling tool. Through the integration as business software, indicated as a constant banner (refered to as DeepVTool in Fig.8), it supports all essential processes and it will have predefined interfaces to other tools, which are relevant for the processes. While the new business software is the technical backbone, the other tools, indicated as tool 1 till tool 4, are mainly used for economical matters, like a sales tool or a calculation tool. All technical solution components from the business software can be automatically exported to other tools when needed. Thus, the workflow is well supported and needed information can be easily accessed at the right time.

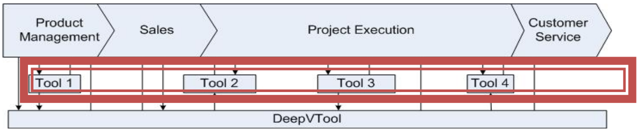

Fig. 8. Envisioned tool supported process chain

Another effect of the integration of the business tool is that several monolithic standalone tools can also be completely replaced. Thus the software landscape will be cleaned up. Thus streamlining makes the work of the people easier, as fewer tools have to be used. On the other hand the reduction of tools is also significantly displayed in the cost structure of the IT department.

Realizing this clabject-based modeling approach also has manifold benefits for the whole workflow of an organisation, which are based on some well known concepts:

- Reusability
- Correctness
- Quality assurance
- Visualization

The application of these concepts has its individual benefits on the different processes within an organisation. Therefore all relevant information should be added gradually to the test facility model in each process phase according to the particular requirements. Thus the level of detail represented by the model increases over the project progression. Fig. 9 shows the step by step refinement schematically: (1) The Product Management is responsible to have all components available for usage in the tool. (2) In the Sales phase general project information and requirements are applied in the model. Thus a first visual test facility system model is generated, which can be used as discussion base with the customer. (3) In the Project Execution phase important data and details are added, according to their relevance. For example, detailed information on the utility requirements (power supply, water supply, …) are added. (4) In the operations phase the most detailed test facility model is used to configure and operate the whole system.

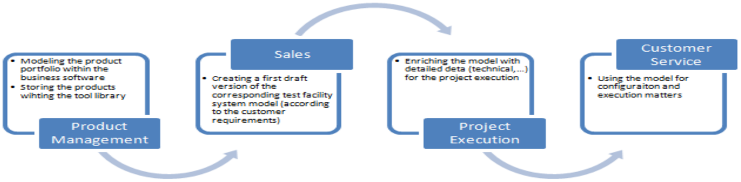
Modeling the product portfolio within the business software
Storing the products wihting the tool library
Product Management
Sales
Creating a first draft version of the corresponding test facility system model (according to the customer requirements)
Enriching the model with detailed deta (technical,…) for the project execution
Project Execution
Customer Service
Using the model for configuraiton and execution matters

Fig. 9. Step by step refinement along the process chain

A further advantage is that errors or dependencies in the design and development of test facilities can be identified already at an early stage in the sales phase, avoiding costs that are due to late detection and fixing of such errors in the subsequent project execution phase.

The changes in the workflow also require a change in the role model of the existing processes. It is an advantage of the tool that it is not intended to be a tool for experts only. Employees working in all business areas –from sales to operator engineers - should use the tool in their context. The granularity and level of detail of information however will be adapted to the role concept. Thus the business and software reengineering has also a high effect on the people within a company. It is not only that they have to change what and how they do but the basic ways how they think is altering. Therefore any reengineering project should be accompanied by a special communication strategy within the company to prepare all people affected by the change. Describing the communication strategy within our research project would go beyond of the paper's scope.

5.3 Quantitative and Qualitative Effects

In general there are two relevant points where first savings can be achieved: (1) a reduction of cycle times can either be achieved with an elimination of process steps or (2) with the shift of process steps to upstream processes. The first one can be achieved, as the system offers the possibility that process steps are done automatically. The latter one enables a cost reduction as several work steps can be done in house, which is obviously cheaper, than doing the same task at the customer's location.

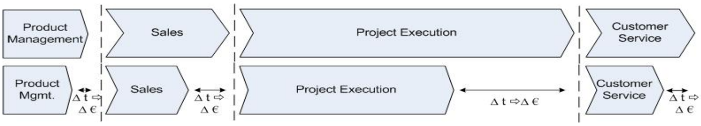

Fig. 10. Process chain streamlining

First analyses show that savings between ten and fourteen per cent can be achieved. This first analysis, however, is only based on scenarios where the business software usage is mainly concentrated on the project execution phase. With a process-wide implementation, as shown in Fig. 10, even more savings can be achieved. Costs savings in a two-digit percentage range might be.

Furthermore the integrated tool with its visualization approach can also increase the customers' satisfaction, as the tool handling becomes easier through the 1:1 representation. The satisfaction will also increase with internal customers. Training times for new employees can be radically shortened, which has on the one hand a motivational effect on the employee and on the other hand a positive financial effect for internal and external customers.

6 Related Work

Addressing the paradigm to introduce a "technical" business tool is generally not dealt with in literature. There are mainly publications dealing with ERP and CRM (Customer Relationship Management) software solutions, compare e.g. [4], [18] and [23]. The introduction of a new software tool and introducing the tool in the organisation as business software, to make use of all the advantages through the integrative approach are not at all discussed.

Technical business software systems, however, are not yet described in literature. Literature on a business software introduction and the effect on the organisation are almost not available. With our project we investigate in both areas: We develop and integrate new software system architecture and investigate on the effects of this business tool on the business processes and thus help to streamline the business processes. An area which is often discussed in literature, which is familiar to our topic, is the whole are of business or enterprise engineering. In the area of business engineering methods mainly concentrate on change projects and rarely describe the effects of this engineering on operations for longer times [8]. It, however, is not so deeply on a specified alignment of IT and business aspects. Furthermore there exists literature on enterprise engineering or parts of enterprise engineering, like [4]. This approach, however, is reduced to only one little aspect of the usage of information systems. Further approaches in the area of enterprise modeling are described and dealt with e.g. by [7]. Thus our research approach deals with an important and quite new topic. The idea of visualization realized in the software architecture already proved right in e.g. the building industry. [7,20]

7 Conclusion and Future Work

In this paper the paradigm change from monolithic standalone software to business software is described in the context of the automotive industry. Therefore a software reengineering is necessary and the therein used architecture is described in this paper. It described the effects of a business tool usage on the processes and on the company as a whole. We argue that monolithic software architectures impede business processes to adapt to changing requirements whereas business software based on a so called clabject-based modeling approach offers more flexibility and thus forms the basis of a business process reorganisation that enables major improvements.

Generally speaking it can be stated and also proven with first results, that an integrated business tool instead of many monolithic software systems can have an enormous effect on the business process within an organisation. It does not matter in which line of business business software is used. The applied research in the area of industrial automation systems proves applicable and undermines the positive effects of business tool usage. This approach promises to overcome shortcomings in legacy systems and most important it supports business processes.

References

1. Aschauer, T., Dauenhauer, G., Pree, W.: Towards a Generic Architecture for Multi-Level Modeling. In: Proceedings of the 2009 Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture WICSA/ECSA 2009, pp. 121–130 (2009)
2. Aschauer, T., Dauenhauer, G., Pree, W.: Multi-Level Modeling for Industrial Automation Systems. In: Proceedings of the 35th Euromicro Software Engineering and Advanced Applications (SEAA 2009) Conference 2009, Patras, Greece, August 27-29 (2009)
3. Atkinson, C., Kühne, T.: Model-Driven Development: A Metamodeling Foundation. IEEE Software 20(5), 36–41 (2003)
4. Bucher, T.: Ausrichtung der Informationslogistik auf operative Prozesse. Kovac Verlag, Hamburg (2009)
5. Chandler, A.D.: Strategy and structure: Chapters in the history of the industrial enterprise, 22nd print. Edition. MIT Press, Cambridge (2001)
6. Davenport, T.: Process Innovation: Reengineering Work Through Information Technology. Harvard Business School Press (1993)
7. Eastman, C., Teucholz, P., et al.: BIM Handbook. A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors. Wiley & Sons Inc., New Jersey (2008)
8. Frank, U.: Multi-Perspective Enterprise Modeling (MEMO) - Conceptual Framework and Modeling Languages. In: Proceedings of the Hawaii International Conference on System Sciences (HICSS-35), Honolulu 2002, 10 p (2002)
9. Gadatasch, A.: Grundkurs Geschaeftsprozessmanagement. Vieweg + Teubner, GWV Fachverlage GmbH, Wiesbaden, Germany, 5., erweiterte und überarbeitete Auflage edition (2008)
10. Hammer, M., Champy, J.: Reengineering the Corporation: A Manifesto for Business Revolution. Harpers Paperbacks. Rev. Upd. (2003)
11. Kawalek, J.: Rethinking Information Systems in Organizatons. Integrating Organisational Problem Solving. Routledge, New York (2008)
12. Lewin, K.: Researching Organizational Values and Beliefs. Kindle Edition (2001)
13. Martyr, A.: Engine Testing: Theory and Practice, 3rd revised edn. Butterworth Heinemann (2007)
14. Muthum, S., Withman, L.: Business Process Reengineering: A consolidated methodology. In: Proceedings of the 4th Annual International Conference on Industrial Engineering Theory, Applications and Practices, Texas (1999)

15. Object Management Group: Unified Modeling Language Infrastructure. Version 2.1.2. (2007)
16. Raisch, S.: Balanced Structures:Designing Organisations for Profitable Growth. In: Long Range Planning, pp. 483–508 (2008)
17. SAP Website (2010), `http://www.sap.de`
18. Semar, W.: Weiss + Appetito Holding AG - Software für den ganzen Konzern. In: Wölfle, R., Schubert, P. (Hg.) Dauerhafter Erfolg mit Business Software - 10 Jahre Fallstudien nach der Experience Methodik, pp. 51–64. Carl Hanser, München (2009)
19. Sundblad, S., Sundblad, P.: Business Improvement through better architected software. Microsoft Architect Journal 10 (2007)
20. The Economist. From blueprint to database (2010), `http://www.economist.com/`
21. Ungar, D., Smith, R.B.: Self: The power of simplicity. In: Proceedings of OOPSLA 1987. ACM SIGPLAN Notices, vol. 22, pp. 227–242 (1987)
22. Winter, R.: Business Engineering. In: Integrierte Informations Systeme, pp. 17–37. Springer, Berlin (2008)
23. Wölfe, R., Schubert, P. (eds.): Integrierte Geschäftsprozesse mit Business Software. Hanser Verlag, Munich (2005)