

# Aligning Software Architecture and Business Strategy with Continuous Business Engineering

Kurt Sandkuhl<sup>(✉)</sup>

Institute of Computer Science, Rostock University, Albert-Einstein-Str. 22,  
18057 Rostock, Germany  
Kurt.Sandkuhl@uni-rostock.de

**Abstract.** Continuous Engineering (CE) investigates fundamentals and basic principles of evolution in IT-engineering processes. Continuous Software Engineering (CSE) applies these principles in the context of software engineering and develops methodologies, concepts and techniques for evolvable software systems. Continuous business engineering (CBE) is closely related to principles of CE and the research field of CSE. The main purpose of this position paper is to extend earlier work on CBE by integrating the concept of capabilities. The paper describes the scope of CBE research, investigates connections to other research fields and discusses approaches in the field of CBE for aligning software architecture, capabilities and business strategy.

**Keywords:** Continuous engineering · Continuous capability engineering · Business strategy

## 1 Introduction

Most enterprises and public authorities nowadays are highly dependant on their IT-infrastructure and IT-applications to preserve their competitiveness in a global market. In particular in industry domains and service sectors, where the whole value chain has to be supported by IT, it is crucial to have solutions which are flexible in case of changes in business model or market environment. Examples are the utility sector and the banking industry. In these application fields it is of economic and strategic importance to be able to quickly adapt software systems to changes in customers' requirements, business goals or company processes. Evolution of software systems in alignment to business strategy has become a core issue. This issue is even more emphasized, as most domain-specific infrastructures are long-living. They incorporate process and product knowledge of the individual owners and often were optimized over several decades. In financial industries, for example, there are still software components in use developed in the 1980s. These components cannot simply be redesigned and replaced by new components in state-of-the-art technology, as this would require big investments, bear high risks and questionable business benefits.

Furthermore, the topic of adaptability is not a new one but has been under discussion since many years. According to a study by the META-Group in 2000, senior business executives from US-based fortune 500 companies were not satisfied with the

contribution of the information technology (IT) in their enterprises: only 12% of all IT-projects contributed from their point of view significantly to reach strategic goals [5]. A survey of McKinsey from 2014 shows that priorities of enterprises shifted towards service innovation and digitization, but the pressure on IT regarding adaptiveness and contribution to business value still exists [11]. This confirms the need for aligning IT-infrastructure with the business strategy to increase the business value of information technology. This also underlines the high attention that most medium and large businesses pay to their internal IT. But what is the contribution of information technology to the success of the enterprise? What is its business value? Does the infrastructure support changes in the business strategy in an adequate way?

During the last twenty years, a number of research projects and activities have investigated the issue of “business value of information technology”, including the fields of IT-controlling [9], evaluation of software architectures [6], or strategic management [7]. The field of Continuous Business Engineering (CBE) contributes to this research area by addressing the joint evolution of business capabilities and IT-infrastructure in an enterprise or organization. A capability is the ability and capacity that enable an enterprise to achieve a business goal in a given context [1]. Business goals are means for designing and expressing the business strategy of an enterprise.

CBE aims at integrating formulation and implementation processes on business strategy level and engineering processes on IT-infrastructure level. We consider this task as continuously ongoing process dedicated to ensure integrated evolution of business model and IT-infrastructure. CBE has to integrate various research aspects, like continuous transformation of business goals into capabilities, continuous transformation of capabilities into IT-infrastructures, continuous development of software and systems architectures, etc. CBE therefore investigates methods, concepts and technologies for linking together business oriented models and technical models.

Continuous business engineering is closely related to the principles of Continuous Engineering (CE) and the research field of Continuous Software Engineering (CSE). CE investigates fundamentals and basic principles of evolution in IT-engineering processes. CSE applies these principles in the context of software engineering and develops methodologies, concepts and techniques for evolvable software systems (see Sect. 2). In a simplified picture, CBE can be described as defining correspondences between a model of the business strategy and a model of the IT-infrastructure. Section 4 of this paper describes three approaches for linking these models based on different degrees of coupling: (a) describing the business strategy based on a balanced scorecard and linking the software architecture via indicators, (b) combining the balanced scorecard approach with enterprise ontologies, and (c) integration of business strategy and software architecture in a joint enterprise model.

The main purpose of this position paper is to extend earlier work in the field [18] by describing the scope of CBE research, identifying connections to other research fields and investigating approaches in the field of CBE for aligning software architecture and business strategy. These aspects are reflected in the structure of the following sections: Sect. 2 summarizes current activities in CE and CSE. Section 3 introduces the notion and scope of CBE, and related research subjects. Section 4 introduces approaches for aligning business strategy and software architecture.

## 2 Continuous Software Engineering

Continuous Engineering investigates fundamentals and core principles of evolution in IT-engineering processes. Continuous Software Engineering (CSE) [14] applies these principles in the field of software engineering by researching, developing and applying methodologies, concepts and techniques for evolvable software systems.

One of the core goals of CSE are long-living, evolvable software systems of high quality that can be forward developed continuously [8]. An essential part of CSE is to achieve consistency and transparency between (a) all artifacts of a software development process within a development cycle (e.g. requirements, specification, architecture design, and implementation) and (b) the various forward development cycles of a software system and their modifications. This requires the identification of variations, invariants and dependencies in order to predict the potential impact of initial and induced modifications. CSE is based on a series of integrated methods and concepts. The most important among them are:

*Model-driven Development:* CSE is based on a defined and highly mature engineering process defining all development activities with tasks and expected results. Results of all activities are represented as formalized models and transformation from activity to activity is defined.

*Components as the Basis of Software Systems:* Components encapsulate clearly defined functionality made available via interfaces. Components can be newly designed or refactored from a legacy system. CSE provides approaches how components together with their interfaces, invariants and contexts should be identified, modeled and described.

*Reference Architectures for Application Areas:* Software reference architecture defines the general structure of the applications of an application area. It also determines which components should be available, as well as required aids for the software developer, such as architectural templates for designing a system.

*Support for the Software Development Process:* CSE aims to integrate specification, design and documentation methods, as well as the use of reference architectures or architectural templates into the software construction process. In order to do so, special guidelines and aids for different processes are developed in the framework of continuous software engineering.

*Evolution Strategies:* A variety of reasons can be the cause for forward development or changes in software systems. These reasons include, for example, changes in business models, new requirements from regulators or modifications to the service or technical infrastructure. Reasons for and situations of change can be categorized and derived from evolution scenarios. CSE defines process models for these scenarios and procedures for designing software systems.

*Management and Organizational Techniques:* In addition to the continuous evolution of software systems and communication infrastructures, even the development and evolution processes require monitoring, control and continuous redesign. Thus, their management and organizational techniques are observed in the context of CSE.

### 3 Continuous Business Engineering

#### 3.1 Notion and Scope

Continuous business engineering can be defined as engineering process integrating forward-development and management processes on business level (i.e. for business strategy and capabilities required for implementing business strategies) and engineering processes on IT-infrastructure level. CBE investigates continuous transformation of business goals into capabilities, continuous development of IT-infrastructures in support of capabilities, continuous evaluation of IT-infrastructure with respect to business needs, continuous improvement of software and infrastructure engineering process, continuous development of software and systems architectures, etc. CBE therefore develops methods, concepts and technologies for linking together business oriented models and technical models.

In a simplified picture, continuous business engineering can be described as defined correspondence between business strategy and model of IT-infrastructure:

- On business strategy level, a model exists expressing the business strategy of the organisation in question. Capability models have proven to be suitable for this purpose [12]. Capabilities are the ability and capacity that enable an enterprise to achieve a business goal in a given context. Development and evolution of this model is performed and controlled by a management process.
- On IT-infrastructure level, a model exists representing the existing IT-infrastructure and/or the software architecture of the organization in question. Forward-development and evolution of this model is performed and controlled by an engineering process.

Between business strategy and model of IT-infrastructure correspondences exist. These correspondences enable control, assessment and supervision of the IT-infrastructure with respect to the business needs expressed in a business strategy. The link between business model and model of IT-infrastructure will cause a number of benefits for the enterprise and open various possibilities of triggering activities in the IT-infrastructure when changes in the business model occur. We expect advantages in

- Identifying work processes, organisational structures and software components faster that will be affected by changes in business strategy. This will lead to shorter innovation cycles in the IT-infrastructure,
- Identifying the potential for innovation in the business strategy and business model easily implementable due to existing capabilities,
- Assessing and evaluating the business value of the IT-infrastructure. If correspondences between business model and IT-model are defined, this implicitly will include criteria for evaluation of the infrastructure. Continuous assessment of IT-infrastructure will be possible.

### 3.2 Related Research Subjects

Besides the connection to CE, CBE is related to a number of research subjects contributing concepts, methods or technologies for linking business and engineering.

*Knowledge Modelling:* We see methods for describing semantics of IT-components and services as well as for describing intentions of business strategies and objectives as one of the key technologies for CBE. This includes approaches in the field of ontologies and topic maps as well as related standards and services. Relevant issues in this area are knowledge representation techniques, semantic match-making, competence modeling, evolution scenarios for information models with technologies like topic maps, semantic nets or semantic web technologies.

*Capability Management:* Recent progress in the area of capability management showed the importance of explicating the deployment context of business services, which implement capabilities, and to use the context for adjustments during business service delivery [2]. This way of abstracting from deployment contexts and adding flexibility and adaptability to operations is an important contribution to achieve CBE.

*System-Integration and dis-Integration:* one specific research subject connected to CBE is the area of system-integration and dis-integration. System integration issues arise whenever inter-enterprise solutions and software systems have to be implemented, e.g. in networks of suppliers, project-based joint venture between companies or electronic business scenarios. In the field of system-integration, CBE aims at providing mechanisms for dynamic integration between a set of IT-infrastructures. As interoperability on communication and service level is widely available due to standards, the focus should be on business model level. Dynamic integration on this level includes the detection of suitable capabilities and their implementation according to the business needs, selection of the best implementation and integration of the selected service into the infrastructure. For this task a service description only including specifications of the interface (syntax, semantics) and communication protocols is not sufficient (WSDL, .NET). We are aiming at using additional conceptual descriptions and match-making based on these descriptions.

Relevant issues in this area are component model including semantic component description for IT-infrastructures, definition of invariants and interdependencies (constraints), reference architectures and architecture patterns for evolvable IT infrastructure, evolution scenarios based on component model with technologies like systems management platforms, light-weight integration protocols.

*IT-Assessment:* The field of IT-assessment aims at evaluating the quality of IT-infrastructures with respect to the business strategy. Changes in business strategy cause evolution requirements for technical infrastructure. CBE aims at predicting where changes are necessary, define parameters for construction for longevity.

Relevant issues in this area are definition of target systems for IT strategy; mapping from business models to capabilities to IT infrastructures; evaluation of process, product, and organizational quality; competence modelling as basis for assessments; evolution scenarios for IT strategies and business models with technologies like balanced scorecard approaches, technical due diligence methods and benchmarking approaches.

## 4 Business Strategy, Capability and Software Alignment

Starting from the concepts of CSE, this section proposes three approaches for linking software architecture and business strategy: (a) describing the business strategy based on a balanced scorecard and linking capabilities and software architecture via indicators, (b) combining the balanced scorecard approach with enterprise ontologies, and (c) integration of business strategy, capability and software & service architecture in a joint enterprise model. These approaches are an extension of earlier work in this field [18]. As they approach incorporate different degrees of coupling, we will start with discussing this aspect.

### 4.1 Degrees of Coupling

Today's enterprises have to be capable to handle changes in various dimensions, including a number of external factors like changes in their markets (e.g. new competitor), in framework conditions (e.g. modified laws enforcing product features), in customers' demands (e.g. new functionality), or in their delivery processes (e.g. new technology increasing productivity). Similar to the system-oriented technical meaning of evolution, we can consider the enterprises process' of "purposeful stepwise advancement due to changes of the environment" adapting to changing environmental conditions in order to survive" as evolution.

Joint evolution of business strategy and IT-infrastructure has to be driven by the business perspective and requires a coupling between business strategy and software architecture. This opens possibilities for various integration levels between both. One extreme would be to automatically cause the necessary changes in the IT-infrastructure whenever changes in the business strategy occur. The other extreme of course is to do no integration at all and use the correspondences only for evaluation and assessment purposes. Between these two poles, various levels of integration can be achieved depending on formalization level of both models.

In our investigations, we distinguish different degrees of coupling (loose vs. tight) characterised by possible actions to manage and control evolution:

- Supervision: monitoring performance or assessing compatibility to strategic objectives
- Initiate change process: definition of requirements and change requests and initiation of change processes, accordingly
- Configuration changes: change parameters or declarations for generic components or change configuration of overall system (no changes in the implementation)
- Architecture and design changes: introduce new software components or substitute existing components, e.g. based on architecture patterns or a component library
- System changes: modify design and implementation of the software system automatically.

## 4.2 Balanced Scorecard Approach

Contemporary management literature has discussed extensively the field of formation, description, and implementation of business strategies [13]. Although different approaches for modelling a business strategy exist, which are tailored to different sizes and natures of enterprises [16], the following elements are commonly seen as essential parts of a business strategy:

- Goals and targets to reach including time frames
- Actions dedicated to achieve goals and implement the targets
- Organizational units of the enterprise responsible for implementing the actions
- Indicators, policies, assessments and other means to evaluate the progress of implementing the business strategy
- Processes implementing measurement of indicator
- Management systems with roles and processes coordinating actions and forward development of the business strategy

It has been observed by a number of researchers that strategy formulation is a non-trivial task involving a number of strategic business units. Furthermore, strategy formulation and strategy implementation are difficult to integrate.

For our investigations, we will use the Balanced Scorecard approach from Kaplan and Norton [10] to capture and model business strategy. Although this approach does not provide an own modelling language, it has from our perspective a number of advantages, e.g. being adaptable to specific organizations and having achieved a broad usage within industry and public authorities, much research work and experience published, IT tools available to support modelling a scorecard and making it operational, and own experience in using this approach for modelling IT-Strategy [17].

The Balanced Scorecard approach was developed in order to complement the traditional financial accounting model, which is very much focused on past performance, with measures for drivers of organizational future performance. Thus, the Balanced Scorecard is seen as a means to establish the *balance* between financial and non-financial aspects when developing and implementing vision and strategy of the company. A typical balanced scorecard captures the business strategy of an enterprise in four perspectives:

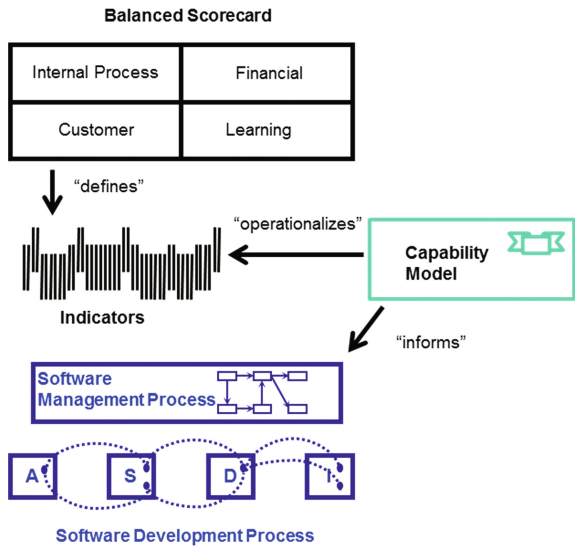
- Financial perspective: which goals have to be reached to succeed financially?
- Customer perspective: what should be the image of the enterprise from a customer's perspective?
- Internal business perspective: what business processes must be excelled at?
- Learning and growth: how to sustain the ability to change and improve?

For each of these four perspectives, strategic objectives have to be defined clarifying the vision, and measurements have to be developed and linked to the objectives. The measurements typically are implemented based on performance indicators and processes for obtaining these indicators. In the context of complex organisations, Kaplan & Norton recommend to develop not only one corporate scorecard, but also a separate scorecard for each strategic business unit conducting activities in an entire value chain.

The balanced scorecard can be used as strategic management system, i.e. to develop the strategy over the long run. This requires implementation of a management cycle including setting targets and planning strategic initiatives how to reach them, establishing strategic feedback and learning, and reviewing business strategy and modifying the vision and objectives accordingly.

Our approach for linking the business strategy – explicated in a balanced scorecard – and the software architecture is to operationalize the actions required to achieve the business strategy in capabilities, which are made explicit in a capability model. Furthermore, the approach includes to develop a separate scorecard for the business unit responsible for software and service management and to integrate this software scorecard closely with the corporate scorecard. Close integration means that the software scorecard has to have the same perspectives as the corporate scorecard and that each objective of the software scorecard has to contribute to reach at least one objective of the corporate scorecard.

Measurements and performance indicators of the software scorecard can then be used for assessing and evaluating the software architectures contribution to implement the business strategy. This balanced scorecard approach has been applied successfully in a number of enterprises [17]. Capability models form an important contribution to this approach as they also explicate deployment contexts and dependencies between IT-based business service and the (software) services implementing them. Figure 1 visualizes the overall approach:



**Fig. 1.** Balanced Scorecard-based Approach to Alignment

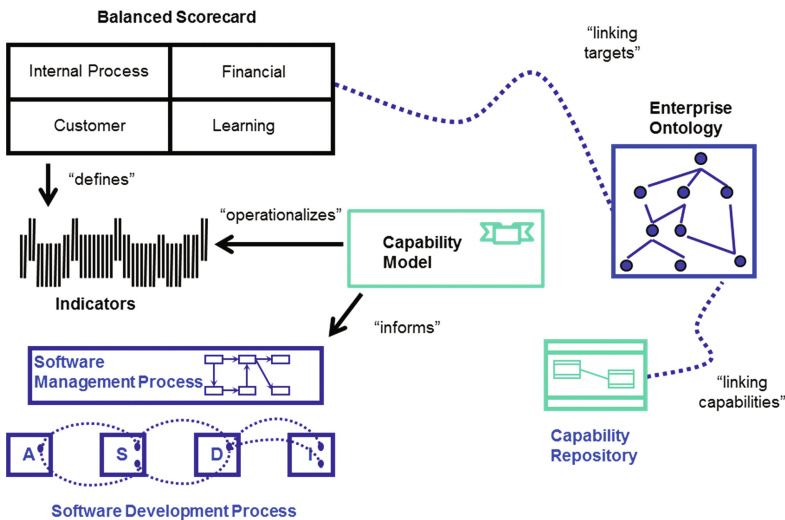


### 4.3 Combing Enterprise Ontology and Balanced Scorecard

Linking business strategy and software architecture via a balanced scorecard provides only basic possibilities for aligned evolution. The indicators can be used to discover a tendency whether the IT-infrastructure supports the business strategy adequately; the capability models add how well different deployment contexts are supported. This is primarily a view on the past performance with only limited possibility to create a future projection. Aligning strategy and IT from our point of view also should include to predict the impact of a change in business strategy, e.g. to identify the software components involved and cause-effect-relationships between strategic targets and software architecture parts.

Our approach is to use concept paths in enterprise ontologies as meta-data for linking strategy, capability and software architecture (see Fig. 2). Enterprise ontologies [20] capture the concepts and terms of an enterprise or a strategic business unit and their relationships. In our approach, we cover three different perspectives of an enterprise in the ontology:

- work processes and tasks within the enterprise
- organizational structure in the enterprise including established roles
- product or service structure of an enterprise related to the business area



**Fig. 2.** BSC and EO Approach to Alignment

These perspectives can be represented by using semantic nets [15] or semantic technologies. The enterprise ontology provides possibility to define a mapping between capability, software component and business strategy via concept paths: For each capability required for implementing a business strategy and every software component being part of the architecture, the work processes, roles, and organization structures supported by this capability and component are identified by defining concept paths in

the ontology. A concept path is a sequence of concepts being connected by associations in the ontology [3].

At the same time, roles, processes and services/products with high importance for strategic objectives are identified and linked to these objectives via concept paths. Thus, an intentional model and more formalized representation corresponding to the business strategy is created. Linking of business strategy and software architecture, e.g. in order to identify which components will be affected in case of strategy changes, requires a mapping between concept paths representing the strategic objectives and concepts paths representing the software components. This matching based on concept paths already has been applied in a number of projects in the context of Web-Portals [4].

#### 4.4 Enterprise Knowledge Modeling

The combination of Balanced Scorecard and Enterprise Ontology extends the possibilities for alignment from indicator-based performance evaluation to further formalization of business strategy and contributes to prediction of impact of changes. But it does not allow to automatically initiate and perform changes in components or a systems configuration of system. This goal requires from our point of view a tighter coupling with sound formalization, e.g. by integration of both, business strategy, capability and software architecture, within a joint model. This model, represented in a suitable modeling language, either must provide possibility to express elements and relationships for both areas or it has to integrate existing modeling approaches, e.g. Balanced Scorecard and CSE, in a single meta-model.

Numerous enterprise modeling approaches have been developed that integrate business and IT-viewpoints in a single model. Examples are Zachman's framework, TOGAF or the GERAM activities (see [19] for an overview to approaches). The Zachman framework for enterprise architecture structures representation of enterprise and system knowledge in two dimensions. The first dimension includes various viewpoints: data, function, network, people, time and motivation. The second dimension consists of abstraction levels: scope, business model, system model, technology model, detailed representations and functioning enterprise. From a theoretical point of view, i.e. looking at the completeness of the modeling approach, these concepts have the potential to implement the required tight coupling and support joint evolution. Furthermore, work on capability management and capability design and delivery (CDD) contributed an approach which supports co-evolution of business strategy and IT-based business services [2].

Based on our experience from earlier alignment projects, we see at least two aspects to be investigated in future research: Business strategy includes a lot more than processes, objectives and indicators. A holistic approach would have to take into account business culture, innovation processes, or organizational competences, which are to a large extent creative and hard to capture in a technical model. These aspects clearly would have to be simplified or excluded, when developing an enterprise model based on the above-mentioned approaches, the effect being a loss of context information and semantics, and of cause-effect-relationships. Medium-sized and even large enterprises have problems to provide a fairly complete software architecture model or a well-defined

business strategy. Development of an enterprise model integrating both aspects will be in many cases not feasible due to missing information.

## 4.5 Conclusion

Based on the component-based and model-driven concepts from CSE for modeling the software architecture, we have presented three approaches to define correspondences providing different levels of support for joint evolution. Using our classification from Sect. 4.1, we can put these approaches into relationship with respect to their degree of coupling:

The balanced scorecard approach only facilitates a loose coupling between business strategy and software architecture based on indicators. This coupling supports supervision of the software architecture but no stronger integration. Balanced scorecard in combination with enterprise ontology enables us to identify software components that are affected by changes. Based on this, initiation of changes is possible, being a higher degree of coupling. Enterprise modelling supports the highest degree of coupling by promising – at least from a model point of view – to initiate and perform system changes in case of changes in the business strategy.

## 5 Summary

The business models of many enterprises nowadays are highly dependent on their IT-infrastructure. In order to preserve the competitiveness in a global market, it is crucial for them from a business perspective to quickly implement new business services, to react to new market demands or to implement new regulations. In this context, it is of decisive economic and strategic importance from a technical perspective to be able to quickly adapt software systems to changes. Alignment of IT-infrastructure with business strategy has become a core issue. The approach of Continuous Business Engineering (CBE) contributes to this challenge by addressing the joint evolution of business strategy, capabilities and software architecture. CBE aims at integrating forward-development on business strategy level and engineering processes on capability and software architecture level. We consider this task as a continuously ongoing process dedicated to ensure integrated evolution. In a simplified picture, CBE can be described as defining correspondences between a model of the business strategy and a model of capabilities and the software architecture.

This paper describes the scope of CBE research, identifies connections to other research fields and to investigate three approaches in the field of CBE based on different degrees of coupling: (a) describing the business strategy based on a balanced scorecard and linking the architecture model via indicators, (b) combining the balanced scorecard approach with enterprise ontologies, and (c) integration of business strategy, capabilities and software architecture in a joint enterprise model. Future work in the context of these approaches will be dedicated to (a) elaborating the advantages and limitations of the three different approaches, (b) a more detailed comparison to enterprise architecture

management and capability management, and (c) applications in real-world projects in order to verify the potential.

## References

1. Ahlemann, F., Stettiner, E., Messerschmidt, M., Legner, C.: Strategic Enterprise Architecture Management: Challenges, Best Practices, and Future Developments, p. 2012. Springer, Berlin (2012)
2. Bērziša, S., et al.: Capability driven development: approach to designing digital enterprises. *Bus. Inform. Syst. Eng.* **57**(1), 15–25 (2015)
3. Billig, A., Sandkuhl, K., Wendt, A.: Basic support for evolutionary web-portals: the BaSeWeP approach. In: Proceedings of the IASTED 2000, Las Vegas, November 2000
4. Billig, A., Sandkuhl, K.: Match-Making based on semantic nets: the XML-based BaSeWeP approach. In: Proceedings XSW 2002. Springer (2002)
5. Buchanan, D., Soley, R.: Aligning Enterprise Architecture and IT-Investments with Corporate Goals. META-Group (2002). <http://www.metagroup.com/>
6. Clements, P., Kazman, R., Klein, M.: Evaluating Software Architectures. Addison-Wesley Professional, October 2001
7. Croteau, A.-M., Bergeron, F.: An information technology trilogy: business strategy, technological deployment and organizational performance. *J. Strateg. Inform. Syst.* **10**, 77–99 (2001). Elsevier
8. Grosse-Rhode, M., Kutsche, R., Bübl, F.: Concepts for the Evolution of Component-Based Software Systems. Technical report, Fraunhofer ISST, Germany, October 2000
9. Hochstrasser, B., Griffiths, C.: Controlling IT Investment: Strategy and Management. International Thomson Business Press (1991)
10. Kaplan, R., Norton, D.: Balanced Scorecard. Harvard Business School Press, Boston (1996)
11. Khan, N., Sikes, J.: IT under pressure: McKinsey Global Survey results. McKinsey, March 2014. <http://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/it-under-pressure-mckinsey-global-survey-results>
12. Koç, H., Kuhr, J.-C., Sandkuhl, K., Timm, F.: Capability-driven development - a novel approach to design enterprise capabilities. In: El-Sheikh, E., Zimmermann, A., Jain, L.C. (eds.) *Emerging Trends in the Evolution of Service-Oriented and Enterprise Architectures*. ISRL, vol. 111, pp. 151–177. Springer, Cham (2016). doi:[10.1007/978-3-319-40564-3\\_9](https://doi.org/10.1007/978-3-319-40564-3_9)
13. Littler, K., Aisthorpe, P., Hudson, R., Keasey, K.: A new approach to linking strategy formulation and strategy implementation. *Int. J. Inform. Manage.* **20**(6), 411–428. Elsevier Science, December 2000
14. Müller, H., Weber, H. (eds.): Continuous Engineering of Industrial Scale Software Systems. Dagstuhl Seminar #98092, Report No. 203, Schloss Dagstuhl, March 1998
15. Nilsson, N.: Principles of Artificial Intelligence. Springer, Heidelberg (1982)
16. Quezada, L., et al.: A methodology for formulating a business strategy in manufacturing firms. *Int. J. Prod. Econ.* **60–61**, 87–94 (1999). Elsevier
17. Sandkuhl, K., Gohlke-Micknis, S., Hacker, J., Jakoby, A.: Die IT-StrategieCard - Ein Zielsystem für strategische IT-Themen auf Basis der Balanced Scorecard. Proc. Wirtschaftsinformatik 2001, Augsburg, September 2001
18. Sandkuhl, K., Smirnov, A.: Continuous business engineering: towards aligned evolution of business strategy and software architecture. In: Proceedings of the 10th ISPE International Conference on Concurrent Engineering, Madeira, Portugal (2003). ISBN 90 5809 622 X

19. Sandkuhl, K., Stirna, J., Persson, A., Wißotzki, M.: Enterprise Modeling: Tackling Business Challenges with the 4EM Method. The Enterprise Engineering Series. Springer, Heidelberg (2014). ISBN 978-3662437247
20. Uschold, M., et al.: The Enterprise Ontology. Artificial Intelligence Application Institute, Edinburgh (1997)

Advanced Information Systems Engineering Workshops  
CAISE 2017 International Workshops, Essen, Germany,  
June 12-16, 2017, Proceedings  
Metzger, A.; Persson, A. (Eds.)  
2017, XXVI, 123 p. 31 illus., Softcover  
ISBN: 978-3-319-60047-5