5-15-2012

REFRAMING THE GOVERNANCE DEBATE: A MULTILEVEL PERFORMANCE MEASUREMENT APPROACH BASED ON CAPABILITIES

Daniel Fuerstenau
Freie Universität Berlin

REFRAMING THE GOVERNANCE DEBATE: A MULTILEVEL PERFORMANCE MEASUREMENT APPROACH BASED ON CAPABILITIES

Fürstenau, Daniel, Freie Universität Berlin, DFG Pfadkolleg Research Center, Garystr. 21, 14195 Berlin, Germany, daniel.fuerstenau@fu-berlin.de

Abstract

Conventional IT planning approaches, which are based on applications, begun to be disputed with the rising interest in service-oriented architectures. But, given the status-quo of the IT landscapes in most companies IT management faces a dilemma. Should IT governance for transformation projects remain at the application-level or should services become the heart of the transformation? In this paper we reframe the ongoing debate on IT governance by introducing a multilevel approach centered on capabilities. In particular, we present an information model, aggregation mechanisms to support performance management based on the model and a planning procedure that guides IT managers and enterprise architects on how to apply our method. Finally, we introduce an example and a prototype to contextualize our method. First findings suggest its usefulness in supporting large enterprise transformation projects.

Keywords: Enterprise Transformation Planning, Capability-based Planning, Performance Measurement, Enterprise Architecture, IT Governance

1 Introduction

Over the last years, IS researchers and practitioners attention shifted from an application-centric toward a service-oriented perspective. Nevertheless, most companies IT portfolios consist of large amounts of legacy systems which cannot be transformed into services in short time. In particular, change resistance is a well-known barrier for new systems introduction (Markus 1983). To transform today's idiosyncratic, often 'silo-like' business applications into more flexible structures, we need fresh governance approaches to integrate both perspectives in a consistent manner. These approaches should account for the premature state of modularization in most IT landscapes. Services cover only 10% of the IT landscape compared to the potential coverage in most companies (Becker et al. 2011). Often times IT landscapes are not service-enabled.

In sum, IT management faces a dilemma. While aspirations shift to the service-level, in practice the given legacy structures remain an essential part of the IT landscape. Thus, the locus of control between applications and services should be revisited. In this paper, we reframe the debate by presenting a multilevel approach based on capabilities to integrate both perspectives.

The target audience of this contribution is IT management (CIO-level), enterprise architects (EA's), business executives and researchers concerned with enterprise transformation. Our approach could be applied in a large-scale transformation project, such as a post-merger integration, but a broader application as part of the regular planning effort of IT management is envisaged.

This paper is structured as follows: We present the state-of-the-art in capability-based planning (section 2), and then we outline our approach (section 3). We give an application example and explain prototypical tool support (section 4). This paper concludes with implications and an outlook of future research areas (section 5).

2 State-of-the-Art

While most companies center their IT governance processes on applications, a gap between business and IT was long recognized. From a technical standpoint scholars and practitioner argue that service-oriented architectures (SOA) bridge this gap by decomposing the IT architecture into loosely-coupled services with standardized interfaces (Erl 2005). Also, from a business perspective SOA is recognized as an enabler to facilitate alignment. As SOA is moving to maturity, SOA governance receives more attention. Furthermore, governance process integration for service and application portfolios becomes important. In particular, because the lifecycles of mission-critical business applications often span decades and decreasing returns for additional SOA projects may appear, a major shift in IT landscapes of most companies is improbable. Consequently, we expect hybrid IT landscapes. These landscapes demand multilevel, business-led planning approaches. Even if capability-based planning has been considered before, in application-centered organizations, inert structures may hamper its success. In addition, semantic misconceptions may throw up barriers for capability-based planning.

TOGAF, a practitioners EA framework, fosters capability-based planning in its latest version. Here, capability-based planning is a technique to derive architecture requirements in a business-led manner (TOG 2009). TOGAF presents capability-based planning as part of the cyclic Architecture Development Method (ADM). In the business architecture phase of the cycle one should determine the capabilities delivered by work packages. Furthermore, in TOGAF metamodel, a work package can be associated with all other modeling elements, e.g. application or services. But *how* the association should be used, has so far received too little attention.

The IS literature, as a different stream of research, has also long recognized capabilities. In particular, the resource-based view uses capabilities as a central construct. Different IS researchers conceptualize types of capabilities. Bharadwaj (2000) classifies IT infrastructure capabilities, IT human resource capabilities and intangible IT-enabled capabilities. Bhatt and Grover (2005) differentiate value, competitive and dynamic IT capabilities. Piccoli and Ives (2005) identify technical skills, IT management skills and relationship assets as IT capabilities. For Nevo and Wade (2010) emergent IT-enabled capabilities result from successful integration of IT assets and organizational resources.

Overall, no integrated model was found. Authors use the term 'capability' context-specific. Currently the literature is lacking a consistent integration, especially among construction-oriented and behavioural approaches. We see two meanings in the literature. First, *organizational* capabilities describe 'what is done' in a company on a high level. Second, *IT capabilities* indicate IT systems functions or IT management resources. In the following, we begin to integrate these different views.

3 Method

3.1 Research Approach

The goal of this contribution is to construct a modelling method to support IT management in the task of enterprise transformation. To achieve this goal we adopt a design-science perspective (Hevner et al. 2004). Our research is based on a relevant business problem we gathered in a field setting (see section 3.2). We develop an information model (section 3.2), aggregation mechanisms (section 3.3), a procedural model (section 3.4) and a prototype as *design artifacts*. *Design evaluation* in this paper is based on an example derived from a practical application of the approach in a field setting (see section 4). Nevertheless, further research is needed to detail the design evaluation.

3.2 Theoretical-conceptual Background and Information Model

We start with a model-based perspective common in EA planning (Aier et al. 2008). While EA planning in general developed a number of approaches facilitating alignment (e.g. Johnson and Ekstedt 2007; Aier and Winter 2009), further work is necessary to integrate different views on capabilities and to support real-world enterprise transformations. In particular, we address the requirements of a field setting (1.) to make available modelling concepts supporting transformation in a hybrid IT landscape with applications and services (2.) to provide an assessment approach based on performance indicators (3.) to allow for aggregation of indicators on multiple levels and (4.) to track progress over time. We only discuss the first three requirements in this contribution.

Figure 1 proposes modelling concepts and relations among them. Concepts are business applications, services, IT-enabled capabilities and capability groups. Here, a *business application* is a functional aggregation of IT components. Next, a *service* is also seen as a functional aggregation of IT components, but is often more fine-grained. An example is 'Business Partner Search' with different service interfaces allow searching and retrieving business partner data. Services are often deployed on an enterprise service platform serving as a container for a number of services to support a business process. Both, applications and services are IT artifacts. Moreover, IT artifacts provide *IT-enabled capabilities*. IT-enabled capabilities are abstractions of IT functions described in a business domain language. As such, capabilities are part of the integration layer between business and IT architecture. In a broader context, *organizational* capabilities include skills, social, cultural *and* IT aspects. Here, we only focus on IT support. Hence, an IT-enabled capability represents the IT-supported dimension of an organizational capability. In addition to their interrelatedness, they are defined by their properties. We describe capabilities by a unique name, description and indicators (among other attributes not presented here). Additionally, we cluster capabilities into *capability groups* used as common building blocks. Hierarchical abstraction into capability groups allows handling similar capabilities together.

Overall, our information model is complementary to other EA approaches. Still, the proposed information model differs from i.e. TOGAF 9 to important extends. Most important, the association between IT artifacts and capabilities has unambiguous semantics. Apparently, effort is needed to apply our approach. Existing sources of application and service data should be employed. Nevertheless, further effort for data gathering and modeling might be necessary, especially to achieve the business domain language of abstraction for capabilities. Still, this modeling eases transformation planning.

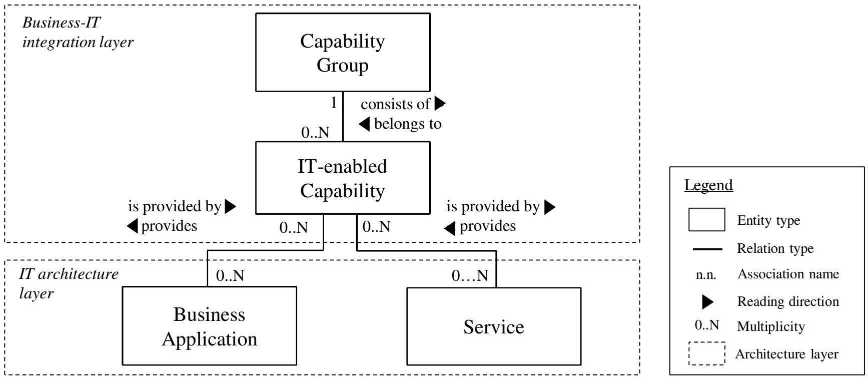

Figure 1: Information model for capability-based planning

3.3 Aggregation Mechanisms

From the information model we now derive aggregation mechanisms from IT artifacts to capabilities. Our approach is based on indicators to assess the situation. While we appreciate metric measures, often times data gathering is cumbersome. Therefore, measurement relies on itemized rating scales - that is Likert scales (e.g. 1 – poor, 5 – good performance, 0 – not applicable).

In our model, the IT architecture consists of $\boldsymbol{N}$ business applications and services. Thus, the vector of all IT artifacts is designated as $\boldsymbol{a} = (a_1, a_2... a_N)$. Each $\boldsymbol{a_i}$ is a compound of different performance indicators. Let $\boldsymbol{M}$ be the number of indicators, e.g. scalability, modularity, standardization and flexibility. A performance profile for a sample application a_5 might be *2322* indicating medium performance over all dimensions. The performance profiles of all applications and services can then be displayed in a $\{NxM\}$ matrix. Next, we introduce $\boldsymbol{P}$ capabilities in our model designated by $\boldsymbol{c} = (c_1,...,c_P)$. We map a given IT artifact $\boldsymbol{a_i}$ to a given capability $\boldsymbol{c_i}$. This mapping is established when two objects are connected by the modeler.

As indicated in the information model (see figure 1) one capability can be supported by *0...N* business applications and services. Thus, a decision-rule is necessary, because capabilities can be supported by more than one IT artifact. We assign a variable K to each capability indicating the number of providing IT artifacts. Now, we use two kinds of *aggregation functions*. The first aggregation is an *average function* designated by $f_{AVG} = 1/K\ (a_1 + a_2 + ... + a_N)$. The second alternative is a *minimum function.* Here, $f_{MIN} = min\ (a_1,...,a_N)$. The modeler decides on the preferred option. Furthermore, we perform a mapping from the capabilities to the capability groups. Here, the same logic applies as for the mapping from IT artifacts to capabilities. In addition, a colour-coding is used. A coding function maps a colour code to each performance value of an IT artifact $\boldsymbol{a_i}$ (1 is displayed red, 5 is green). A fine-grained mapping is applied to account for floating numbers in the aggregated values of the capabilities and capability groups.

$$\boldsymbol{c_i} = \begin{cases} a_i, \text{ when } K=1 \\ \textit{aggregation function} \text{ for } a_1 .. a_N, \text{ when } K > 1 \\ 0, \text{ otherwise} \end{cases}$$

In sum, our mechanisms allow us to aggregate information over multiple levels gathered for IT artifacts to the level of capabilities and capability groups. Also, different aggregation functions are available. So, one can evaluate capabilities and their underlying IT support in an integrated model.

Here, capabilities become the main locus for governance. Still, measurement relates to applications and services - *not* capabilities. From a governance point of view, decentralized IT unit members (e.g. application and service owners) can deliver indicator values for their owned artifacts. Delegation is important, because often times established governance processes relate to applications. Furthermore, a single person might not have the knowledge to assess all capabilities in a domain.

3.4 Capability-based Planning Procedure

In this section we propose a *planning procedure* as additional design artifact. The procedure includes the following steps (see figure 2): (1) scope and goal definition (2) capability mapping (3) analysis and assessment (4) measure implementation and (5) continuous development. The procedure is carried out by a *project team* accompanied by a *domain architect*, *business unit* and *IT unit members*. A domain architect is a person responsible for IT planning in one or more domains. Here, a domain is a stable building block of the companies' business model used for IT governance (e.g. shared services).

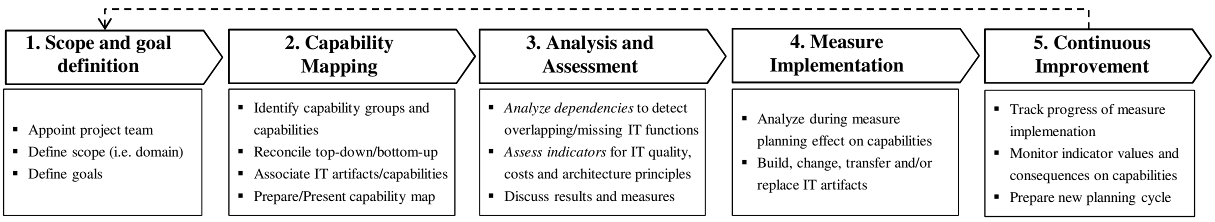

Figure 2. Procedural Model of proposed method

Define Scope and Goals: First, the project group is appointed and the scope is defined. Given a domain structure, one domain is selected. Otherwise, existing business or IT areas could be used.

Capability Mapping: Starting with a domain, the project team identifies capability groups and capabilities (see section 4 for examples). We propose a mixed, top-down and bottom-up, procedure for capability mapping. *Bottom-up*, the project team performs a functional decomposition of current core IT applications and services. While IT unit members (e.g. application owner) deliver their knowledge as input, the project team ensures a domain-wide perspective. *Top-down*, the project team identifies important functional areas. In general, candidates for capability groups include major business objects, business products or business processes among other selection criteria. As a means for identification, we propose meetings led by the project leader accompanied by the domain architect. Meetings (face-to-face or web conference) must include business unit participants to ensure proper availability of business knowledge. After the sessions, the project team reconciles bottom-up and top-down view. Then, the team proposes a first draft of the capability map. Here, IT unsupported areas appear as white spots on the map. The team presents the capability map to IT unit and business unit members. An iterative feedback process supported by visual means, e.g. posters, fosters stakeholder communication.

Analysis and Assessment: Based on the capability map the project team carries out analysis and assessment. *Analysis* reveals redundancies and missing IT-enabled capabilities using dependency analysis as a major technique. Redundancy means functional overlapping of IT support in different units, countries or segments. Missing IT functions account for desired, but currently not available IT support. For the *assessment* indicators are applied (see section 3.3). The project team agrees on appropriate indicators and defines their target values. Indicators give information on the quality of IT support, costs and/or architecture principle compliance. The enriched results with indicators are visualized on the capability map for management communication and discussion of change areas.

Measure implementation: The next step is measure implementation. Derived from the assessment stem measures like building new IT artifacts, changing existing IT artifacts, transfer of IT artifacts (e.g. to another provider/outsourcing partner) or replacement. The measures affect the capabilities.

Continuous Improvement: Carrying out the steps in a continuous fashion is beneficial (dashed line in figure 2). In particular, progress regarding the indicators should be tracked in regular time intervals.

4 Example and Prototype

Example: In order to evaluate our design artifacts, we introduce an example of a European insurance provider. The company conducts a large-scale enterprise project to transform its core business systems. Further, the company faces the given business problem (see section 3.2). The company has domains for ‘channel integration’, ‘sales/marketing’, core products in ‘life insurance’, ‘accidents’ and ‘cars’, as well as a ‘shared services’ and ‘party’ domain. The project team used our method to analyse the ‘party’ domain in detail (step 1). Out of proposals from the IT and business units the project team identified IT capabilities for ‘partner origination’, ‘partner relocation’, ‘partner retrieval’ and ‘end of partner relationship’. The capabilities became clustered in a capability group called ‘partner lifecycle management’ (step 2). Next, the team conducted an analysis (step 3). Several overlapping customer management systems were identified, but no immediate consolidation potential was revealed.

Application/Service *($a_1,..,a_N$)*	**Performance indicator** (1 – poor, 5 – good, 0 – not applicable)			
	Scalability	Modularization	Standardization	Flexibility
a_1: Customer-DB (CDB)	1	1	4	1
a_2: Partner System (PAS)	3	2	3	3
a_3: Archive Search (ARS)	3	2	2	0

Table 1. Assessment results for example applications

Thus, the project team conducted further analysis based on indicators (see section 3.4). An example assessed was the ‘partner retrieval’ capability which was supported by three applications (see table 1). Aggregated results are summarized in table 2 based on average (= 2,33) and minimum (= 1,00). Based on the assessment results, the project team could compare the given applications. The team proposed a further investigation to replace CDB during the transformation, because of its overlapping functionality with the other applications and its weak support for ‘partner retrieval’. A detailed analysis then revealed multiple, proprietary interfaces as one reason for overhauling complexity. A service interface for partner retrieval was finally proposed to enhance standardization.

App./Service ($a_1,..,a_N$)		**Capability** ($c_1,..,c_P$)			**Capability Group**		
Name	Assessment ’Scalability’	Name	Average ’Scalability’	Minimum ’Scalability’	Name	Average ‘Scalability’	Minimum ‘Scalability’
a_1: CDB	1	c_1: Partner Retrieval	2,33	1,00	Partner Lifecycle Mgmt.	2,17	1,00
a_2: PAS	3						
a_3: ARS	3						
a_1: CDB	1	c_2:Partner Origination	2,00	1,00			
a_4: CAM	3						

Table 2. Aggregation of the assessment results over two levels from IT artifacts to capabilities

Prototype: The method proposed so far has been embodied in a prototypical implementation. The prototype is based on an established EA tool (BOC 2011). The visualization includes heat mapping functionality - that is colourful visualisation of objects in the tools model editor and web browser. If the modeller choses the indicator dimension under consideration, e.g. standardization, and selects the analysis level, e.g. applications, capabilities or capability groups, then the visualization is updated. Of interest is the combination of heat mapping and transformation planning, which we cannot present here. The idea is to compare the indicators at different points in time, e.g. before and after measure implementation.

5 Conclusion and Outlook

As a starting point we claimed that current IT governance approaches do no resolve governance dilemmas in hybrid IT landscapes. Capability-based planning reframes the debate in the context of enterprise transformation projects. In our contribution we proposed a multilevel performance measurement approach based on capabilities. Given the turbulent business environment, business and IT artifacts are more in flux than ever. While applications and services evolve rapidly, business-led planning which accounts for stable *and* unstable areas in the IT landscape becomes more and more a necessity. Our contribution adds a fresh view on the growing stream of literature on business-led planning. In particular, we begin to approach capability monitoring methodologically. Therefore, we view our contribution as a first step on a path toward a better integration of different theoretical and methodical research streams. While the implementation was already used in a practical context, further research is needed to elaborate the design evaluation. In addition, we currently explore temporal aspects in more detail as another challenging area of research.

Acknowledgement

The author would like to thank Dr. Lutz Kirchner and Christoph Moser from BOC Group for their support on ideas for this paper. The author was with BOC Group before March 2011. Developmental feedback from the editors and three anonymous ECIS reviewers is also gratefully acknowledged.

References

Aier, S. and Winter, R. (2009). Virtual Decoupling for IT/Business Alignment – Conceptual Foundations, Architecture Design and Implementation Example. . In: Business & Information Systems Engineering 1, p. 150-163.

Aier, S.; Riege, C. and Winter, R. (2008). Enterprise Architecture – Literature Overview and Current Practices. In: Busienss & Information Systems Engineering (4), p. 292-304.

Bharadwaj, A. (2000). A resource-based perspective on information technology capability and firm performance: An empirical investigation. In: MIS Quarterly 24, p. 169-196.

Bhatt, G. D. and Grover, Varun (2005). Types of Information Technology Capabilities and Their Role in Competitive Advantage. In: Journal of MIS 22 (2), p. 253-277.

Becker, A.; Widjaja, T. and Buxmann, P. (2011). Value Potentials and Challenges Service-Oriented Architectures. In: Business & Information Systems Engineering 3 (4), p.199-210.

BOC (2011). Homepage – Section ADO*it*. URL: http://www.boc-group.com, Access: 27/11/2011.

Erl, T. (2006). Service-oriented architecture. Concepts, technology, and design. 6th print. Upper Saddle River (New Jersey): Prentice Hall Professional Technical Reference.

Hevner, A. R.; March, S. T.; Park, J. and Ram, S. (2004). Design science in Information Systems research. In: MIS Quarterly 28 (1), p. 75-105.

Johnson, P. and Ekstedt, M. (2007). Enterprise architecture. Models and analyses for information systems decision making. Lund: Studentlitteratur.

Markus, M. L. (1983). Power, Politics, and MIS Implementation. In: CACM 26 (6), S. 430-444.

Nevo, S. and Wade, M. R. (2010). The formation and value of IT-enabled resources. Antecedents and consequences of synergistic relationships. In: MIS Quarterly 34 (1), p. 163-183.

Piccoli, G. and Ives, B. (2005). Review: It-dependent strategic initiatives and sustained competitive advantage. In: MIS Quarterly 29, p. 747-776.

TOGAF (2009). The Open Group Architecture Framework. Version 9. URL: http://www.togaf.org/, Latest Access: 27/11/2011.