

12-31-2013

Improving IS Functions Step by Step: the Use of Focus Area Maturity Models

Marlies van Steenbergen

Sogeti Netherlands, marlies.van.steenbergen@sogeti.nl

Rik Bos

Open University, The Netherlands, rik.bos@ou.nl

Sjaak Brinkkemper

Information and Computing Science, Utrecht University, The Netherlands, s.brinkkemper@uu.nl

Inge van de Weerd

VU University, Amsterdam, i.vande.weerd@vu.nl

Willem Bekkers

Centric, willem.bekkers@centric.eu

Follow this and additional works at: <http://aisel.aisnet.org/sjis>

Recommended Citation

van Steenbergen, Marlies; Bos, Rik; Brinkkemper, Sjaak; van de Weerd, Inge; and Bekkers, Willem (2013) "Improving IS Functions Step by Step: the Use of Focus Area Maturity Models," *Scandinavian Journal of Information Systems*: Vol. 25: Iss. 2, Article 2. Available at: <http://aisel.aisnet.org/sjis/vol25/iss2/2>

This material is brought to you by the Journals at AIS Electronic Library (AISeL). It has been accepted for inclusion in Scandinavian Journal of Information Systems by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Improving IS Functions Step by Step: The use of focus area maturity models

Marlies van Steenberg
Utrecht University / Sogeti Netherlands
Marlies.van.steenbergen@sogeti.nl

Rik Bos
Open University, The Netherlands
Rik.bos@ou.nl

Sjaak Brinkkemper
Utrecht University, The Netherlands
S.brinkkemper@uu.nl

Inge van de Weerd
VU University Amsterdam, The Netherlands
i.vande.weerd@vu.nl

Willem Bekkers
Centric
Willem.bekkers@centric.eu

Abstract. With new technologies developing rapidly and becoming more common, organizations need to keep up their IS capabilities. Maturity models are designed as a means to support capability development. The existing literature is strongly dominated by studies of fixed-level maturity models, i.e., maturity models that distinguish a limited set of generic maturity levels, such as the well-known CMM. We argue that, while fixed-level maturity models may be well-suited to assessing the maturity of IS capabilities, another form of maturity model, the focus area maturity model, is better suited to supporting incremental improvement. In this paper we define the concept of focus area maturity model. We use a design-science research method, basing our work on both extensive industry experience and scientific investigation.

Accepting editor: Pär Ågerfalk

Keywords: Design research methodology, design science, maturity model, maturity matrix, method engineering, process improvement, enterprise architecture, software product management.

1 Introduction

According to the resource-based view of the firm the success of firms depends on their abilities to deploy their resources to competitive advantage (Wernerfelt 1984). As technologies are becoming more common and more accessible, organizations must distinguish themselves by the way they apply these technologies. In order to do so, they have to develop their IS capabilities (Montealegre 2002; Scott 2007).

The field of IS encompasses a great diversity of capabilities, focusing on external relationships, internal alignment and deploying internal resources (Wade and Holland 2004). Developing capabilities is not always straightforward. Repenning and Sterman (2002) refer in this context to the paradox of the useful but unused innovations, such as TQM. They discuss the importance of investment in both first and second order improvement. The fact that capabilities go through a process of development is also stressed by Helfat and Peteraf (2003), who introduce the concept of the capability lifecycle. Capability development in support of a new strategy is a gradual process rather than a sudden event, that is cumulative and expansive (Montealegre 2002).

Maturity models are a means to assess the capabilities of an organization in regard to a specific discipline and to guide their development (Cleven et al. 2012; Poepelbuss et al. 2011; Mettler et al. 2010; Scott 2007; Rosemann and De Bruin 2005). Maturity models describe typical patterns in the development of organizational capabilities, distinguishing well-described stages. The vast majority of maturity models encountered in the literature distinguish four to six stages, following the example of the generic 5-level maturity model used by CMM (Hansen et al. 2004).

While the concept of maturity model has become established in the IS community (Cleven et al. 2012), maturity models seem to be primarily useful for assessing maturity rather than guiding step by step development to a higher level of maturity. Many maturity models do not provide guidance in selecting and prioritizing actual improvement measures (Poepelbuss and Röglinger 2011; Poepelbuss et al. 2011; Huang and Han 2006; Mettler et al. 2010). This lack of guidance is a problem, as executives frequently have difficulties in deciding where to start when transforming business processes (Hammer 2007). A second criticism of maturity models that recurs in the research literature is their apparent lack of theoretical foundation. Kohlegger et al. 2009, for instance, state that many authors of maturity models simply build on their predecessors without much thinking about the appropriateness of their design decisions. They conclude that the nature of maturity models has not been theorized well in literature. The lack of theoretical foundation is also noticed by other authors (McCormack et al. 2009; de Bruin et al. 2005; Poepelbuss et al. 2011; Solli-Saether and Gottschalk 2010).

The objective of our research is to provide a metamodel that researchers can use to define a step by step capability improvement path for a specific IS function until a desired state has

been reached. By IS function we mean the whole of competencies and means to realize a well-described goal, such as for instance project portfolio management or IT governance. An IS function encompasses both assets and capabilities in the sense of the resource-based view.

To meet this objective we introduce an alternative type of maturity model to the common *fixed-level* or *stage-based* maturity model, which we label *focus area* maturity model. The focus area maturity model adds to the existing maturity model concept by focusing on the development part of developing IS functions instead of just the measuring part. It allows for the fact that different parts of an IS function may have different paces and stages of development while at the same time there are dependencies between the development paths of these parts. By doing so, it provides more guidance in improving IS functions step by step. The focus area maturity model originates from testing practice where it is called the Test Process Improvement (TPI) model (Koomen and Pol 1999). The research question we aim to answer is: *Is the concept of focus area maturity model suitable for guiding the development of IS functions?* In this paper we generalize the TPI, formally define the concept of focus area maturity model and illustrate its use in the fields of enterprise architecture management and software product management. We thus contribute to the knowledge base by taking an approach that has proven its use in practice in one field, and evaluating and generalizing it for use in other IS fields. Thus we extend the concept of focus area maturity model from practice in one field to a broadly applicable generic metamodel.

The paper is structured as follows. To motivate our research and place it in context the next section discusses previous work in the field of maturity models. Section 3 presents the design-science research approach we followed to generalize and extend the concept of focus area maturity model for use in other domains. Section 4 then formally defines the concept of focus area maturity model, followed in section 5 by its demonstration in the two functional domains of enterprise architecture management and software product management. Section 6 concludes with discussions, limitations and further research.

2 Defining maturity models

The number of research papers on maturity models is extensive (for a recent overview see Poepelbuss et al. 2011). Many of these papers describe specific maturity models. Other papers discuss the concept of maturity model, trying to ground it in theory or suggesting frameworks to be used in future research on maturity models (Patas 2012; Mettler et al. 2010; Poepelbuss and Röglinger 2011).

Maturity models are conceptual models based on the idea that organizational capabilities develop through a number of anticipated, desired or logical stages from an initial state to a more mature state (Gottschalk 2009). More mature in this context means better equipped to fulfill its purpose, i.e. having a higher level of sophistication, capability or availability of specific characteristics (Mettler et al. 2010). The basic components of a maturity model are (i) a number of overall maturity levels, (ii) a number of aspects or areas (henceforth called focus areas) that can be developed along a predefined evolutionary path to achieve the defined maturity levels and (iii) descriptions of each step on the evolutionary path. In addition, a maturity model may contain suggestions on how to perform the various steps in terms of improvement actions.

The best-known maturity model within the field of IS, and the one that was an inspiration to the development of many more maturity models, is SEI's Capability Maturity Model for Software (CMM) and its successors (Paulk et al. 1993). Over the past years more than a hundred maturity models have been developed for domains as varied as corporate data quality management (Hüner et al. 2009), e-learning (Marshall and Mitchell 2004), e-government (Andersen and Henriksen 2006) and offshore sourcing (Carmel and Agarwal 2002; Strutt et al. 2006), to name but a few. Benefits expected from maturity models are, among others, providing a roadmap to improve processes, stimulating management into more long term commitment, enhancing support for institutional planning by the ability of benchmarking and providing a means of organizing the diverse collection of ideas and heuristics (Marshall and Mitchell 2004).

The research literature reveals a number of distinguishing characteristics of existing maturity models. First of all, the nature of maturity models may be descriptive, prescriptive or comparative (De Bruin et al. 2005). This is directly related to the purpose of the model: whether it is meant purely to assess an organization's current state (descriptive), whether it is meant to suggest improvement actions (prescriptive), or whether it is aimed at comparison with other organizations (comparative). Most authors focus on the descriptive or comparative purpose of maturity models. Much less attention is paid to how to select improvement actions (Poepelbuss and Röglinger 2011).

A second characteristic is the dimension the model applies to. Many maturity models, including CMM, focus on only one dimension, often the process dimension. In general, however, performing a specific function involves more dimensions, such as people and objects, that can be addressed in their own right (Bharadwaj 2000; Feeney and Willcocks 1998; Mettler and Rohner 2009; Niazi et al. 2005; Ravichandran and Rai 2000). Hammer (2007) argues that for business processes to perform well and to sustain that performance, not only characteristics that apply to individual processes are needed, but also enterprisewide capabilities such as leadership, culture and expertise.

A third characteristic is the maturity growth structure. Many models adopt the five maturity levels defined by CMM. Other models define their own maturity levels. Usually the number of levels is somewhere between three and six. Each level has a label and, usually, a description of the characteristics of the level as a whole. Achievement of a level is measured either by having an aspect of the domain fully implemented (staged approach) or by having an aspect of the domain implemented to the extent required by the maturity level (continuous approach). The nature of the levels differs according to the focus of the model. In CMM, for instance, the levels are defined in terms of the degree of process management, i.e. *initial*, *repeatable*, *defined*, *managed*, *optimizing*. In other maturity models, the levels are defined in terms of resulting situations, usually referred to as stages, for instance the Public Sector Process Rebuilding model contains the levels *catalogue*, *transaction*, *vertical integration* and *horizontal integration* (Andersen and Henriksen 2006), while Gottschalk and Solli-Saether (2006) present a maturity model for IT outsourcing that distinguishes the stages *cost*, *resource* and *partnership*. In the domain of enterprise architecture management, Ross et al. (2006) discuss four stages that an organization passes through, making increasingly effective use of enterprise architecture: *business silo*, *standardized technology*, *optimized core* and *business modularity*.

Only a few models depart from the idea of a fixed number of maturity levels, distinguishing more than one dimension and defining maturity levels for each dimension. Examples are

the Corporate Data Quality Maturity Model (CDQ MM) by Hüner, Ofner and Otto (2009), the IS/ICT Management Capability Maturity Framework (Renken 2004) and the E-Learning Maturity Model (Marshall and Mitchell 2007).

An alternative framework to the fixed-level, one-dimensional maturity model approach can be found in the resource-based view (RBV) of the firm (Wernerfelt 1984; Barney 1991). The RBV considers companies as individual bundles of resources. The unique combination of resources a company possesses may provide it competitive advantage. The core concept in the RBV is the resource. Resources are assets (tangible and intangible) and capabilities (including skills and processes) that are available and useful in detecting and responding to market opportunities or threats (Wade and Hulland 2004). Originating in the field of management, the RBV has also been used in IS research, applying its ideas to IS resources. Examples of resources in the field of IS are *IS planning and change management* and *IS infrastructure*. Resources can be defined in varying degrees of generality and can be complex and multi-dimensional. Thus, a resource named manage external linkages can be divided into contract facilitation, informed buying, vendor development and contract monitoring as separate resources. Though the RBV originally presents a static and isolated view on resources, certain IS resources can only be developed over time and interaction with other resources is almost always an important factor in providing strategic benefits (Wade and Hulland 2004). The perspective of development of resources is further explored in the concept of dynamic capabilities (Teece et al. 1997; Eisenhardt and Martin 2000).

The choices made in a specific maturity model depend on the underlying rationale and conceptual model. The rationale defines what constitutes progressive maturity for the scoped functional domain as a whole as well as for each of the distinguished focus areas constituting the functional domain. However, few existing maturity models describe a rationale and conceptual model based on existing theories (Poeppelbuss et al. 2011).

As fixed-level maturity models are often difficult to implement (Brodman and Johnson 1994; Kuilboer and Ashrafi 2000; Reifer 2000; Zahran 1997), lacking step by step guidance (Huang and Han 2006), it is worthwhile to look for an alternative approach to measuring and improving IS function maturity. The main criticism on maturity models is the lack of a theoretical foundation for the existence of a fixed number of maturity levels (Solli-Saether and Gottschalk 2010) and the lack of guidance in defining improvement plans (Huang and Han 2006). The underlying conceptual model of the TPI model does not posit a fixed number of maturity levels defined a priori. Instead it starts out bottom up, building a step by step improvement schema that emerges from looking at the constituent parts of the functional domain. This is in line with the RBV and its applications in the field of IS. As IS functions are to be regarded as resources rather than processes, the RBV provides an appropriate conceptual framework for IS function maturity development.

3 Research method

As the objective of our research is to provide a metamodel, as stated in the introduction, a design-science research approach is appropriate (Hevner et al. 2004).

To put our research into context we use the extended framework for design science presented in Wieringa 2010. Wieringa adds the requirement of applicability to the well-known discussion of rigor versus relevance. Whereas relevance is context dependent because of the changeable goals of stakeholders, applicability is the more stable match between theory and the condition of practice of a concrete case. Wieringa 2010 divides IS design-science activities into practical problem solving and research question investigation. Within the latter he categorizes design-science research questions into three categories: problem investigation questions, artifact validation questions and conceptual questions. The research question we address in this paper belongs to the artifact validation category. We take an artifact that has proven its value in one domain and try to conceptually generalize it for use in other domains: from the area of practical problem solving we take the Test Process Improvement (TPI) model (Koomen and Pol 1999) which is input for the research question investigation. The scientific knowledge input for the research question investigation is the knowledge on maturity models and the resource-based view of the firm. The scientific knowledge produced is the focus area maturity model concept, which can be used by researchers to define incremental development paths for specific IS functions.

To structure our design-science process of developing the metamodel we chose the framework of Peffers et al. 2008. Peffers et al. (2008) define a design-science research process model that distinguishes six activities and four different entry points to the design-science research process: problem-centered, objective-centered, design and development-centered and client/context initiated approaches. As we took an existing artifact which we analyzed and generalized to enable application in other domains, our approach should be considered a design and development-centered approach. A design and development-centered approach "... would result from the existence of an artifact that has not yet been formally thought through as a solution for the explicit problem domain in which it will be used" (Peffers et al. 2008, p.56). Figure 1 provides an overview of our research approach.

We use the six activities in design-science research Peffers et al. distinguish to describe our research approach:

1. **Problem identification and motivation.** The problem motivating our research is how to develop capabilities in a given functional domain in an incremental, balanced manner. As argued in the introduction, it is important for organizations to fully develop their IS capabilities. Research shows, however, that this is not a straightforward matter (Hammer 2007, Repenning and Serman 2002). Therefore, practitioners and researchers are looking for well-founded development paths. Existing fixed-level maturity models are lacking in providing detailed guidance. In the field of testing we found a Test Process Improvement model that seems to meet the objective of providing step by step guidance. We successfully applied its underlying structure to the fields of enterprise architecture management and software product management. It is worthwhile to investigate whether the model underlying TPI can be put to broader use in the field of IS.
2. **Define the objectives for a solution.** The TPI model emerged from practice and represents many years of practical experience in testing. We want to broaden its applicability to other IS functions. For this purpose our objective is to conceptualize the TPI into a generic type of maturity model.

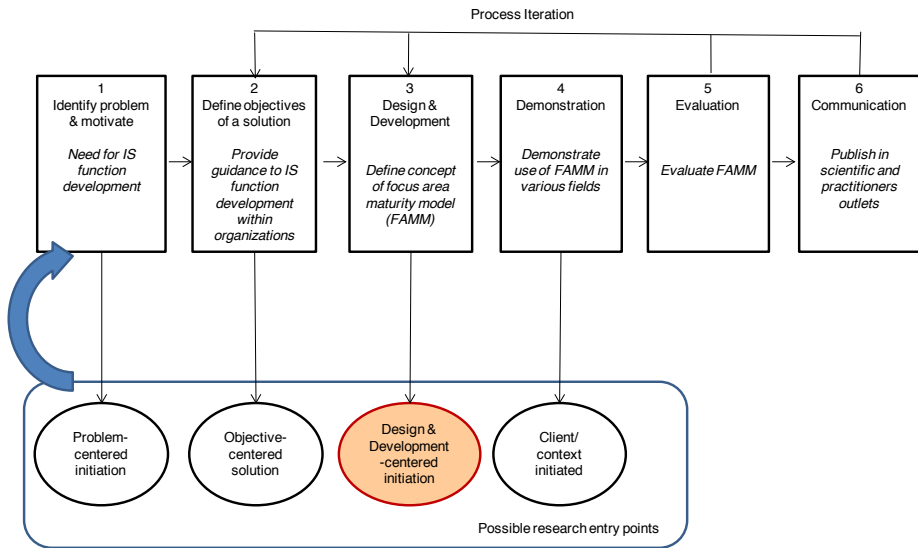


Figure 1. Design science research methodology process model (after Peffers et al. 2008)

3. **Design and development.** We took the TPI model and its analogous application to enterprise architecture management and software product management, analyzed the underlying structure and generalized it to a generic type of maturity model, which we label focus area maturity model. We developed a metamodel, precisely defined the constituent parts of the metamodel and mathematically formalized it. To explicate the differences between the focus area maturity model and the common fixed-level maturity model we use the framework of general design principles for maturity models by Poepelbuss and Röglinger 2011.
4. **Demonstration.** We demonstrate the use of the concept of focus area maturity model by discussing its application in the two domains of enterprise architecture management and software product management. The first application of the focus area maturity model outside testing was in the field of enterprise architecture management. This demonstrated the applicability of the concept outside the testing domain. Next the concept was applied to the domain of software product management. This confirmed the applicability of the concept in other domains, and also extended the model with situational specificity. Both applications also demonstrate how the issues of theoretical foundation and step by step guidance are dealt with in a focus area maturity model.
5. **Evaluation.** Though we extensively evaluated the applications of the focus area maturity model to the fields of enterprise architecture management and software product management, rigorous evaluation of the concept itself is still in progress and not completed yet. In order to facilitate such evaluation we designed a focus area maturity model devel-

opment method which we applied to the field of smart grid security. However, further evaluation of the development method as well as the results is needed. This should include measurement of both ease of implementation and impact on organizational performance.

6. **Communication.** Besides communication of the focus area maturity model in the scientific community by publication in conferences and journals, application of the model to enterprise architecture management is published in practitioners' books.

The result of the design and development activity is presented in section 4. Section 5 discusses the demonstration of the focus area maturity model in the fields of enterprise architecture management and software product management.

4 Definition of focus area maturity model

Before providing a definition of the focus area maturity model concept we explain its structure with the aid of the instantiation we developed for the field of enterprise architecture management.

Development of a focus area maturity model starts with defining the focus areas that a specific functional domain consists of. For each of these focus areas it then defines a series of progressively mature capabilities. A capability is the ability to achieve a certain goal, making use of the available resources (cf. Bharadwaj 2000). The capabilities are specific to the focus areas they are related to. An overall incremental development path is defined by juxtaposing all capabilities of all focus areas relative to each other in a matrix. This is done by placing capabilities that are to be developed after already developed capabilities further to the right in the matrix. Capabilities can follow other capabilities either because they depend on the existence of the earlier capabilities or because it has practical advantages to develop one capability after the other. Determining the order of capabilities is done on the basis of theoretical knowledge, empirical knowledge and expert knowledge of the functional domain. An example of a focus area maturity model for the domain of enterprise architecture management is given in figure 2 (Van Steenbergen et al. 2010).

The focus areas are in the left column. The letters (A to D) to the right of the focus areas, stand for the progressively mature capabilities related to each of the focus areas. For example, the focus area *Development of architecture* has the capabilities A: *Architecture undertaken as project*, B: *Architecture as a continuous process* and C: *Architecture as a facilitation process*, representing a progression in maturity. The focus area *Use of architecture* has three capabilities A: *Architecture used informatively*, B: *Architecture used to steer content* and C: *Architecture integrated into the organization*. The positioning of the letters in the matrix indicates the order in which the capabilities of the different focus areas can be implemented to develop the enterprise architecture functional domain in a balanced manner.

The fourteen columns in the matrix of figure 2 define the overall maturity levels, with level 0 being the lowest level and level 13 the highest. The rightmost column for which an organization has achieved all focus area capabilities positioned in that column and in all columns to its left, indicates the overall maturity level of that organization. This overall maturity level can be

Focus Area	Maturity Level	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Development of architecture			A			B			C						
Use of architecture				A			B				C				
Alignment with business			A				B				C				
Alignment with the development process				A				B		C					
Alignment with operations						A			B			C			
Relationship to the as-is state						A				B					
Roles and responsibilities					A		B					C			
Coordination of developments								A			B				
Monitoring					A		B		C		D				
Quality management									A		B				C
Maintenance of the architectural process								A		B		C			
Maintenance of architectural deliverables						A			B						C
Commitment and motivation			A					B		C					
Architectural roles and training					A		B			C				D	
Use of an architectural method					A						B				C
Consultation				A		B				C					
Architectural tools								A				B			C
Budgeting and planning					A							B		C	

Figure 2: A focus area maturity model for the functional domain of enterprise architecture management (Van Steenberg et al. 2010)

visualized by colouring the cells of each row up until the first capability that has not been implemented yet by the organization, giving a maturity profile of the organization. The rightmost column that contains only coloured cells, indicates the maturity level of the organization assessed. Thus, the organization in figure 2 is at maturity level 1, as the capability A of *Use of architecture* in column 2 has not been achieved. It also shows an imbalance: the focus area *Alignment with the development process*, is fully developed, while others, like *Use of architecture* and *Monitoring*, are not yet developed at all. To get a balanced enterprise architecture management function, this organization first has to develop the focus area *Use of architecture* to its first capability (the A in column 2), achieving overall maturity level 2. The next step would be to develop *Monitoring* to the first capability (the A in column 3), progressing to level 3.

For the purpose of developing a functional domain step by step, the focus area maturity model is more appropriate than the fixed-level maturity model, because it allows a more fine-grained approach. Because the nature of the focus area maturity model enables it to distinguish more levels of maturity, it allows for defining smaller steps between the levels and for expressing more dependencies between focus areas. Thus practitioners receive more guidance in setting priorities when developing their functional domain.

As regards theoretical foundation, the fact that the focus area maturity model for a functional domain is assembled from its constituting focus areas, allows for better theoretical grounding. The focus area maturity model does not imply an a priori set of maturity stages that are hard to theoretically justify. Instead it recognizes the differences between focus areas and allows for defining progressively mature capabilities per focus area. The definition of the specific focus areas

and their capabilities for a specific functional domain can, and should, be grounded in empirically tested theory about the functional domain itself.

Below we present precise definitions of each of the concepts mentioned (table 1) as well as a metamodel (figure 3). To provide an overview, however, we first give an informal description of the focus area maturity model and its constituent parts in their mutual relations.

A focus area maturity model is used to establish the *maturity* of an organization in a specific functional domain. A focus area maturity model must have a well-defined scope in the sense of the functional domain it applies to. A *functional domain* is described by the set of *focus areas* that constitute it. With each focus area a set of *capabilities* is associated. The capabilities are juxtaposed relative to each other in a *maturity matrix*. A capability is positioned to the right of another capability if it is to be implemented after that capability because it is *dependent* on its realization. Besides hard dependencies, pragmatic reasons may also inform this positioning. Based on the positioning of the capabilities in the maturity matrix a number of *maturity levels* can be distinguished. To establish the actual maturity of an organization in the functional domain, an *assessment instrument* based on the maturity matrix is used. The assessment instrument contains *assessment questions* linked to the capabilities. Answering the assessment questions for a specific organization produces a *filled-in assessment*. The filled-in assessment can be translated into a *maturity profile* for the organization, depicted by the coloured cells in the maturity matrix. To guide the organization in incremental development of the functional domain, *improvement actions* are associated with the capabilities. All of this together constitutes the *focus area maturity model*.

The metamodel of the focus area maturity model is given in figure 3. We use the notation presented by Van de Weerd and Brinkkemper (2008), which is based on standard UML conventions, with some minor adjustments.

In the appendix a mathematical formalization of the focus area maturity model is provided.

Looking at the elements of the focus area maturity model from the perspective of the resource-based view we regard focus areas as resources and capabilities, as the name already suggests, as capabilities. Capabilities may use or produce assets to achieve their goals. Which assets are used, or in what way, is organization-dependent and may change over time.

We do not focus on the relation between IT functions and competitive advantage in this paper. A discussion of how this relation can be made, using the resource attributes defined in the resource-based view, can be found in (Patas 2012). Note, however, that we apply the concepts of the resource-based view slightly different from (Patas 2012), as we focus on the capabilities of an organization rather than its assets.

To compare the focus area maturity model with the fixed-level maturity model we use the framework of design principles for maturity models defined by Poeppelbuss and Röglinger (2011). Poeppelbuss and Röglinger distinguish three types of design principles: basic design principles, principles for descriptive use and principles relevant to prescriptive use. The basic design principles (DP 1.1 – 1.4, *ibid.*) concern the application domain, purpose of use and target group. They also include the design process and definition of the central constructs. The design principles for a descriptive purpose of use (DP 2.1 – 2.2, *ibid.*) concern the criteria for measuring maturity as well as the assessment methodology. The design principles for a prescriptive purpose of use (DP 3.1 – 3.3, *ibid.*) concern improvement measures and how to select improvement measures.

<i>Concept</i>	<i>Definition</i>
<i>Maturity</i>	Maturity indicates the degree of development.
<i>Functional Domain</i>	A Functional Domain is the whole of activities, means, responsibilities and actors involved in the fulfillment of a well-defined function within an organization.
<i>Focus Area</i>	A Focus Area is a well-defined coherent subset of a Functional Domain. The total set of focus areas is a partition of the functional domain, i.e. different focus areas are disjoint and the union of all these focus areas is the complete functional domain.
<i>Capability</i>	A Capability is the ability to achieve a predefined goal.
<i>Dependency</i>	A capability is Dependent on another capability if it can only be achieved after that other capability has been achieved.
<i>Maturity Matrix</i>	A Maturity Matrix provides a partial ordering of Capabilities within a Functional Domain across Focus Areas over a sequence of Maturity Levels.
<i>Maturity Level</i>	A Maturity Level is a well-defined evolutionary plateau within a Functional Domain.
<i>Assessment Instrument</i>	An Assessment Instrument is a tool to determine Maturity within a Functional Domain.
<i>Assessment Question</i>	Assessment Questions are used to determine the current or target Maturity Level of an organization within a Functional Domain.
<i>Filled-in Assessment</i>	A Filled-in Assessment is the set of Assessment Questions together with their answers as provided during a particular assessment.
<i>Maturity Profile</i>	A Maturity Profile is a specific set of Capabilities within a Functional Domain that has been achieved by an organization.
<i>Improvement Action</i>	An Improvement Action is the description of an activity that is expected to result in achieving a specific Capability.
<i>Maturity Model</i>	A Maturity Model is an instrument to assess and develop the ability of an organization to perform within a Functional Domain.

Table 1. Definitions of concepts of the focus area maturity model

The differences between the fixed-level maturity model and the focus area maturity model are found in the basic and prescriptive design principles. Table 2 summarizes the differences (the coding of the design principles is taken from Poepfelbus and Röglinger). The focus area maturity model does not presuppose specific maturity levels. Instead its theoretical foundation is to be based on the functional domain it applies to. Maturation is not defined by a preset number of

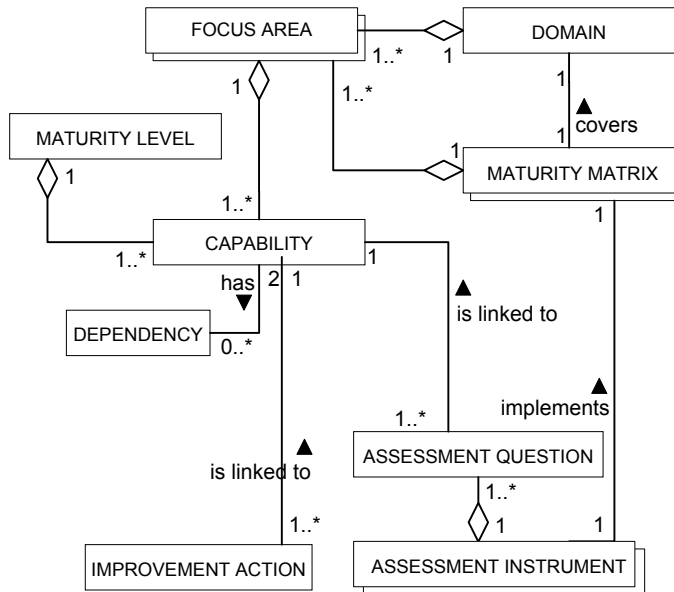


Figure 3. Metamodel of focus area maturity model

stages, but by the maturation aspects of the individual constituent parts of the domain and the dependencies between them. This allows for a finer granularity. The same difference in granularity exists for the improvement measures: the focus area maturity model defines improvement measures at the level of focus area capabilities and the dependencies between them instead of at the level of overall maturity levels.

Because of the finer granularity, focus area maturity models provide better step by step guidance for improvement.

5 Application of the focus area maturity model to other IS functions

The concept of focus area maturity model has been applied to the domains of enterprise architecture management and software product management. These two applications of the concept of focus area maturity model are briefly presented here. A more extensive discussion can be found in (Van Steenbergen et al. 2007, Van Steenbergen et al. 2010, Bekkers et al. 2010 and Van de Weerd et al. 2010).

<i>Design Principle</i>	<i>Fixed-level model</i>	<i>Focus area model</i>
DP 1.2 Definition of central constructs related to maturity and maturation	Maturation is defined in terms of pre-defined stages. Global maturation steps. Rationale is based on conceptual model of generic dimension.	Maturation is defined in terms of the constituent aspects. Specific maturation steps. Rationale is based on conceptual model of specific functional domain.
DP3.1 Improvement measures for each maturity level and level of granularity	Improvement measures are grouped at the level of pre-defined maturity levels.	Improvement measures are grouped at the level of focus area capabilities and the dependencies between them.
DP 3.2 Decision calculus for selecting improvement measures	Support for selection of improvement measures is limited to the level of pre-defined maturity levels.	Support for selection of improvement measures can be provided at the level of focus area capabilities and the dependencies between them.
DP 3.3 Target group-oriented decision methodology	Decision methodology is based on selecting focus areas and/or pre-defined maturity levels.	Decision methodology is based on selecting focus areas and/or dependencies between focus area capabilities.

Table 2. Comparison of fixed-level and focus area maturity models.

5.1 A focus area maturity model for enterprise architecture management

The Dynamic Architecture Maturity Matrix (DyAMM) is the application of the focus area maturity model in the field of enterprise architecture management (figure 2). Development of the DyAMM is embedded in a larger program developing an approach to enterprise architecture management that focuses on a goal-oriented, evolutionary development of the architectural function, called Dynamic Architecture (Van Steenberg et al. 2010). In the DyAMM, maturity is defined as the degree to which all aspects of the enterprise architecture practice are implemented. The DyAMM has been applied to over 70 organizations. A number of organizations use the DyAMM to give direction to an improvement program of years, performing a yearly assessment to monitor progress.

Besides the definition of focus areas, each with their own progressively mature capabilities, the DyAMM also contains an assessment instrument. Each capability is associated with one to four yes/no assessment questions to assess its implementation and one or more improvement actions that may support achieving it. Maturity is assessed by answering the yes/no questions. A capability is considered achieved, only if all questions associated with it are answered confirma-

tively. Table 3 shows as an example the questions associated with capability A of the focus area *Use of architecture*. In all, there are 137 assessment questions.

The DyAMM assessment instrument is primarily meant to be applied by independent assessors. The assessors fill the matrix by answering all 137 questions, basing their answers on interviews with relevant stakeholders and studying documentation. However, the DyAMM is also used as a self-assessment to be completed by individuals for their own organization. Organizations can answer the 137 questions, which are available on-line (DYA 2013), for themselves, and acquire a maturity profile as depicted in figure 2.

<i>Nr.</i>	<i>Question</i>
9	Is there an architecture that management recognizes as such?
10	Does the architecture give a clear indication of what the organization wants?
11	Is the architecture accessible to all employees?

Table 3. Questions to measure maturity level A of focus area *Use of architecture*.

5.2 A focus area maturity model for software product management

The second field of application is software product management (SPM). Many software companies have made a shift from developing custom-made software to developing product software. To cope with this shift, software companies need to introduce the right SPM processes. The Software Product Management Matrix (SPMM) was developed to support analysis and incremental improvement of SPM processes. It is based on the previously developed SPM Competence Model, that presents 14 SPM processes divided over four business functions: Portfolio management, Product planning, Release planning and Requirements management (Bekkers et al. 2010).

The focus areas of the SPMM are divided into four groups, corresponding to the four business functions identified in the earlier published SPM Competence Model. The number of capabilities within a focus area varies from three (A-C) to six (A-F). For a capability to be achieved it must be institutionalized and documented. In figure 4, the SPMM is presented.

With the SPMM a Situational Assessment Method (SAM) is associated (Bekkers and Spruit 2010). The SAM adds situationality to the SPMM by allowing for the fact that not all capabilities apply to every form of organization. It disables capabilities that do not apply, based on situational factor (SF) values of that organization. In addition, a selection stage is added in the SAM in which method fragments that fit the organization are selected and suggested to the organization. The method fragments are stored in a Knowledge Base, organized according to the capabilities they cover and to the SF restrictions under which they apply. The SAM can select the best-suited method fragments by looking at the method fragments which cover those

Maturity Level	0	1	2	3	4	5	6	7	8	9	10
Focus Area											
Requirements management											
Requirements gathering		A		B	C		D	E	F		
Requirements identification			A			B		C			D
Requirements organizing				A		B		C			
Release planning											
Requirements prioritization			A		B	C	D			E	
Release definition			A	B	C				D		E
Release definition validation					A			B		C	
Scope change management				A		B		C		D	
Build validation					A			B		C	
Launch preparation		A		B		C	D		E		F
Product planning											
Roadmap intelligence				A		B	C		D	E	
Core asset roadmapping					A		B		C		D
Product roadmapping			A	B			C	D		E	
Portfolio management											
Market analysis					A		B	C	D		E
Partnering & contracting						A	B		C	D	E
Product lifecycle management					A	B			C	D	E

Figure 4: The maturity matrix for Software Product Management (Van de Weerd et al. 2010)

capabilities, and comparing the results from the situational context of the organization to the SF restrictions of the method fragments (Bekkers and Spruit 2010).

5.3 Discussion

Both maturity models are based upon existing conceptual models. Thus, in the SPMM, the focus areas are deduced from the previously developed Reference Framework for Software Product Management (Bekkers et al. 2010). The main activities that are carried out by a product manager identified in this framework are directly transformed into focus areas for the maturity matrix. The DyAMM, in turn, has its base in the DyA model developed in the DyA program (Van Steenberg et al. 2010).

In developing the models, use is made of well-founded research methods like questionnaires, case studies and statistical analysis. Validation of the models is done in a number of ways. The DyAMM was validated firstly by applying it in a number of cases. This led to an adjustment of the model in a very early stage. After this adjustment the model was validated in a number of new cases (Van Steenberg et al. 2007), which did not lead to further adjustments. Finally, the DyAMM has been quantitatively validated on a repository of 56 assessments that was collected

in the period 2005-2008. This led to a few adjustments in the assessment questions (Van Steenberg et al. 2010). In 2012 a new version of the DyAMM is developed, based on literature research, feedback from additional assessments and consultation with expert users of the matrix. For the SPMM, the capabilities were identified in a brainstorming session with four SPM experts. After the brainstorming session, the results were compared with the existing SPM literature and, if necessary, refined or redefined. Finally, the capabilities were iteratively refined in expert interviews until agreement was reached. To position the SPM capabilities in the matrix, first an initial positioning was done based on the dependencies identified in the previous steps and on the experience of the researchers. The SPMM was tested in 12 case studies in companies of varying sizes. Subsequently, the maturity matrix was validated with expert validation and a survey among 45 product managers and product management experts. In this survey, participants were asked to position the different capabilities in the order they would implement them in their own organization (Van de Weerd et al. 2010).

The applicability of a focus area maturity model in practice is illustrated by companies that have been using the DyAMM over many years to evolve their architecture practice and consequently established greater effectiveness of the practice.

The maturity models are made available to practitioners in various ways. Thus, the DyAMM is published in a practitioner book (Van den Berg and Van Steenberg 2006) as well as in the form of an on-line self-assessment service (DYA 2013).

6 Conclusions

In this paper we establish the focus area maturity model for incrementally developing functional domains within the field of IS. We position focus area maturity models as an alternative to the fixed-level maturity models frequently encountered in both research literature and industry. Unlike the fixed-level maturity models, focus area maturity models are based on the identification of specific maturity levels per focus area. This leads to the distinction of more refined maturity levels, allowing for the recognition of more dependencies between focus areas and the definition of smaller steps between the levels. This incremental nature of focus area maturity models explicitly supports practitioners in setting priorities when executing capability improvement programs.

Existing research literature is strongly dominated by studies of fixed-level maturity models. Though some authors recognize the limitations of fixed-level maturity models, a comprehensive analysis of other types of maturity models is still lacking. With the research presented in this paper we aim to fill this gap. Using the resource-based view of the firm as a conceptual framework, we recognize IS functions and their constituent focus areas as resources to be developed over time.

The focus area maturity model approach presented in this paper provides researchers with an empirically demonstrated model to describe well-founded and feasible evolutionary development paths for functional domains. For practitioners a specific focus area maturity model, such as DyAMM or SPMM, provides an instrument that not only supports them in charting the current state of a functional domain, but above all provides detailed guidance in executing

a capability improvement program for the domain. It is to be expected that the need for guidance in developing functional domains will continue. The focus area maturity model is a means to not only explicate what target to strive for, but also how to move toward the achievement of that target.

There are some limitations to our research. Most importantly, we still have to complete an extensive evaluation of the concept of focus area maturity model. Though it has received wide acceptance in the fields of enterprise architecture management and software product management, validation of the relation between higher maturity and higher positive impact still has to be done.

We see two important venues for further research. The first is to validate the relation between maturity and organizational impact building on the insights of the resource-based view. This can only be done for a specific maturity model, such as the DyAMM or the SPMM and requires either a longitudinal study or a cross-sectional survey. The second venue of research, which is currently under way, is to extend the focus area maturity model concept with a validated development method.

7 Acknowledgement

We would like to thank the editor and anonymous reviewers for their valuable comments and improvement suggestions for earlier versions of this paper.

8 Appendix: Mathematical formalization

To thoroughly penetrate the concept of focus area maturity model, we developed a precise mathematical formalization.

We start with the concept of functional domain, which we will consider as a set FD of otherwise unspecified elements (representing activities, actors and responsibilities). This set is made up of all focus areas, so a focus area is a subset of FD and in fact FD can be partitioned into a number of focus areas: $FD \dot{\cup}_{FA \in I} FA$.

Here I denotes the set of all focus areas related to FD and the dot on top of the union symbol denotes that the union is disjoint, i.e., different focus areas have no elements in common. The size of I , i.e. the number of focus areas, depends on the functional domain, e.g., 18 for the domain of enterprise architecture and 16 for software product management. Another fundamental concept comes from the assessments organizations have to pass in order to reach a certain maturity level for a specific focus area. To keep terminology straight, we will distinguish the term ‘(overall) maturity level’ related to the complete functional domain from the term ‘level’ related to a focus area. We therefore introduce a totally ordered set (L, \leq_L) of levels and since an assessment is specific for a pair consisting of a focus area and a level, we are interested in the Cartesian product $I \times L$. We abstract away from the ‘assessment’ and concentrate on the set $I \times L$. Since not every element of I needs to have the same number of levels, this Cartesian product is in general

too large. For the general definition of maturity matrix we allow subsets C of $I \times L$. In the two example matrices, C denotes the set of capabilities and the pairs $(FA, l) \in C$ correspond to the cells in the matrix that are filled with a capital letter. From these examples we derive that C is not just a set, but a partially ordered one. The columns in the example matrices are the final concept we need and are formally described by a specific mapping ML from C to the natural numbers. Putting it all together gives the following definition. A maturity matrix consists of

1. A triple $(I, (L, \leq_L), (C, \leq_C))$ where I is a set, (L, \leq_L) is a completely ordered set and (C, \leq_C) is a partially ordered set with $C \subseteq I \times L$. Moreover, the ordering on C respects the ordering on L in the sense that if $c_1 = (FA, l_1)$, $c_2 = (FA, l_2) \in C$ and $l_1 \leq_L l_2$ then $c_1 \leq_C c_2$.
2. An order preserving mapping $ML: C \rightarrow N$ with $\text{Im}(ML) = \{1, \dots, m\}$ for some $m \in N$.

As an example take the SPMM where I is the set of 16 focus areas, $L = \{A, B, \dots\}$ is the set of 6 levels (so $I \times L$ consists of 96 elements), and L is totally ordered in the obvious way ($A < B < \dots$). Furthermore, C is the set of 63 capabilities, consisting of specific pairs (FA, l) where $FA \in I$, $l \in L$ and C is partially ordered by the intra- and inter-process capability dependencies, e.g., relations of the form $(FA, A) < (FA, B)$ (intra-process) and relations of the form $(FA_1, l_1) < (FA_2, l_2)$ where $FA_1 \neq FA_2$ (inter-process). Finally, the mapping ML assigns every capability to one of the numbers 1 through 12 while preserving the order (so if $c_1 \leq_C c_2$, then $ML(c_1) \leq ML(c_2)$).

The overall maturity level of an organization can now be defined. Since an organization that just started the development of a functional domain could very well have none of the capabilities defined for this domain, it makes sense to allow a zero maturity level. Even if they have some capabilities of maturity level 1, but not all of them, we still define their overall maturity level as zero. Only if they have all capabilities of maturity level 1 (i.e. all capabilities of the set $ML^{-1}(1)$), then their overall maturity level will be 1 or higher.

In general, if C_A is the set of capabilities of the organization (C_A is a subset of C), then the *maturity level* of that organization is the maximum value l for which $ML^{-1}(\{1, \dots, l\}) \subseteq C_A$. Note that if we substitute $l=0$ the set $ML^{-1}(\{1, \dots, l\}) = ML^{-1}(\emptyset) = \emptyset$ is a subset of C_A , so this definition also holds if C_A is empty or if C_A does not contain all capabilities with maturity level 1 (in both cases the maturity level of the organization will be 0).

We still need to formalize the concepts dealing with assessments. To every capability c , a set Q_c of yes/no questions is related (e.g., in DyAMM, these sets have 1 to 4 elements). The assessment instrument consists of the set of all these questions together. If we denote this set by Q , then $Q = \bigcup_{c \in C} Q_c$. A filled-in assessment is then a mapping $F: Q \rightarrow \{\text{yes}, \text{no}\}$ and this mapping determines the maturity profile of an organization as the set $\{c \in C \mid F(Q_c) = \{\text{yes}\}\}$. So a capability c belongs to the maturity profile if *all* questions related to c are answered with 'yes'.

9 References

- Andersen, K.V., and Henriksen, H.Z., (2006). E-government Maturity Models: Extension of the Layne and Lee Model. *Government Information Quarterly*, (23): 236-248.
- Barney, J., (1991). Firm Resources and Sustained Competitive Advantage. *Journal of Management*, (17:1): 99-120.

- Bharadwaj, A.S., (2000). A Resource-Based Perspective on Information Technology Capability and Firm Performance: An Empirical Investigation. *MIS Quarterly*, (24:1): 169-196.
- Bekkers, W., and Spruit, M.R., (2010). The Situational Assessment Method Put to the Test: Improvements Based on Case Studies. In: *Proceedings of the Fourth International Workshop on Software Product Management*, Sydney, Australia.
- Bekkers, W., Weerd, I. van de, Spruit, M., and Brinkkemper, S., (2010). A framework for process improvement in software product management. In: *Proceedings of the 17th European Conference, EuroSPI 2010, Communications in Computer and Information Science* (99): 1-12.
- Berg, M. van den, and Steenberg, M. van., (2006). *Building an Enterprise Architecture Practice*, Springer, Dordrecht.
- Brodman, J.G. and Johnson, D.L., (1994). What Small Businesses and Small Organizations Say About the CMM. In: *Proceedings of the 16th International Conference on Software Engineering*, IEEE Computer Society Press, Los Alamitos, CA, pp. 247-262.
- Bruin, T. de, Freeze, R., Kulkarni, U., and Rosemann, M., (2005). Understanding the Main Phases of Developing a Maturity Assessment Model. In: *Proceedings of the 16th Australasian Conference on Information Systems*, Sydney, Australia.
- Carmel, E. and Agarwal, R., (2002). The Maturation of Offshore Sourcing of Information Technology Work. *MIS Quarterly Executive*, (1:2): 65-77.
- Cleven, A., Winter, R., and Wortmann, F., (2012). Managing Process Performance to Enable Corporate Sustainability: A Capability Maturity Model. In: *Green Business Process Management*, J. vom Brocke et al. (eds.), Springer-Verlag, Berlin Heidelberg, pp. 111-129.
- DYA. (2013). <http://www.dya.info/en/would-you-set-work-straight-away/check-your-architectural-maturity>.
- Eisenhardt, K.M. and Martin, J.A., (2000). Dynamic Capabilities: what are they? *Strategic Management Journal*, (21): 1105-1121.
- Feeney, D.F. and Willcocks L.P., (1998). Core IS Capabilities for Exploiting Information Technology. *Sloan Management Review*, (39:3): 9-21.
- Gottschalk, P., and Solli-Saether, H., (2006). Maturity model for IT outsourcing relationships. *Industrial Management & Data Systems*, (106:2): 200-212.
- Gottschalk, P. (2009). Maturity Levels for Interoperability in Digital Government. *Government Information Quarterly*, (26): 75–81.
- Hammer, M., (2007). The Process Audit. *Harvard Business Review*, (April): 1-16.
- Hansen, B., Rose, J., and Tjornehoj, G., (2004). Prescription, description, reflection: the shape of the software process improvement field. *International Journal of Information Management*, (24): 457-472.
- Helfat, C.E., and Peteraf, M.A., (2003). The Dynamic Resource-Based View: Capability Lifecycles. *Strategic Management Journal*, (24): 997-1010.
- Hevner, A.R., March, S.T., Park, J., and Ram, S. (2004). Design Research in Information Systems Research, *MIS Quarterly* (28:1): 75-105.
- Huang, S., and Han, W., (2006). Selection priority of process areas based on CMMI continuous representation. *Information & Management*, (43): 297-307.
- Hüner, K.M., Ofner, M., and Otto, B., (2009). Towards a Maturity Model for Corporate Data Quality Management. In: *SAC'09*, March 8-12, Honolulu, Hawaii.

- Kohlegger, M., Maier, R., and Thalmann, S., (2009). Understanding Maturity Models Results of a Structured Content Analysis. In: *Proceedings of I-KNOW '09 and I-SEMANTICS '09*, Graz, Austria, pp. 51-61.
- Koomen, T., and Pol, M., (1999). *Test Process Improvement, a Practical Step-by-step Guide to Structured Testing*, Addison-Wesley, Boston.
- Kuilboer, J.P., and Ashrafi, N., (2000). Software Process and Product Improvement: An Empirical Assessment. *Information and Software Technology*, (42:1): 27-34.
- Marshall, S., and Mitchell, G., (2004). Applying SPICE to e-Learning: An e-Learning Maturity Model? In: *Sixth Australian Computing Education Conference (ACE2004), Conferences in Research and Practice in Information Technology* (30), R. Lister and A. Young (eds.), Dunedin.
- Marshall, S., and Mitchell, G., (2007). Benchmarking international e-learning capability with the e-learning maturity model. In: *Proceedings of EDUCAUSE in Australasia*, Melbourne, Australia.
- McCormack, K., et al., (2009). A Global Investigation of Key Turning Points in Business Process Maturity. *Business Process Management Journal*, (15:5): 792-815.
- Mettler, T., and Rohner, P., (2009). Situational Maturity Models as Instrumental Artifacts for Organizational Design. In: *Proceedings of the 4th international Conference on Design Science Research in information Systems and Technology*, Philadelphia.
- Mettler, T., Rohner, P., and Winter, R., (2010). Towards a Classification of Maturity Models in Information Systems. In: *Management of the Interconnected World*, A. D'Attri et al. (eds.), Springer-Verlag, Berlin Heidelberg, pp. 332-340.
- Montealegre, R., (2002). A Process Model of Capability Development: Lessons from the Electronic Commerce Strategy at Bolsa de Valores de Guayaquil. *Organization Science*, (13:5): 514-531.
- Niazi, M., Wilson, D., and Zowghi, D., (2005). A Maturity Model for the Implementation of Software Process Improvement: an Empirical Study. *Journal of Systems and Software*, (74:2): 155-172.
- Patas, J., (2012). Towards Maturity Models as Methods to Manage IT for Business Value – A Resource-based View Foundation. In: *AMCIS 2012 Proceedings*, paper 16.
- Paulk, M., et al., (1993). Key practices of the capability maturity model, Version 1.1. *Technical Report CMU/SEI-93-TR-25*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.
- Peffer, K., Tuunanen, T., Rothenberger, M.A., and Chatterjee, S., (2008). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, (24:3): 45-78.
- Poepelbus, J., and Röglinger, M., (2011). What makes a useful maturity model? A framework of general design principles for maturity models and its demonstration in business process management. In: *Proceedings of the European Conference on Information Systems (ECIS)*, paper 28.
- Poepelbus, J., Niehaves, B., Simons, A., and Becker, J., (2011). Maturity Models in Information Systems Research: Literature Search and Analysis. *Communications of the Association for Information Systems*, (29:1): 505-532.
- Ravichandran, T., and Rai, A., (2000). Quality Management in Systems Development: An Organizational System Perspective. *MIS Quarterly*, (24:3): 381-415.

- Reifer, D.J., (2000). The CMMi: It's Formidable. *Journal of Systems and Software*, (50:2): 97-98.
- Renken, J., (2004). Developing an IS/ICT Management Capability Maturity Framework. In: *Proceedings of SAICSIT 2004*, pp. 53-62.
- Repenning, N.P., and Sterman, J.D., (2002). Capability Traps and Self-Confirming Attribution Errors in the Dynamics of Process Improvement. *Administrative Science Quarterly*, (47:2): 265-295.
- Rosemann, M., and de Bruin, T., (2005). Towards a Business Process Management Maturity Model. In: *Proceedings of the European Conference on Information Systems (ECIS)*, Regensburg, Germany.
- Ross, J.W., Weill, P., and Robertson, D.C., (2006). *Enterprise Architecture as Strategy*, Harvard Business School Press, Boston, Massachusetts.
- Scott, J.E., (2007). Mobility, Business Process Management, Software Sourcing, and Maturity Model Trends: Propositions for the IS Organization of the Future. *Information Systems Management*, (24:2): 139-145.
- Solli-Saether, H., and Gottschalk, P., (2010). The Modeling Process of Stage Models. *Journal of Organizational Computing and Electronic Commerce*, (20:3): 279-293.
- Steenbergen, M. van, Berg, M. van den, and Brinkkemper, S., (2007). A Balanced Approach to Developing the Enterprise Architecture Practice. In: *Enterprise Information Systems*, J. Filipe, J. Cordeiro, and J. Cardoso (eds.), LNBP 12, pp. 240-253.
- Steenbergen, M. van, Schipper J., Bos, R., and Brinkkemper, S., (2010). The Dynamic Architecture Maturity Matrix: Instrument Analysis and Refinement. In: *ICSOC/ServiceWave 2009*, A. Dan, F. Gittler, and F. Toumani (eds.), LNCS 6275, Springer-Verlag, Berlin Heidelberg, pp. 48-61.
- Strutt, J.E., Sharp, J.V., Terry, E., and Miles, R., (2006). Capability maturity models for offshore organisational management. *Environment International*, (32): 1094-1105.
- Teece, D.J., Pisano, G. and Shuen, A., (1997). Dynamic Capabilities and Strategic Management. *Strategic Management Journal*, (18:7): 509-533.
- Wade, M., and Hulland, J., (2004). Review: The Resource-Based View and Information Systems Research: Review, Extension, and Suggestions for Future Research. *MIS Quarterly*, (28:1): 107-142.
- Weerd, I, van de, Brinkkemper, S., (2008). Meta-modeling for Situational Analysis and Design Methods. In: *Handbook of Research on Modern Systems Analysis and Design Technologies and Applications*, M.R. Syed and S.N. Syed (eds.), Idea Group Publishing, Hershey, pp. 38-58.
- Weerd, I, van de, Bekkers, W. and Brinkkemper, S., (2010). Developing a Maturity Matrix for Software Product Management. In: *Proceedings of the 1st International Conference on Software Business*, S. Jansen and P. Tyrvaïnen (eds.), LNBP 51, pp. 76-89.
- Wernerfelt, B. (1984). A Resource-Based View of the Firm. *Strategic Management Journal* (5:2): 171-180.
- Wieringa, R., (2010). Relevance and Problem Choice in Design Science. In: *Proceedings of the 5th International Conference on Design Science Research in Information Systems and Technology*, R. Winter, J.L. Zhao and S. Aier (eds.), LNCS 6105, Springer-Verlag, Berlin Heidelberg, pp. 61-76.
- Zahran, S., (1997). *Software Process Improvement: Practical Guidelines for Business Success*, Addison-Wesley, Reading, MA.

