A Design Science Research Perspective on Maturity Models in Information Systems

Author: Tobias Mettler
Date of Publication: May 12, 2009
Report Number.: BE IWI/HNE/03

Universität St. Gallen
Hochschule für Wirtschafts-, Rechts- und Sozialwissenschaften (HSG)

Institute of Information Management
Chair of Prof. Dr. Robert Winter
Mueller-Friedberg-Strasse 8
CH-9000 St. Gallen
Switzerland
Tel.: + 41 (0) 71 224 2420
Fax: + 41 (0) 71 224 2777

A DESIGN SCIENCE RESEARCH PERSPECTIVE ON MATURITY MODELS IN INFORMATION SYSTEMS

Abstract

In order to ascertain and measure dedicated aspects of information systems 'maturity', a wide range of assessment models have been developed by both, practitioners and academics over the past years. In spite of its broad proliferation, the concept has not been untroubled by criticism. Unnecessary bureaucracy, poor theoretical foundation, and the impression of a falsified certainty to achieve success are just a few examples for that. As there is still a significant lack of knowledge on how to design theoretically sound and widely accepted maturity models, it is the aim of this paper to discuss the typical phases of maturity model development and application by taking a design science research perspective. Given that both, development and application are intimately connected, different decision parameters are identified that are relevant in respect to rigour and relevance of the maturity model.

Keywords: Assessment Models, Design Decisions, Design Principles, Design Science Research, Maturity Models, Organisational Engineering

1 INTRODUCTION

The concept of maturity models is increasingly being applied within the field of Information Systems (IS), both as informed approach for continuous improvement (e.g. Paulk et al. 1993; Ahern et al. 2004) or as means of self or third-party assessment (Hakes 1996; Fraser et al. 2002). Since the initial finding in the 1970's (cf. Gibson & Nolan 1974; Crosby 1979), a multiplicity of different instantiations have been developed in science and practice. This popularity of maturity models was especially intensified by the introduction of the Capability Maturity Model (CMM) in the late 1980's and its successors ISO/IEC 15504 also known as SPICE (Software Process Improvement and Capability determination) and BOOTSTRAP (Haase et al. 1994; Kuvaja 1999). However, as the organisations constantly face the pressures to obtain and retain competitive advantage, invent and reinvent new products and services, reduce costs and time to market, and enhance quality at the same time, the need for and the development of new maturity models will certainly not diminish given that they assist decision makers to balance these sometimes divergent objectives on a more or less comprehensive manner. Furthermore, by incorporating formality into the improvement activities, decision makers can determine if the potential benefits have been realised or not. In addition, "conflicts of interest can be avoided by using a measurement model developed externally to the organisation" (Fraser & Vaishnavi 1997).

Nonetheless, the concept has not been untroubled by criticism. For instance, Pfeffer & Sutton (1999) argued that the purpose of maturity models is to identify a gap which can then be closed by subsequent improvement actions. However, lots of these models do not describe how to effectively perform these actions. This 'knowing-doing gap' can be very difficult to close, even when the decision makers know what to do. On the other hand, when exclusively focusing on the processes and objects assessed by maturity models, a kind of 'falsified certainty' is given to the decision makers. For example, CMM has been criticised because of its overemphasis on the process perspective and its disregard on people's capabilities (Bach 1994). Besides that, a too strong focus on formalisation of improvement activities accompanied by extensive bureaucracy can hinder people from being innovative (Herbsleb & Goldenson 1996). The most important point of critique on maturity models is, however, its poor theoretical basis (Biberoglu & Haddad 2002). Most of the models base on 'good practice' or 'success factors' derived from projects that have demonstrated favourable results to an organisation or an industry sector. Moreover, data for an appraisal frequently depends on the people being asked (also referred to as key informant bias). Thus being compliant with the maturity model would not necessarily guarantee that an organisation would achieve success. There is no agreement on a 'one true way' to assure a positive outcome (Montoya-Weiss & Calantone 1994). According to de Bruin & Rosemann (2005) the reasons for these sometimes ambiguous results of maturity models lays in the insufficient emphasis on testing the models in terms of validity, reliability and generalisability and on the little documentation on how to develop and design such a model. On that account, de Bruin & Rosemann describe in their work a first design methodology for the development of 'maturity assessment models'. However, as the approaches for developing and evaluating design science research (DSR) significantly advanced in the last few years, it is the aim of this paper to engross the thoughts in two directions. First, from a developer's perspective, we want to detail the seminal work of de Bruin & Rosemann by presenting particularised decision parameters for building and testing a maturity model. Second, we want to extend this one-sided reflexion by the user's perspective, showing all phases required for successfully implementing a maturity model in an organisation.

In order to achieve this goal, the paper is organised as follows. First, we give a definition of the term 'maturity' and describe the possible dimensions of progressing the levels of maturity in IS. The section that follows is dedicated to the clarification of the form and function of maturity models and its foundation within DSR. Basing on the prior results, we then discuss the designer's and the user's perspective on maturity models, subsequently. Finally, we present some concluding remarks and offer some suggestions for future research endeavours.

2 DEFINING MATURITY IN INFORMATION SYSTEMS

2.1 Processes, objects and people as fractions of maturity

In general, ‘maturity’ can be defined as “the state of being complete, perfect or ready” (Simpson & Weiner 1989). Maturity thus implies an evolutionary progress in the demonstration of a specific ability or in the accomplishment of a target from an initial to a desired or normally occurring end stage.

In the constituent literature about maturity models the term ‘maturity’ is in most instances reflected on a unidimensional manner, either focussing on (1) process maturity, i.e. to which extent a specific process is explicitly defined, managed, measured, controlled, and effective (e.g. Paulk et al. 1993; Fraser & Vaishnavi 1997), (2) object maturity, i.e. to which extent a particular object like a software product, a company report or similar reaches a predefined level of sophistication (e.g. Gericke et al. 2006), or on (3) people capability, i.e. to which extent the workforce is able to enable knowledge creation and enhance proficiency (e.g. Nonaka 1994).

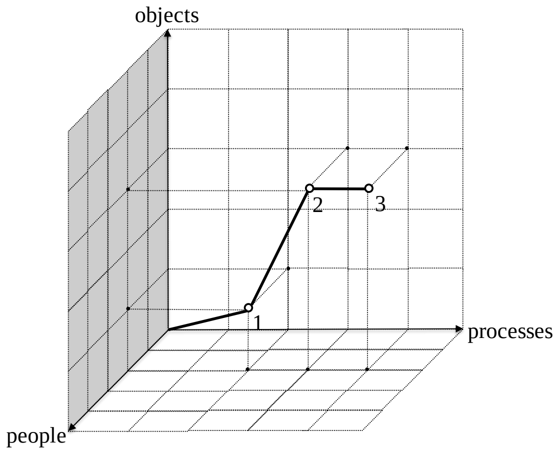

Figure 1. Different ways of progressing the level of maturity.

The commonly used basis for assessing maturity in IS are therefore people, processes or objects[1] (in the following referred to as maturity factors). Weinberg (1992) was the first of a series of authors who delineated their strong dependency and effect on maturity with respect to software engineering. However, as mentioned already, their mutual influence is not always given (cf. *Figure 1*).

In the first case, the level of sophistication is increased for all the three maturity factors. This can be designated as the ideal case and is obtained when the model designed for reproducing the state of maturity likewise considers an improvement of object, process, and people abilities (e.g. learn the workforce how to collect customer requirements in a standardised manner by using standardised means). In the second case, only two factors are enhanced, notably object and process maturity (e.g. using a customer relationship management system instead of writing the customer requirements into unstructured documents). And finally, maturity can be enhanced in one direction only (e.g. using a notebook to collect the data directly at the customer's site). Understanding these different ways of progressing the level of maturity is a crucial task in the development of a maturity model and especially important when testing the model with respect to its validity.

[1] We prefer the term ’object’ instead of ‘product’ because it comprises more than the final outcome to the customer like for example intra- and inter-organisational services and infrastructure.

2.2 Relation between maturity and diffusion

Another way to reflect maturity in IS can be found in the seminal work of Rogers (1962) and Utterback (1971) on the emergence and diffusion of innovations. Diffusion research centres on the conditions, which increase or decrease the likelihood that a technological idea, product or practice will be adopted. *Diffusion* is thereby understood as the "process by which an innovation is communicated through certain channels over a period of time among the members of a social system" (Rogers 1962). Innovations either can be categorised as *product innovations*, i.e. the involvement of a new or improved object whose characteristics differ significantly from previous ones, or as *process innovations*, i.e. the involvement of a new or significantly improved practices to enhance production and/or delivery of objects (OECD 1997). In both cases the difference in the characteristics to the prior solutions may emerge due to the application of new technologies, knowledge or materials. The enhancement of people's capabilities is thus implicitly presupposed. Nevertheless, the theories of emergence and diffusion of innovations reveal interesting conditions with respect to the understanding and development of maturity in IS (and the respective models for reproducing it).

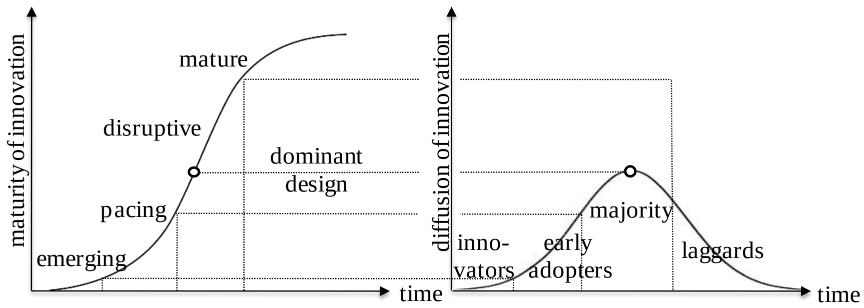

Figure 2. Interrelation between maturity and diffusion of innovations on the basis of (Rogers 1962; Utterback & Abernathy 1975).

According to Utterback & Abernathy (1975) the progress of a particular innovation follows an s-curve (cf. left hand side of Figure 2). As time goes by, innovation (mainly originating from many minor product or process improvements) passes through different stages of maturity. Of particular interest is thereby the disruptive phase where a dominant design of a solution (i.e. a general agreed standard or best practice) becomes evident. However, dominant designs may not be better than other designs, but acceptance of an innovation will be on peak. As regards to the development of maturity models the cognition of the state in which an innovation is situated is thus extremely important, especially when the model is prescriptive. For instance, when focussing on the assessment of emerging innovations the levels of maturity may be extremely uncertain given that no dominant design is found already. Recommended improvement activities therefore probably will be estimated as speculation. On the other hand, when concentrating on mature innovations the levels of maturity are clear but the potential for improvement is only marginal. In such a case the results from an appraisal may be understood as 'bureaucracy' or 'platitude' since no substantial benefits may be gained. A similar train of thoughts can be made when considering the diffusion of innovations (cf. right hand side of Figure 2). When using too fundamental or forward-looking criteria for the maturity assessment of an organisation, the application of the model will show an accumulation of the results on a predefined sophistication level (e.g. Hayes & Zubrow (1995) discovered that 73% of the assessed organisations between 1987-1994 were stuck in the CMM-level 1 because the requirements of the process area 'project management' were far too hard to meet). Thus when defining the levels of maturity for a particular domain, a tradeoff between the state of an innovation's uncertainty and its actual diffusion (which assists in predicting whether and how an innovation will be successful) has to be considered in order to guarantee 'useful insights' (i.e. trustworthy but not too obvious improvement activities) from the application of the model.

3 MATURITY MODELS AS SUBJECT OF DESIGN RESEARCH

Design oriented research has a long tradition in the field of IS, especially in Europe. However, numerous contributions have been made by Non-European scholars as well, especially Simon's foundational "The Sciences of the Artificial" (1969). While natural sciences try to explain and predict behavioural aspects of reality (e.g. from an individual, group, organisation and market perspective) by developing and verifying theories (March & Smith 1995), design oriented research aims at building and evaluating 'artificial solutions', in order to extend existing capability limitations (Hevner et al. 2004). According to Hevner et al., the construction and evaluation of these solutions can be reflected on a generic level, focussing primarily on the *design research process* and on creating standards for its rigour (by some authors designated as 'science of design' (Cross 2001) or 'design science' (McKay & Marshall 2007)) or on a specific level, aiming at creating *artefacts* (i.e. design research products) to concrete classes of relevant problems by using rigorous scientific methods for its construction and evaluation (also referred to as 'design research' (Gericke & Winter 2008)). March & Smith (1995) differentiate (1) constructs (the *language* to specify problems and solutions), (2) models (the *representation* of the identified problems and future solutions), (3) methods (the *procedure* how to solve these problems and develop the future solutions), and (4) instantiations (the *physical conversion* as proof-of-concept of the prior artefacts) as unique artefact types. This understanding has been broadly adopted by way of example in (Hevner et al. 2004; Peffers et al. 2006; Vahidov 2006). However, some scholars demand an enlargement of this classification by including 'design theories' as a fifth artefact type (e.g. Walls et al. 1992; Venable 2006; Gregor & Jones 2007) in order to provide a sounder basis for arguing for the rigour and legitimacy of design-oriented research. Hence, the question raises what artefact type a maturity model is associated with: a model, a method, or a theory? We consider each of these three perspectives before describing how to develop a maturity model in the subsequent section.

3.1 Model versus Method

In IS, modelling normally is applied "in the analysis stage of systems development when abstract models of the represented system and its organizational environment are created" (Wand et al. 1995). Models generally represent a formal description of "some aspects of the physical or social reality for the purpose of understanding and communicating" (Mylopoulos 1992). Depending on the notion of the representation, they either are descriptive (i.e. they give an unprejudiced reproduction of some aspects of reality), explanatory (i.e. they deliver a depiction of causal connections to better understand reality) or predictive (i.e. they recommend an efficient solution state of a future reality). However, no matter the kind of form, all of them reflect the *state* of a particular application domain whether it is the exact description of the current situation or a suggestion for a more efficient or ideal target state.

In contrast, methods are used "to perform a systems development project, based on a specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities with corresponding development products" (Brinkkemper 1996). Consequently, methods are systematic (i.e. they deliver rules on how to act and instructions on how to solve problems), goal-oriented (i.e. they stipulate standards on how to proceed or act to achieve a defined goal), and repeatable (i.e. they are inter-subjectively practicable) (Braun et al. 2005). As opposed to models where state descriptions (*what*) are the root of the matter, methods rather focus on the specification of activities (*how*).

Maturity models, although numerously developed and applied in the past years, lack a clear definition and contextualisation. When reviewing the constituent literature, the current understanding of maturity models in IS tends to be somehow in-between the prior described artefact types, as they normally describe the typical behaviour exhibited by a person or organisation at a number of predefined levels of maturity for each of several maturity factors of the area under study (cf. Section 2) and of the procedures which might be regarded as appropriate to reach the next level (cf. Figure 3). In summary, they combine state descriptions (i.e. the maturity levels) with a number of key practices (i.e. the improvement activities) to address a series of goals.[2]

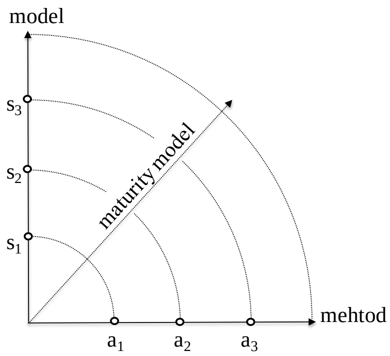

Figure 3. Positioning of maturity models in-between models and methods.

3.2 Theory versus (IT-)Artefact

Different semantics and individual views on theory and theorising in DSR complicate the fixation of what is understood as *design theory*. Gregor & Jones (2005) define design theories as "principles inherent in the design of an IS artefact that accomplishes some end, based on knowledge of both IT and human behaviour". According to them, eight separate components can be differentiated: (1) purpose and scope (i.e. the meta-requirements of the problem solution), (2) constructs (i.e. the entities of particular interest), (3) principles of form and function (i.e. the meta-description of the problem solving), (4) artefact mutability (i.e. possible state changes in the design of the solution), (5) testable propositions (i.e. truth statements about the theory), (6) justificatory knowledge (i.e. underlying kernel theories), (7) principles of implementation (i.e. concrete description of the problem solving process), and (8) an expository instantiation (i.e. the physical implementation). Whereas the first six components are mandatory, the last two are optional parts of a design theory.

In reference to maturity models it is difficult to take a firm stand whether they are theory or not. Depending on the maturity and diffusion of the studied phenomenon (cf. Figure 2) the ways in progressing the levels of maturity may be clear (mature innovations) or extremely uncertain (emerging innovations). This has an extraordinary influence on the accomplishment of the mentioned theory requirements. When building a maturity model for a highly innovative phenomenon, justificatory knowledge to be based upon is weak or missing and principles of form and function are unclear as no dominant design prevails. Furthermore, the required cases to derive maturity levels and recommendations may be missing as well. As a consequence, testable propositions will probably remain untested since no person or firm has accomplished the final maturity requirements. Therefore the resulting maturity model is rather an artefact than a theory. On the other hand, if the studied phenomenon is mature, the knowledge base for building and the cases for testing the design propositions are available and design mutability is more or less manageable. Accordingly, all necessary and sufficient conditions for building a design theory are given for a maturity model addressing a ‚mature' domain.

4 MATURITY MODEL DEVELOPMENT AND APPLICATION

According to Simon (1969), "everyone designs who devises courses of action aimed at changing existing situations into preferred ones". In the literature, either design decisions for the development

[2] Maturity models should not be confused with *maturity grids*, which aim at illustrating the number of levels of maturity in a simple, textual manner and thus are comparable with Likert-scale questionnaires with anchor phrases (cf. Fraser et al. 2002).

(e.g. de Bruin & Rosemann 2005) or the application (e.g. Gallagher & Shrum 2004) of maturity models are discussed. In our opinion, the development of the maturity model is intimately connected with the application phase and therefore should not be reflected separately (cf. Figure 4). Furthermore, whilst design decisions within the generic phases may vary, the order of the phases is crucial (e.g. decisions made when preparing the deployment of the maturity model will impact the further application of the model). To illustrate this, we will reflect the most important decision parameters from a developer's and a user's perspective in the following sections.

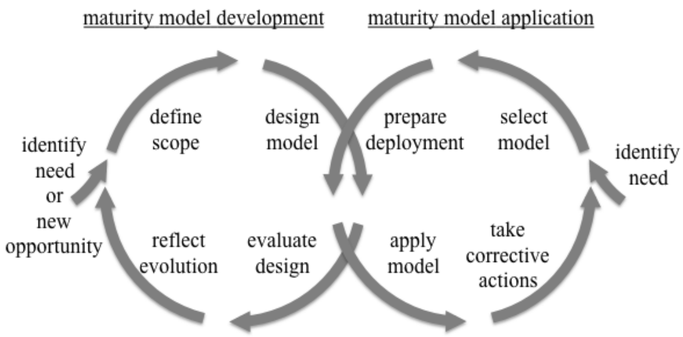

Figure 4. Two sides of the same coin? Phases of maturity model development and application.

4.1 The developer's perspective

Developing maturity models by conducting design-oriented research means finding solution patterns for important unsolved problems or giving advice in solving problems in more effective or efficient ways (cf. Hevner et al. 2004). As DSR is a problem-oriented approach, the development cycle is usually initiated by a "need and require intention" (Purao 2002). However, according to Järvinen (2007) a business need is not necessarily required but a new opportunity as "opportunity-based innovation can have a great economic value". The complete development cycle consists of four phases: (1) define scope, (2) design model, (3) evaluate design, and (4) reflect evolution (cf. Figure 4).

<table>
<tr><th>Phase</th><th>Decision parameter</th><th colspan="12">Characteristic</th></tr>
<tr><td rowspan="5">Define scope</td><td>Focus / breadth</td><td colspan="6">General issue</td><td colspan="6">Specific issue</td></tr>
<tr><td>Level of analysis/ depth</td><td colspan="3">Group decision-making</td><td colspan="3">Organisational considerations</td><td colspan="3">Inter-org. considerations</td><td colspan="3">Global & societal considerations</td></tr>
<tr><td>Novelty</td><td colspan="3">Emerging</td><td colspan="3">Pacing</td><td colspan="3">Disruptive</td><td colspan="3">Mature</td></tr>
<tr><td>Audience</td><td colspan="4">Management-oriented</td><td colspan="4">Technology-oriented</td><td colspan="4">Both</td></tr>
<tr><td>Dissemination</td><td colspan="6">Open</td><td colspan="6">Exclusive</td></tr>
<tr><td rowspan="6">Design model</td><td>Maturity definition</td><td colspan="3">Process-focussed</td><td colspan="3">Object-focussed</td><td colspan="3">People-focussed</td><td colspan="3">Combination</td></tr>
<tr><td>Goal function</td><td colspan="6">One-dimensional</td><td colspan="6">Multi-dimensional</td></tr>
<tr><td>Design process</td><td colspan="4">Theory-driven</td><td colspan="4">Practitioner-based</td><td colspan="4">Combination</td></tr>
<tr><td>Design product</td><td colspan="4">Textual description of form</td><td colspan="4">Textual description of form and functioning</td><td colspan="4">Instantiation (assessment tool)</td></tr>
<tr><td>Application method</td><td colspan="4">Self-assessment</td><td colspan="4">Third-party assisted</td><td colspan="4">Certified professionals</td></tr>
<tr><td>Respondents</td><td colspan="3">Management</td><td colspan="3">Staff</td><td colspan="3">Business partners</td><td colspan="3">Combination</td></tr>
<tr><td rowspan="3">Evaluate design</td><td>Subject of evaluation</td><td colspan="4">Design process</td><td colspan="4">Design product</td><td colspan="4">Both</td></tr>
<tr><td>Time-frame</td><td colspan="4">Ex-ante</td><td colspan="4">Ex-post</td><td colspan="4">Both</td></tr>
<tr><td>Evaluation method</td><td colspan="6">Naturalistic</td><td colspan="6">Artificial</td></tr>
<tr><td rowspan="3">Reflect evolution</td><td>Subject of change</td><td colspan="3">None</td><td colspan="3">Form</td><td colspan="3">Functioning</td><td colspan="3">Form and functioning</td></tr>
<tr><td>Frequency</td><td colspan="6">Non-recurring</td><td colspan="6">Continuous</td></tr>
<tr><td>Structure of change</td><td colspan="6">External / open</td><td colspan="6">Internal /exclusive</td></tr>
</table>

Table 1. Decision parameters during maturity model development.

In the ‘define scope phase’ the developer is faced with the most important design decisions (cf. Table 1). First, as it will influence all the following decision parameters, the focus of the phenomenon to be studied is set. By either choosing a generalistic (e.g. learning organisations as per Aggestam (2006)) or a more specific subject-matter (e.g. risk management (RM) for medical device companies as per Burton et al. (2006)) the breadth of the maturity model is defined. On the other hand, the level of analysis conditions the maturity model’s ‘operating altitude’. For instance RM can be explored on the group level (e.g. decision-making within a particular department of the firm), on the organisation level (e.g. RM integrated as part of corporate governance), on the inter-organisational level (e.g. collaborative RM between the medical device company and the health care organisations) or on a more global and societal level (e.g. societal safety aspects of RM). Furthermore, as ‘utility’ is in the centre of DSR interest (cf. Hevner et al. 2004), considerations about the novelty of the subject (e.g. focusing on a mature phenomenon but trying to give advice in a more effective or efficient way or solve new emerging problems), the audience (tailoring the model to the particular needs of a management-oriented or a technology-oriented audience, or both), and the dissemination of the model (open to the specified audience or exclusive access only) have to be taken as well.

When the scope is set, the actual maturity model is built in the ‘design model phase’. As described in Section 2, it is extremely important to have a clear understanding of what is meant by ‘maturity’. Having a process-focused understanding of maturity implies to centre on activities and work practices (e.g. inputs and outputs of specified tasks) in order to define more effective procedures. With an object-focused understanding, the features of work products (e.g. functional and non-functional requirements) are investigated in order to enhance their mode of operation. When comprehension of ‘maturity’ is inclined to people’s skills and proficiency, the emphasis of the model lies more on the soft capabilities (e.g. people’s feelings and behaviour). Through this clarification of ‘maturity’, the goal function of the model (i.e. the way how maturity is progressed) is tacitly influenced (e.g. efficiency is almost always the underlying goal of process-oriented maturity). Also it remains still important to ponder whether the progress of maturity is one-dimensional (i.e. solely focussing on one target measure like efficiency) or multi-dimensional (i.e. focussing on multiple, sometimes divergent goals or competitive bases). If the goal function is clear, the nature of the design *process* (e.g. theory-driven versus practitioner-based or a combination of both) has to be determined in order to identify the knowledge base for deriving the maturity levels, the metrics, and the corresponding improvement recommendations. This decision is of particular interest as it has a major influence on the choice of the research methods to be used (e.g. literature review versus focus group discussions) and on the scientific and practical quality of the resulting design product. The latter is also determined by the shape of the design product (whether it is a pure textual description of the form and/or functioning of the maturity model or if it is instantiated as software assessment tool), the application method (whether the data collection is based upon a self or a third-party assessment), and the setting of the respondents for data collection (e.g. management, staff, business partners or a combination). Certainly, design process and product is also strongly constrained by the skills of the developers and the resources available for building the model (e.g. academic and business partners, research and programming skills).

The ‘evaluate design phase’ is concerned with the verification and validation of the designed maturity model. In line with Conwell et al. (2000) *verification* is the process of determining that a maturity model ”represents the developer’s conceptual description and specifications with sufficient accuracy” and *validation* is the degree to which a maturity model is ”an accurate representation of the real world from the perspective of the intended uses of the model”. In doing so, Pries-Heje et al. (2008) state that it is a purposeful strategy to define the subject (what), the time-frame (when), and the method for evaluation (how). As regards to the subject of evaluation it is possible to test the design process (i.e. the way the model was constructed) or the design product (i.e. the model itself). In order to counter the before mentioned criticism on the rigour of maturity models, it is our opinion that both should be subject of the evaluation. Then, the developer has to decide the point in time of evaluation (ex-ante versus ex-post). This – in combination with the novelty of the topic and the availability of the required respondents – has an influence whether artificial (e.g. experiment) or naturalistic (e.g. case study) methods are used.

At last, in the ‘reflect evolution phase’, the design mutability of the model is contemplated. This is of particular importance – but for all that sometimes neglected – as, on the one hand, the maturity of the phenomenon under study is growing and therefore the model’s solution stages and improvement activities have to be refaced from time to time (e.g. modify requirements for reaching a certain maturity level due to the emergence of new best practices and technologies), on the other hand, changes in the form and function are needed to ensure the standardisation and global acceptance of the model (e.g. amend the model schema from a CMM to a CMMI-compliant structure). Therefore basically all-or-none can be subject of evolution; that is the form (e.g. the underlying meta-model or model schema of the maturity model), and the functioning (e.g. the way how maturity is assessed). Finally, it has to be determined whether evolution is a non-recurring or continuous matter and how change is ‘produced’. For instance, modifications can be openly induced by model users or exclusively by the developer himself.

4.2 The user’s perspective

The successful application of a maturity model normally passes through four phases: (1) select model, (2) prepare deployment, (3) apply model, and (4) take corrective actions (cf. Figure 4). In contrast to the maturity model development cycle, the application cycle always starts with a business need by reason that the application of a maturity model always is associated with extensive costs. Taking the user’s perspective the following decision parameters are identified (cf. **Fehler! Verweisquelle konnte nicht gefunden werden.**).

<table>
<tr><th>Phase</th><th>Decision parameter</th><th colspan="12">Characteristic</th></tr>
<tr><td rowspan="6">Select model</td><td>Origin</td><td colspan="6">Academic</td><td colspan="6">Practitioner-based</td></tr>
<tr><td>Reliability</td><td colspan="4">Untested</td><td colspan="4">Verified</td><td colspan="4">Validated</td></tr>
<tr><td>Practicality</td><td colspan="6">General recommendations</td><td colspan="6">Specific improvement activities</td></tr>
<tr><td>Accessibility</td><td colspan="6">Free</td><td colspan="6">Charged</td></tr>
<tr><td>Design mutability</td><td colspan="3">None</td><td colspan="3">Form</td><td colspan="3">Functioning</td><td colspan="3">Form and functioning</td></tr>
<tr><td>Application method</td><td colspan="4">Self-assessment</td><td colspan="4">Third-party assisted</td><td colspan="4">Certified professionals</td></tr>
<tr><td rowspan="5">Prepare deployment</td><td>Driver / Responsibility</td><td colspan="6">Business</td><td colspan="6">IT</td></tr>
<tr><td>Realisation</td><td colspan="6">Informal appraisal</td><td colspan="6">Formal assessment project</td></tr>
<tr><td>Application area</td><td colspan="6">Specific entity</td><td colspan="6">Multiple entities</td></tr>
<tr><td>Respondents</td><td colspan="3">Management</td><td colspan="3">Staff</td><td colspan="3">Business partners</td><td colspan="3">Combination</td></tr>
<tr><td>Training</td><td colspan="4">None</td><td colspan="4">Basic</td><td colspan="4">Extensive</td></tr>
<tr><td rowspan="2">Apply model</td><td>Execution</td><td colspan="6">Go</td><td colspan="6">No go</td></tr>
<tr><td>Frequency of application</td><td colspan="6">Non-recurring</td><td colspan="6">Repeated</td></tr>
<tr><td rowspan="3">Take corrective actions</td><td>Target setting</td><td colspan="6">Uncoupled</td><td colspan="6">Coupled</td></tr>
<tr><td>Implementation</td><td colspan="6">On the fly</td><td colspan="6">Project</td></tr>
<tr><td>Implementer</td><td colspan="4">Line organisation</td><td colspan="4">Staff organisation</td><td colspan="4">Externals</td></tr>
</table>

Table 2. Decision parameters during maturity model application.

The ‘select model’ phase starts with a broad search for potentially applicable maturity models with regard to the identified business need. As to date the knowledge base is poor and a maturity model classification or reference database is missing, this can be a very laborious endeavour. However, in order to limit the number of models found, criteria for selection are needed. For instance, these can be the origin of the model (i.e. whether it has its source from academia or practice), reliability (i.e. how well the maturity model has been evaluated), accessibility (i.e. if it is free for use or not), practicality of recommendations (i.e. whether the recommendations are problem-specific or more general in nature and hence need more detailing), and the method of application (e.g. self-assessment or an appraisal by certified professionals). In addition, design mutability (i.e. convertibility of model elements and ease of integration in existing organisational model base) is important as well, since synergy effects in respect of training, appraisals and improvement activities are left behind when applying multiple maturity models for different problem domains that are not integrated within and across the organisation. Ultimately, all these decision parameters yield to a suitable match with respect to cost (e.g. time and financial expenses) and value (e.g. the ease with which the produced results of the maturity model have a positive effect on business activities).

When a particular model is selected, the 'prepare deployment phase' begins. In this phase it is fundamental to find a potential sponsor or responsible for the appraisal. Additionally, formality of realisation has to be determined (e.g. rather informal appraisal versus formal assessment) and the corresponding application area and respondents must be located. Finally, training with the relevant stakeholders is conducted.

In the 'apply model phase' two basic decisions are identified. First, should the appraisal really be performed, and second, how many times it shall be executed. In the final 'take corrective actions phase', the appraisal results are critically reflected. It has to be decided whether the progress on maturity should be coupled or uncoupled of the regular target system of the organisation and whether the implementation of the improvement activities can be done on the fly or a specific project is needed and who should effect the corrective actions (e.g. the affected line organisation, the staff organisation, or external consultants).

5 CONCLUSION

With this paper, we introduced a definition of the term *maturity* and a contextualisation of the concept of *maturity models* in IS taking a design science research perspective. As knowledge on how to design theoretically sound and widely accepted maturity models is lacking, we further proposed, on the basis of the work of de Bruin & Rosemann (2005), a phase model for both, development and application of such models. To illustrate their mutual interrelation, all relevant decision parameters for each phase are discussed. Therewith, and given that no uncriticised standard design methodology exists so far, we intended to contribute to a subject matter of IS, which in practice, because of its complexity, is widely recognised, while it is somehow neglected in research. However, the proposed phase model is limited in that it is only supported by few research experiences. Nevertheless, our findings may provide a basis for further research.

In order to generalise the findings so far, it is therefore necessary to incorporate experiences from other domains and to extensively test the propositions, for instance by developing own maturity models or by investigating the user's behaviour during maturity model application. Moreover, the identification of useful classification criteria for maturity models and the development of a maturity model base will serve practice as well as research.

References

Aggestam, L. (2006). Towards a maturity model for learning organizations - The role of knowledge management. In: Proceedings of the 17th International Conference on Database and Expert Systems Applications, Krakow, Poland, 141-145.

Ahern, D. M., Clouse, A. and Turner, R. (2004) CMMI distilled: A practical introduction to integrated process improvement. 2nd Edition. Addison-Wesley, Boston, London.

Bach, J. (1994). The Immaturity of CMM. American Programmer 7 (9), 13-18.

Biberoglu, E. and Haddad, H. (2002). A survey of industrial experiences with CMM and the teaching of CMM practices. Journal of Computing Sciences in Colleges 18 (2), 143-152.
Braun, C., Wortmann, F., Hafner, M. and Winter, R. (2005). Method construction: A core approach to organizational engineering. In: Proceedings of the 20th Annual ACM Symposium on Applied Computing, Santa Fe, USA, 1295-1299.
Brinkkemper, S. (1996). Method engineering: engineering of information systems development methods and tools. Information and Software Technology 38 (4), 275-280.
Burton, J., McCaffery, F. and Richardson, I. (2006). A risk management capability model for use in medical device companies. In: Proceedings of the 2006 International Workshop on Software Quality, New York, USA, 3-8.
Conwell, C. L., Enright, R. and Stutzman, M. A. (2000). Capability maturity models support of modeling and simulation verification, validation, and accreditation. In: Proceedings of the 2000 Winter Simulation Conference, San Diego, USA, 819-828.
Crosby, P. B. (1979) Quality is free: The art of making quality certain. McGraw-Hill, New York.
Cross, N. (2001). Designerly ways of knowing: Design discipline versus design science. Design Issues 17 (3), 49-55.
de Bruin, T. 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.
Fraser, M. D. and Vaishnavi, V. K. (1997). A formal specifications maturity model. Communications of the ACM 40 (12), 95-103.
Fraser, P., Moultrie, J. and Gregory, M. (2002). The use of maturity models/grids as a tool in assessing product development capability. In: Proceedings of the IEEE International Engineering Management Conference, Cambridge, UK, 244-249.
Gallagher, B. P. and Shrum, S. (2004). Applying CMMI to systems acquisition. The Journal of Defense Software Engineering 8, 8-12.
Gericke, A., Rohner, P. and Winter, R. (2006). Vernetzungsfähigkeit im Gesundheitswesen - Notwendigkeit, Bewertung und systematische Entwicklung als Voraussetzung zur Erhöhung der Wirtschaftlichkeit administrativer Prozesse. HMD - Praxis der Wirtschaftsinformatik 251, 20-30.
Gericke, A. and Winter, R. (2008). Entwicklung eines Bezugsrahmens für Konstruktionsforschung und Artefaktkonstruktion in der gestaltungsorientierten Wirtschaftsinformatik. Working paper, Institute of Information Management, University of St. Gallen.
Gibson, C. F. and Nolan, R. L. (1974). Managing the four stages of EDP growth. Harvard Business Review 52 (1), 76-88.
Gregor, S. and Jones, D. (2007). The anatomy of a design theory. Journal of the Association for Information Systems 8 (5), 312-335.
Haase, V., Messnarz, R., Koch, G., Kugler, H. J. and Decrinis, P. (1994). Bootstrap: Fine-Tuning Process Assessment. IEEE Software 11 (4), 25-35.
Hakes, C. (1996) The corporate self assessment handbook. 3rd Edition. Chapman & Hall, London.
Hayes, W. and Zubrow, D. (1995). Moving on up: data and experience doing CMM-based software process improvement. Working paper, Carnegie Mellon Software Engineering Institute.
Herbsleb, J. D. and Goldenson, D. R. (1996). A systematic survey of CMM experience and results. In: Proceedings of the 18th International Conference on Software Engineering, Washington D.C., USA, 323-330.
Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004). Design science in information system research. MIS Quarterly 28 (1), 75-101.
Järvinen, P. (2007). On reviewing of results in design research. In: Proceedings of the 15th European Conference on Information Systems, St. Gallen, Switzerland, 1388-1397.
Kuvaja, P. (1999). BOOTSTRAP 3.0 - A SPICE conformant software process assessment methodology. Software Quality Control 8 (1), 7-19.
March, S. T. and Smith, G. G. (1995). Design and natural science research on information technology. Decision Support Systems 15, 251-266.

McKay, J. and Marshall, P. (2007). Science, design, and design science: Seeking clarity to move design science research forward in information systems. In: Proceedings of the 18th Australasian Conference on Information Systems Science, Design and Design Science, Toowoomba, 604-614.

Montoya-Weiss, M. M. and Calantone, R. J. (1994). Determinants of new product perfomance: A review and meta-analysis. Journal of Product Innovation Management 11 (5), 397-417.

Mylopoulos, J. (1992). Conceptual modelling and telos. In Conceptual modelling, databases and CASE (Loucopoulos, P. and Zicari, R., Eds.) 49-68, John Wiley and Sons, New York.

Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. Organization Science 5 (1), 14-37.

OECD (1997). The Oslo manual: Proposed guidelines for collecting and interpreting technological innovation data. OECD, Paris.

Paulk, M. C., Curtis, B., Chrissis, M. B. and Weber, C. V. (1993). Capability maturity model, version 1.1. IEEE Software 10 (4), 18-27.

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V. and Bragge, J. (2006). The design science research process: A model for producing and presenting information systems research. In: Proceedings of the 1st International Conference on Design Science in Information Systems and Technology, Claremont, USA, 83-106.

Pfeffer, J. and Sutton, R. (1999). Knowing "what" to do is not enough: Turning knowledge into action. California Management Review 42 (1), 83-108.

Pries-Heje, J., Baskerville, R. and Venable, J. (2008). Strategies for design science research evaluation. In: Proceedings of the 16th European Conference on Information Systems, Galway, Ireland, 255-266.

Purao, S. (2002). Design research in the technology of information systems: Truth or dare, http://purao.ist.psu.edu/working-papers/dare-purao.pdf, last accessed 09. 10. 2008.

Rogers, E. M. (1962) Diffusion of innovations. Free Press, New York.

Simon, H. A. (1969) The sciences of the artificial. MIT Press, Cambridge, MA.

Simpson, J. A. and Weiner, E. S. C. (1989). The Oxford English Dictionary. Oxford University Press, Oxford, UK.

Utterback, J. M. (1971). The process of technological innovation within the firm. Academy of Management Journal 14 (1), 75-88.

Utterback, J. M. and Abernathy, W. J. (1975). A dynamic model of process and product innovation. Omega 3 (6), 639-656.

Vahidov, R. (2006). Design researcher’s IS artifact: A representational framework. In: Proceedings of the 1st International Conference on Design Science Research in Information Systems and Technology, Claremont, USA, 19-33.

Venable, J. (2006). The role of theory and theorising in design science research. In: Proceedings of the 1st International Conference on Design Science in Information Systems and Technology, Claremont, USA, 1-18.

Walls, J., Widmeyer, G. and El Sawy, O. (1992). Building an information system design theory for vigilant EIS. Information Systems Research 3 (1), 36-59.

Wand, Y., Monarchi, D. E., Parsons, J. and Woo, C. C. (1995). Theoretical foundations for conceptual modelling in information systems development. Decision Support Systems 15, 285-304.

Weinberg, G. M. (1992) Quality software management: Systems thinking. Dorset House Publishing, New York.