ICED13
INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

OPERATIONALIZATION OF THE QUADRANT-BASED VALIDATION IN CASE OF A DESIGNERLY SOFTWARE DEVELOPMENT METHODOLOGY

Els DU BOIS (1,2), Imre HORVATH (1)
1: University of Antwerp, Belgium; 2: Delft University of Technology, The Netherlands

ABSTRACT

Systematic validation of design methods is important to advance the professional practice of engineering design. In contrast, we lack structured approaches for validation of design methodologies. In this paper we present the adaptation of an existing validation framework to external validation of a new, composite, designerly software development methodology (DSDM). A major challenge was that a comparative validation approach could not be considered in our single-case study. Literature review was done to find an appropriate reflexive validation method and useful validation criteria. Yet no effective software validation methodologies were found for our context. The validation quadrant approach however lent itself as framework of the sought validation method, which was adjusted, extended and operationalized by introducing series of validating steps. The quadrant-based external validation method (QEVM) combines structural and performance assessment actions in both the theoretical domain and the application domain. Our conclusion is that QEVM is useful in our single-case, reflexive assessments for the validation of the DSDM. Further research should focus on a context-less use of QEVM.

Keywords: design methodology, decision making, external reflective validation, quadrant based validation methodology, designerly software development methodology

Contact:
Els Du Bois
Artesis
Industrial Design Engineering
Antwerp
2000
Belgium
els.dubois@artesis.be

1 INTRODUCTION

The validation of design methods is important for (i) the continuing advancement of design theory for researchers, to guide the development and evaluation of new methodologies, and (ii) for the professional practice of engineering, to determine which methodology to employ as well as when and how to employ them. Much engineering research is based on mathematical modeling for validation, however other areas of engineering, rely on subjective statements (Sargent 2005). One such area is that of design methodologies within the field of engineering design. Validating these methodologies implies that the complexity, appearing when the methodology involves the use of human knowledge, can be measured (Seepersad, Pedersen et al. 2006).

In this paper we focused on the operationalization of the validation of a specific design methodology. A new designerly software development methodology (DSDM) was developed to face the development of software-enabled intelligent products. The objective of the research, described in this paper, was to externally validate the proposed specific software design methodology. To transfer the validation issue in the context of the DSDM, we realized that the external validation must be achieved using a reflective validation approach, since comparison was not possible as in the study only a single case was developed. As a matter of fact executing an additional comparative validation would imply the need for an extra research cycle to compare the simultaneous development of a specific software product using on the one hand the DSDM and on the other hand a traditional software development method. As evolving research means throughout the project, a single case software tool was developed to support designers in their decision making processes on smart energy saving using ubiquitous controllers (Du Bois and Horvath 2011).

The remainder of the paper is organized as follows: in Section 2 we dive deeper into the literature to find suitable validation methods for our methodology. In short, we could not directly find a specific generic external validation method for this context in the literature. However, we could derive one method called the validation square that seemed to be a generic external validation method, which we could use directly or after adaptation. In Section 3, the theoretical and methodological fundamentals of the validation method are discussed, plus the adjustments and extensions needed for our context. In addition, in Section 4, the operationalization of this quadrant-based method is detailed for the specific purpose. Then, Section 5 reports on the execution of the validation assessment of the DSDM and Section 6 shows the findings. Finally, in Section 7 some concluding propositions are formulated.

2 LITERATURE STUDY

2.0 Approach and objectives

The literature study was decomposed into two parts: (i) we looked for existing reflective validation methods that might be applied for external validation in our context, and (ii) we identified possible validation criteria. Only external validation of our software development methodology was in our interest since internal validation was already done during the development of the methodology, whereas the external validation could only be done at the end of the process when the findings are known. In Figure 1, which shows a general overview, we identified two approaches to validate a methodology: (i) a direct approach and (ii) an indirect approach. In indirect validation, the methodology is evaluated based on its impact on process change, people's satisfaction, process characteristics, behavior aspects, resources, etc. In addition another difference was made between (a) a reflective approach, (b) a comparative approach, and (c) a process-based approach. As discussed above, for this research a reflective approach is the best due to the single case context. For each validation approach, criteria are needed to determine the value and the goodness of the methodology. These criteria are studied in the second part of this literature study.

2.1 Findings about validation methods for software development methodologies

Validation depends on the purpose of the methodology and its intended use (Macal 2005), so considering the context of the validation while selecting the appropriate validation method is most important. In this part of the review, we analyzed existing generic external validation methods in the context of validating software development methodologies. One of the challenging research problems in validating a software engineering methodology (SEM), is to deal with the complexity that emerged because the SEM involves the use of human knowledge in its phases. To measure such knowledge, Lee and Rine (2004) use case study research design, which is an empirical research alternative in designing a research plan that establishes a logical link from the data to be collected to the initial questions of study. For an effective research case study, they say that it is necessary for the validation exercise to first have designed a case study specific to the characteristics of this invented SEM. On the other hand (Gottschalk 2002) advices to use surveys to gather empirical data for the validation of methodologies. Kitchenham et al. (1997) most important methods for software methodology validation are: formal experiments, quantitative case studies and feature analysis validation. Briand et al. (1995) do not only consider the empirical validation but also the theoretical validation of methodologies. Similarly, Schön and Argyris (1975) proposed a framework for evaluating methodologies that included checks on: (i) internal consistency, (ii) congruence with the espoused theory, (iii) testability of the theory, and, ultimately, (iv) effectiveness of the theory.

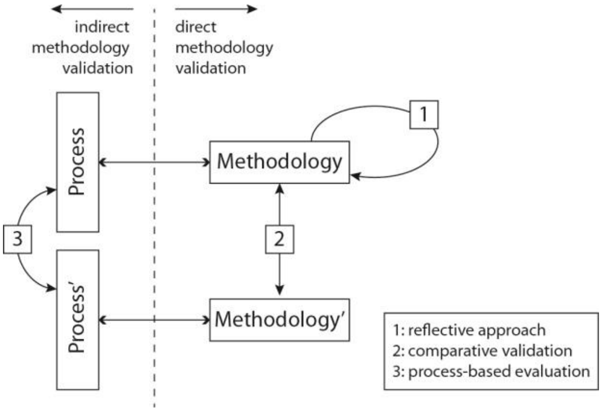

Figure 1: methodology validation possibilities

2.2 Generic methods for external validation of methodologies

Since there was no worthily specific methodology that is developed for validation in our context, related contexts were identified and applied validation methods in these contexts were discussed. A relation was found with software validation, design knowledge validation and model validation domains, and we also considered the domain of research methodology validation. **Software validation** is important in the development process to consider the user's point of view (Wallace and Fujii 1989). Most validation methods are using different kind of prototyping (Hickey and Dean 1998) that serve as the source of requirements and enhance the developers' understanding of the system objectives and functionalities. And validation of the prototype itself is also crucial (O'Keefe and O'Leary 1993). In the domain of **design knowledge validation**, different approaches could be found, we base on the literature review done by (Frey and Dym 2006). One framework suggested by (Simon 1990) emphasizes the fit between problem-solving behaviors and the problem environment, rather than the internal consistency of the behaviors. Schön and Argyris (1975) framework for validating theories can also be used related to professional practice. A similar framework was proposed by Pedersen et al. (2000) in which they suggest a balanced approach. Frey and Dym (2006) conclude from a comparison of design and medicine methodologies to use simulation models in validation where possible, since this technique has proven its quality in the medicine domain. To increase the confidence in a **simulation model** several well documented and comprehensive validation methods should be used combining several validation techniques. Landry et al. (1983) and Sargent (1984) defined five types of validity related to the modeling process: (i) conceptual, (ii) logical, (iii) experimental, (iv) operational, and (v) data validation. As discussed by (Jensen 1995; Kleijnen 1995), many people consider that empirical validation is a more powerful approach to validation. Nevertheless, in case of designing a new system, comparison is not possible, so (Carson 2002) compares the implemented model behavior with its assumptions and specifications. Lastly to discuss **research methodology validation**, we can base on Dellinger's (2007) overview of research validation approaches of whom the qualitative research validation seemed to be most related. Over the past few decades, many researchers have participated in these discussions. Lincoln and Guba (1985) suggested the need to develop an entirely different approach to assess validity than what are traditionally used by quantitative researchers. These theorists developed the concepts of trustworthiness, which corresponds with Campbell and Stanley's (1963) concepts of internal and external validity. Eisner (1991) took this one step further by not using the word validity but instead used the word credibility. Maxwell (1992) identified five types of validity: descriptive validity, interpretive validity, theoretical validity, generalizability, and evaluative validity. Eisenhart and Howe (1992) advocated for a collective validation construct in which general standards for conducting qualitative research should be used as guidelines.

2.3 Findings about validation criteria

In addition to a technique, method or framework to execute the validation, there should also be some criteria by which the proposed design methodologies are judged to ensure that their use will consistently yield the correct design, i.e., that these methods are valid. We could conclude from the literature study that also the validation criteria are context dependent. Here an overview is given of criteria from the context of software applications, (software) design methodology, model development and qualitative research validation. Software applications are according to (O'Keefe and O'Leary 1993; Bednar and Robertson 2007) validated using the following criteria: correctness, consistency, sufficiency, performance, necessity, level of expertise, builder's/user's risk, maintaining objectivity, and reliability. (Kitchenham, Linkman et al. 1997; Olewnik and Lewis 2003) describe the validation of design methodologies by the following criteria: (i) basic: it must be logical, complete, understandable, usable, internally consistent etc. (ii) use: it must be helpful, produce the specified, usable and relevant results, use meaningful reliable information, not bias the designer; (iii) gain: it must provide added value. According to (Jagdev, Browne et al. 1995; Kleijnen 1995; Sargent 2005) the two most important criteria for model validation are model accreditation (model satisfies criteria) and model credibility (confidence to use model and information derived, level acceptable to the user). Additionally, (Carson 2002) measures performance for industrial models by primary measures such as throughput, system cycle or response time, and work in process. In addition, a number of secondary or explanatory measures may be of interest, such as resource utilization, size of local buffers, and throughputs for subsystems or particular types. Many definitions of the various aspects of validity in qualitative research specifically refer to the how-to-dos of establishing credibility, authenticity, trustworthiness, criticality, and integrity, to name a few. Dellinger and Leech (2007) identify them as primary aspects. Secondary criteria refer to important and flexible aspects of quality criteria that are in addition to the primary criteria, including explicitness, vividness, creativity, thoroughness, congruence, and sensitivity.

3 THEORETICAL AND METHODOLOGICAL FUNDAMENTALS

3.1 Initial interpretation of the validation square

We conclude from the literature review that many methods exist to validate methodologies, models and products. In our search to find an appropriate validation method for the validation of the DSDM, we found a general validation approach could be adapted to our context using some specific changes and additions. For the specific purpose of validating the DSDM we will use the method of the validation square (Pedersen, Emblemsvåg et al. 2000), since it was developed to be engineering-oriented, and considered to be the most appropriate. What is important is that in this framework both empirical and theoretical validation aspects are considered. The validation square method is comparable to the framework proposed by Briand et al. (1995) and those of Schön and Agryris (1975), but its major advantage is that it handles and combines different levels of complexity (functional, structural elements, interfaces/communication, technical solutions). Considering the needed validation criteria, we can conclude that the proposed criteria are useful, but still need to be combined for our specific purpose. We used it as an empty framework which should be operationalized for our specific case. According to the Seepersad and Pedersen is the purpose of the ‘validation square’ (VS) method to introduce a rigorous framework for validating engineering design methods. As illustrated in Figure 2, the validation square is divided into four quadrants. Considering the theoretical and methodological fundamentals, the framework is based on two primary tasks: establishing (i) the structural validity of the design methodology (left half), and (ii) the performance validity of the design methodology (right half). In addition, there is also a division into a domain-independent and a domain-specific upper and lower half, the latter is associated with the validity of the method for the domain-specific examples investigated in the research, the matter for broader domains of application. The theoretical parts has a predictive nature while the practical part has a reflective approach for the validation.

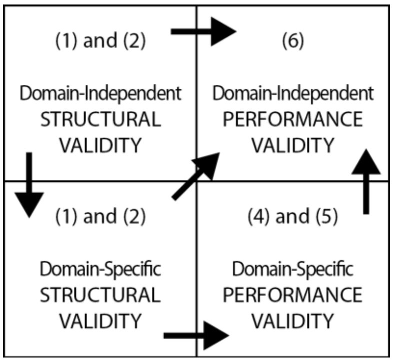

Figure 2: the validation square

3.2 Re-interpretation of the method in application context

An important next step is to transfer the validation square principle into a validation method that is specialized for our context. Since the VS method is written in a very general sense, a first action will be needed to adapt it to our specific application: for validating a designerly software development methodology that was tested in a single case study. Therefore, following changes were needed:

1. Originally, the VS was supposed as a validation method for design knowledge. However, we will use it to validate the DSDM which is a software development methodology.
2. Although several authors (Eisenhardt 1989; Lee and Rine 2004) recommend to use multiple cases in order to adopting several different viewpoints, we will in this research base on only one application case. The principle of extrapolation will be used to reason about other possible applications. The only condition is that this single case is representative which means that the case covers the complete methodology with all constructs. If so, other examples are redundant since they cannot fulfill a role in the validation process that is not fulfilled by the case.
3. As discussed above, a reflective approach must be applied. This means that the both the performance as well as the structural validation should be a done in a qualitative manner, based on the principle of reasoning with consequences.
4. The validation of the individual constructs was already discussed in their respective chapters. Repetition of this information is not relevant so referring to these parts will be used. Nevertheless, validation of each construct as part of the overall methodology is still needed.
5. To apply the validation square method, the most important performances indices should be identified to be investigated in this research, since the initial interpretation did not extend to possible criteria that could be used. Therefore we can base on the criteria that were identified in the literature study to identify the most appropriate ones
6. As shown in the center of Figure 3, we introduce a new figure that is schematically representing the relationship of the different quadrants of the validation square. We split the quadrants so they do not touch each other, this was necessary to show the relationships between the quadrants. Consequently we can point on the discussion between the domain-independent and the domain-specific performance validity, which is needed to conclude the validation.

4 OPERATIONALIZATION OF THE METHOD FOR OUR PARTICULAR CASE

To operationalize the quadrant-based external validation method (QEVM) the execution steps must be detailed for each quadrant. Figure 3 shows the overview of the complete validation process including the execution steps necessary in each quadrant. In the following subsections, each quadrant is explained in more detail, especially focusing on providing evidence on why the step is needed.

4.1 Clarification on testing the theoretical structural validity

The objective of this test is to do a domain-independent structural validation of both the overall method and the individual parent constructs. The validity can be measured using the information flow in the whole process of the methodology, as it is valid if the generation of all needed pieces of information is supported through the methodology (= necessary condition) and when the producing of the information happens when it is needed (= sufficiency condition). To achieve this, the requirements of the outcomes of the method and the process by which the method generated the outcomes should be known (STEP 1). High level requirements should be decomposed into a hierarchical set of more

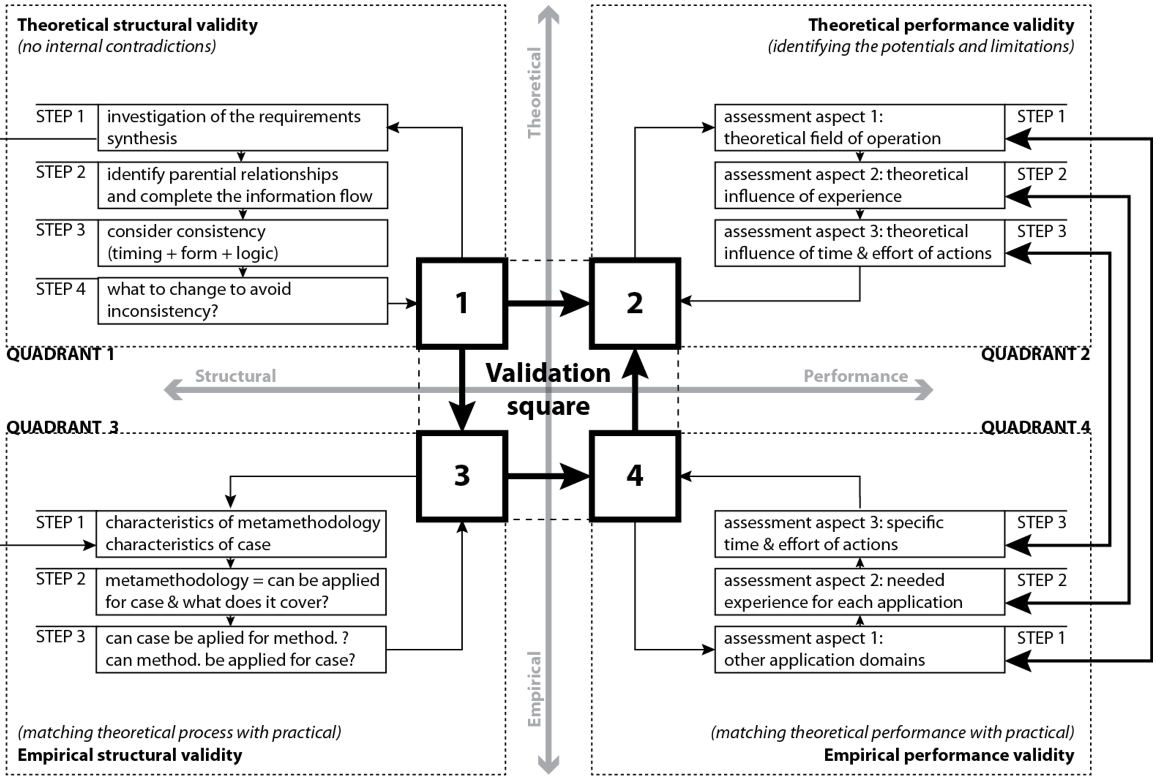

Figure 3: introducing the dedicated executions steps in the quadrant-based validation

specific requirements. Also the characteristics of the intended context for application of the method should be included and may include details of the intended physical domains, types of performance parameters, classes of variables, and product architectural characteristics. As the meta-methodology is based on different constructs it is also important to identify the parental relationships between the meta-methodology and its constructs and complete the information flow (STEP 2). This information should be collected in order to establish the internal consistency of the proposed design methodology by considering the timing, formulations and logic between the different constructs (STEP 3). Lastly, suggestions should be made on how inconsistency can be avoided (STEP 4).

4.2 Clarification on testing the theoretical performance validity

The objective of this quadrant is to validate the domain-independent performance of the methodology.
The theoretical performance is separated from the practical implementation, which can be used as testing means. To validate the theoretical performance, all targets that need to be achieved by the methodology, must be identified and performance aspects must be defined. To execute the validation in this context, three criteria of performance validation should be identified. The first criterion is the identification of the theoretical field of operation (STEP 1). Questions as “Can we find applications domains for methodological efficiency?” and “What are the potential domains of application?” should be answered, by reasoning, to identify the characteristics where it will and where it will not work properly. Finally, the boundaries should become clear by rational analysis and interpretative reasoning with consequences. The next criterion to discuss is the influence of experience on the performance (STEP 2). User’s experience might have a big influence on the performance of the methodology, so it is important to know what experiences (skills, competences, knowledge, …) are needed in general to use the methodology on an appropriate level. The third criterion focusses on the theoretical influence of time and effort of all actions (STEP 3), because the amount of time and effort should be known in advance to be able to balance it with the added value of the methodology.

4.3 Clarification on testing the empirical structural validity

The objective of this quadrant is to do a domain-specific structural validation. Practically, it involves building confidence in the appropriateness of the example problem. Consequently this means that the characteristics of the example problem must be mapped (STEP 1) to see how the methodology and the example case are covering each other (STEP 2). Consequently on the one hand it is important to show that the meta-methodology can be applied for the case and what aspects it covers (and which not). And on the other hand the characteristics of both the design problems for which the methodology is intended and those that are not covered must be identified. By (i) documenting that the data from the example can be used to support conclusions with respect to the performance of the design methods, (ii) documenting the example's simplified assumptions and (iii) mentioning that its data can be compared, contrasted, and processed to evaluate the performance of the proposed design method, the appropriateness of the example case should be shown (STEP 3).

4.4 Clarification on testing the empirical performance validity

In this quadrant the aim is to validate the domain-specific performance of the software development methodology. This should be done by checking how the targets are achieved and what the performance is of the methodology in reaching them. The same performance validation criteria, as in Section 4.2, will be considered but from the perspective of application examples. In order to enlarge our reasoning other application examples should be identified for which the methodology can be as efficient as in the source application or even more efficient (STEP 1). All application examples should be evaluated by following aspects: (i) resembling functionality, (ii) user requirements, (iii) necessary resources, (iv) level of sophistication: modeling, data, environment, (v) communication intensity, and (vi) level of standardization (reusability). Furthermore, the specific experiences needed (STEP 2) and the specific time and effort of all actions (STEP 3) of the identified possible application examples should be discussed. As in the other quadrant, the sample applications should be discussed by rational analysis and interpretative reasoning with consequences. In order to be able to get conclusion from the validation square, the theoretical and empirical performance validity should be compared for each of the above mentioned aspects.

5 EXECUTION OF THE ASSESSMENT

Regarding the empirical validation, we used a single reference application case in a deductive reasoning. Because we could conclude, based upon the results of the experiments, that the reference case was effectively developed using the DSDM, we accept its theory to be true and consequently we could claim comparable things for those products that are part of the same family, and have comparable characteristics.

5.1 Execution of the structural validation

The structural validation could be achieved by discussing the characteristics of the DSDM and its three constructs, on three levels: (i) universal requirements for DSDM, (ii) application independent requirements, and (iii) application specific requirements. The DSDM is focusing on three phases in the development process of software products, and for each phase a specific construct methodology was developed. As shown in Figure 4, the information of the developed software follows a logic path through the process of DSDM and its constructs. On an abstract level, we can say that the DSDM methodology has a linear structured process in which a phase must be finished before going to the next step. But the processes of the different constructs are both iterative and linear: depending on the complexity it is in some parts necessary to do more iteration before having a satisfied result. The consecutive logic of the different constructs in the different phases is supporting the constructive character of the methodology. In the information flow, four moments of data transformation can be identified: (1) data transformation that is needed as a preparation for the construct methodology, (2) data transformation that is done during the methodology execution, (3) Data transformation during the concluding phase of the methodology, and (4) Data transformation in between the different phases of the software development that must be done by the developers in order to be able to go to the next phase (The numbers are referred to in Figure 4). The empirical structural validation was done by matching the theoretical process with a practical application reference case of whom the major objective was to support designers in their decision making process on smart energy saving possibilities. More information on this reference software tool can be found in (Du Bois, Horvath et al. 2010).

5.2 Execution of the performance validation

During the performance validation, the potentials and limitations of the DSDM regarding the performance were identified. As shown in Figure 5, we discussed both the theoretical and the empirical performance of reaching all targets, according to: (i) the field of operation, (ii) the influence of experience on the performance, and (iii) the influence of time and effort of actions. In the theoretical discussion, we first identified the characteristics where the methodology will and where it will not work properly, and we defined the boundaries of the characteristics. Secondly, we identified the different skills and experiences that were needed to perform each action of the DSDM. And in the third assessment steps we focused on the theoretical needed time and efforts. The empirical performance validity was achieved by identifying the performance by reasoning on the reference case plus possible application cases: (i) a software for an alarm system, (ii) a company information system to manage production, (iii) a product-service system for furniture reuse, and (iv) an interactive video-wall to communicate about cultural events. In addition, comparison was done to compare the conclusion of the theoretical validation with the empirical validation.

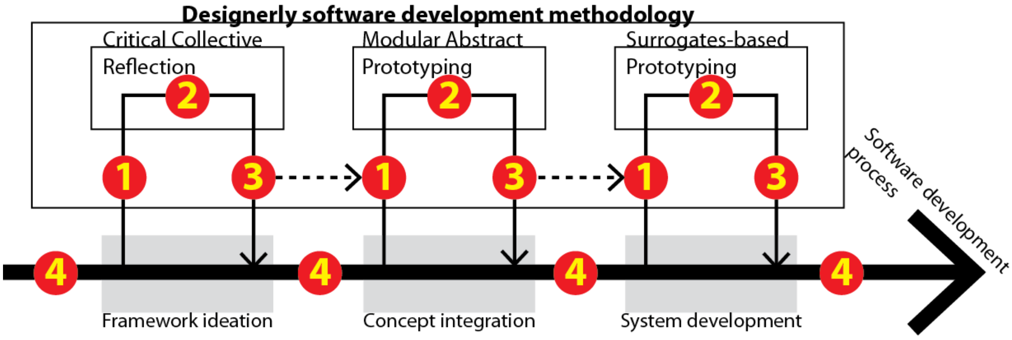

Figure 4. Logical flow of data through the DSDM

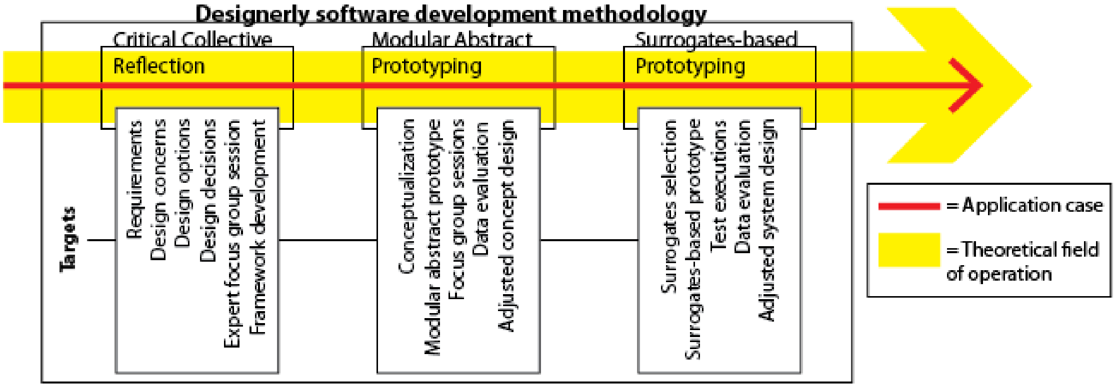

Figure 5. Performances of the DSDM

6 FINDINGS ON THE EXECUTION OF THE ASSESSMENT

In this Section we want to revisit the validation assessment of the designerly software development methodology by summarizing the conclusions of each quadrant after applying the quadrant-based external validation method (QEVM).

6.1 Findings on the first quadrant's validation assessment

In short, no critical internal contradictions were found in the DSDM and considering the use of the different constructs: (i) critical collective reflection, (ii) modular abstract prototyping, and (iii) surrogates-based prototyping. Therefore, we can evaluate the theoretical structural validity positively. To avoid inconsistency in the use of the DSDM, it is important: (i) to follow the logical order of the prescribed phased on the development process; (ii) to apply all sub-methodologies: in some cases it might be better to not execute one of the sub methodologies. However, if one construct is not used, essential information might be missing to go further with next construct. And (iii) to transform data properly or information might be lost or twisted.

6.2 Findings on the second quadrant's validation assessment

Typical application cases have a complex functionality, user requirements that are rather uncertain and unclear in the beginning of the process, and high level of sophistication due to environmental aspects, modeling need and data processing. Instead of just programming skills, the development team also needs research, reasoning, presentation, design computer, and graphic skills to use the DSDM. Different effort and time is needed depending on the specific action. However, the total time needed to execute the development using the DSDM is considered to be lower than other approaches, because less iterations are needed, and because a higher SH-adjustment is achieved.

6.3 Findings on the third quadrant's validation assessment

We found that the application fulfills the specific requirements of the DSDM and that DSDM was a relevant methodology for the development of the software case (Table 1). Also the comparison of the two logical processes of the theory of meta-methodology on the one hand and the process of the concrete case development on the other hand was positive. By comparing the different methodology actions and the concrete steps in the case development, we can make a link of what parts of the software case development are referring to a certain action of the meta-methodology.

Table 1. Methodology and case coverage

DSDM – relevance indicator	**Software case – fulfillment indicator**
• to support the software development process	• A software tool for smart energy saving
• deal with uncertain and unclear user requirements in the beginning of the process	• For the software case, it is difficult to formulate the complete requirements in advance, due to many reasons.
• was developed to deal with complex functionalities	• It is a complex software product (based on engineering principles) • The complexity will grow if a community can grow next to it
• The DSDM aims to co-design with stakeholders	• The stakeholder request high level of involvement.
• support as a communicating means • Project documentation is achieved through the prototypes	• Most importantly is that different stakeholders are involved in the process and in the use of the product. But also high level of modeling is needed able to discuss the design.
• focusing on relative complex projects • multi-abstraction levels	• combining energy saving in household appliances • knowledge base with multiple data types, in a complex integrated model.
• DSDM is especially for multi-disciplinary teams	• Low amount of time, small budget, single person-team, few programming skills, high development skills are also some important characteristics of the software case

6.4 Findings on the fourth quadrant's validation assessment

For each of the criteria we can conclude that the identified application cases match the theoretical application field. To consider the theoretical field of operation, all application examples were evaluated by following aspects: (i) resembling functionality, (ii) user requirements, (iii) necessary resources, (iv) level of sophistication: modeling, data, environment, (v) communication intensity, and (vi) level of standardization (reusability). Next, considering the theoretical influence of experience, we can conclude that although the amount of each skill needed is also depending on the context of the application, also the methodology requires specific skills and experiences. Lastly, the time and effort needed for each action in the development of the identified application cases matches in general the theoretical one. Differences are mainly due to different levels of complexity caused by the amount of stakeholders and functionalities.

7 CONCLUDING PROPOSITIONS AND FUTURE RESEARCH

Proposition 1: The DSDM is a valid methodology for the development of software products that have (i) complex functionality, (ii) user requirements that are rather uncertain and unclear in the beginning of the process, (iii) high level of sophistication due to environmental aspects, modeling needed and data processing, and (iv) request a high level of stakeholder involvement in their development process.
Proposition 2: The quadrant based validation is a valuable method for the validation of this designerly software development methodology.
Proposition 3: The quadrant-based external validation method combines structural and performance assessment actions in both the theoretical domain and the application domain.
Proposition 4: In each quadrant, the different steps allow both qualitative and quantitative assessment according to various criteria in a reflexive manner, starting out from the main structural and performance characteristics.

Proposition 5: The proposed approach has a large application potential and is flexible enough in single-case, reflexive, context dependent assessments. Further research is needed to explore whether this quadrant-based external validation method can be applied in a context independent manner.

REFERENCES

Bednar, J. A. and D. Robertson (2007). Verification and validation, SAPM Spring.

Briand, L., K. El Emam, et al. (1995). Theoretical and Empirical Validation of Software Product Measures. ISERN-95-03, international software engineering research network 23.

Campbell, D. T. and J. C. Stanley (1963). Experimental and quasi-experimental designs for research. Chicago, Rand McNally.

Carson, J. S. (2002). model verification and validation. 34th conference on Winter simulation: exploring new frontiers. San Diego, California, Winter Simulation Conference**:** 52-58.

Dellinger, A. B. and N. L. Leech (2007). "Toward a Unified Validation Framework in Mixed Methods Research." Journal of Mixed Methods Research **1**(4): 309-332.

Du Bois, E. and I. Horvath (2011). An easy-to-use methodological approach for considering ubiquitous controllers in energy use optimization. EcoDesign. Kyoto, Japan.

Du Bois, E., I. Horvath, et al. (2010). Critical review of smart energy saving in household electronics. TMCE 2010, Ancona, Italy, Delft University of Technology.

Eisenhardt, K. M. (1989). "Building Theories from Case Study Research." The Academy of Management Review **14**(4): 532-550.

Eisenhart, M. A. and K. R. Howe (1992). Validity in educational research. The handbook of qualitative research in education M. D. LeCompte, W. L. Millroy and J. Preissle. San Diego, CA, Academic Press**:** pp. 643-680.

Eisner, E. W. (1991). The enlightened eye: Qualitative inquiry and the enhancement of educational practice. Upper Saddle River, NJ, Prentice Hall.

Frey, D. and C. Dym (2006). "Validation of design methods: lessons from medicine." Research in Engineering Design **17**(1): 45-57.

Gottschalk, P. (2002). "Empirical Validation Procedure for the Knowledge Management Technology Stage Model." Informing Science The International Journal of an Emerging Transdiscipline **5**(4): 30.

Hickey, A. and D. Dean (1998). Prototyping for Requirements Elicitation and Validation: A Participative Prototype Evaluation Methodology. AMCIS 1998.

Jagdev, H. S., J. Browne, et al. (1995). "Verification and validation issues in manufacturing models." Computers in industry **25**(3): 331-353.

Jensen, S. Ø. (1995). "Validation of building energy simulation programs: a methodology." Energy and Buildings **22**(2): 133-144.

Kitchenham, B., S. Linkman, et al. (1997). "DESMET: a methodology for evaluating software engineering methods and tools." Computing & Control Engineering Journal **8**(3): 120-126.

Kleijnen, J. P. C. (1995). "Verification and validation of simulation models." European Journal of Operational Research **82**(1): 145-162.

Landry, M., J.-L. Malouin, et al. (1983). "Model Validation in Operations Research." European Journal of Operational Research **14**.

Lee, S. W. and D. C. Rine (2004). Case Study Methodology Designed Research in Software Engineering Methodology Validation. Sixteenth International Conference on Software Engineering and Knowledge Engineering (SEKE'04),, Banff Centre, Banff, Alberta, Canada,.

Lincoln, Y. S. and E. G. Guba (1985). Naturalistic inquiry. Beverly Hills, CA, Sage.

Macal, C. M. (2005). Model verification and validation. w. i. t. a. s. s. m. a. models". Chicago, IL, center for complex adaptive agent system simulation.

Maxwell, J. A. (1992). "Understanding and validity in qualitative research." Harvard Educational Review **62**: 279-300.

O'Keefe, R. and D. O'Leary (1993). "Expert system verification and validation: a survey and tutorial." Artificial Intelligence Review **7**(1): 3-42.

Olewnik, A. T. and K. E. Lewis (2003). On validating design decision methodologies. ASME 2003 Design Engineering Technical Conferences - Design Theory and Methodology Conference, Chicago, IL., ASME.

Pedersen, K., J. Emblemsvåg, et al. (2000). Validating design methods & research: the validation square. ASME Design Engineering Technical Conferences, Baltimore, Maryland, ASME.

Sargent, R. G. (1984). A tutorial on verification and validation of simulation models. winter simulation conference.
Sargent, R. G. (2005). Verification and validation of simulation models. 37th Winter Conference on Simulation. Orlando, Florida**:** 130-143.
Schön, D. and C. Argyris (1975). Theory in practice: increasing professional effectiveness. Jossey-Bass, San Fransisco.
Seepersad, C. C., K. Pedersen, et al. (2006). The Validation Square: How Does One Verify and Validate a Design Method? Decision Making in Engineering Design. Decision Making in Engineering Design. K. E. Lewis, W. Chen and L. C. Schmidt, ASME Press**:** 303.
Simon, H. A. (1990). "Invariants of human behavior." Annual Review of Psychology **41**: 1-19.
Wallace, D. R. and R. U. Fujii (1989). "Software verification and validation: an overview." Software, IEEE **6**(3): 10-17.