

Validating the Design Theory for Managing Project Scope during Software Sourcing and Delivery

Journal:	<i>International Conference on Information Systems 2015</i>
Manuscript ID	ICIS-0712-2015.R1
Track:	10. IS Design and Business Process Management
Keywords:	Design/design science, IS outsourcing, Software project management, Software process improvement
Abstract:	<p>Most software project estimation and measurement (PEM) processes have been designed for providers. Customers need to leverage PEM to better direct software sourcing. A design theory for software project-scoping has been developed that supports the PEM processes of both customers and providers. This paper probes the validity of the theory from providers' and customers' viewpoints through three longitudinal case studies. A software provider used a preliminary version of the theory in dozens of software development projects annually, systematically (1) achieving higher customer satisfaction through better estimation accuracy and (2) improving productivity. Two government organizations used the theory to speed up their software sourcing and reduce their sourcing costs to a half without giving up any functional or quality requirements. Similar results have not been obtained before. Future research is needed to generalize the findings.</p>

Validating the Design Theory for Managing Project Scope during Software Sourcing and Delivery

Completed Research Paper

Timo Käkölä

University of Jyväskylä
40014 University of Jyväskylä
Finland
timokk@jyu.fi

Pekka Forselius

4SUM Partners Ltd.
Tekniikantie 14, 02150 Espoo
Finland
pekka.forselius@4sumpartners.com

Abstract

Most software project estimation and measurement (PEM) processes have been designed for providers. Customers need to leverage PEM to better direct software sourcing. A design theory for software project-scoping has been developed that supports the PEM processes of both customers and providers. This paper probes the validity of the theory from providers' and customers' viewpoints through three longitudinal case studies. A software provider used a preliminary version of the theory in dozens of software development projects annually, systematically (1) achieving higher customer satisfaction through better estimation accuracy and (2) improving productivity. Two government organizations used the theory to speed up their software sourcing and reduce their sourcing costs to a half without giving up any functional or quality requirements. Similar results have not been obtained before. Future research is needed to generalize the findings.

Keywords: Design theory, Information Systems Sourcing, Software Project Scope Management

Introduction

Software project estimation and measurement (PEM) has been researched extensively due to negative organizational and financial consequences of projects that run late and out of budget and fail to deliver desired functionalities and to meet nonfunctional quality requirements. PEM aims at predicting the productivity, total effort and/or cost, and schedule of a project any time and as often as necessary before the end of the project. PEM facilitates project-planning, tracking, control, pricing, and contracting, which are crucial stepping-stones for companies aiming at managing their processes quantitatively at the highest capability and maturity levels (Chrissis et al., 2011; ISO, 2004). It is critical in complex business networks where customers acquire and integrate software from providers through sourcing, a business practice of looking for domestic or foreign providers capable of performing or subcontracting services (Hefley and Loesche, 2010; Hyder et al., 2010). PEM is deployed throughout the sourcing life-cycle from strategic sourcing analysis (where customers plan and prioritize the available sourcing options) through tendering and contracting to service delivery and completion (Käkölä, 2008). Customers use PEM for scope management to keep their expenditures in line with expected sourcing outcomes. Providers use PEM to avoid serious under or over estimates in bids, contracts, and other sales activities.

Most PEM processes and methods have been designed for software providers (Bundschuh and Dekkers, 2008). Functional size measurement is one of the most important processes. It measures the pieces of

software to be produced in function points, expressing the amount of business functionality an information system provides to users, independent of the technology used to implement the information system (ISO, 2010; ISO, 2012). Bundschuh and Dekkers (2008) cover the available processes and methods most comprehensively to date but only from the viewpoint of software providers.

An overly emphasis on software provisioning limits the benefits to be obtained from project estimation and measurement because customers should take important decisions in early phases of the sourcing life-cycle when providers are not necessarily involved yet. If customers initiate software sourcing based on so poorly defined requirements that PEM cannot be used reliably to estimate work efforts, budgets, and schedules based on the requirements, projects are likely to be misdirected from the very beginning. Moreover, software sourcing is typically based on three types of contracts. In *fixed-price contracts*, providers agree to provide software for a fixed price determined based on the requirements set in the initial phases of the life-cycle even if the requirements may significantly change later on. In *time-and-materials contracts*, customers agree to pay providers based on the work performed and materials used no matter how much work is necessary to complete the software. Hybrids of fixed-price and time-and-materials contracts represent the third type. They tend to allocate gains and risks unevenly and typically lead to win-lose (or lose-lose) relationships, potentially reducing the effectiveness of software sourcing and the quality of software deliverables.

Both customers and providers benefit from leveraging PEM processes and methods to establish contractual schemes based on a functional size based *unit pricing model* (Rollo and Wright, 2001; Victorian Government, 2001; Wright, 2000). According to the model, customers pay providers an agreed upon sum per function point (fp) delivered for the functionality they need. Because both parties can win, if the price is realistically and competitively set, the model can facilitate trusting relationships between customers and providers and help customers to better direct software sourcing projects and providers (Dekkers and Forselius, 2010; Käkölä, 2008; Käkölä and Lu, 2015).

There is limited theoretical knowledge available to help both customers and providers to benefit from software sourcing through PEM processes and methods and functional size based unit pricing (Käkölä, 2008; Käkölä and Lu, 2015). Theoretical knowledge is needed to help (1) customers to reach the goals of obtaining sourced systems faster and reducing software sourcing costs significantly without giving up any functional or quality requirements and (2) providers to reach the goals of having satisfied customers and delivering software development projects profitably and systematically within the budget. Achieving these goals is problematic in practice. Therefore, the research reported in this paper is practice-inspired.

Design theories, unlike other theories, support the achievement of goals (Gregor and Jones, 2007; Hevner et al., 2004; Markus et al., 2002; Simon, 1996; Van Aken, 2004; Walls et al., 1992, 2004). Walls et al. (1992, p. 37) argue that the information systems research “field has matured to the point where there is a need for theory development based on paradigms endogenous to the field itself” and call for design theories to fulfill that need. A design theory is ‘a prescriptive theory based on theoretical underpinnings which says how a design process can be carried out in a way which is both effective and feasible’ (ibid, p. 37). It prescribes both the design product and process aspects of a class of artifacts, that is, *what* are (1) the design value propositions to be fulfilled by implementing an instance of the class, (2) meta-requirements describing the problem(s) to be solved by the class, (3) the meta-design prescribing the class of artifacts used to solve the problem(s), and (4) applicable kernel theories from social and natural sciences for understanding, governing, and/or solving the problem(s) shared across all products within the class, and *how* the products should be built (Walls et al., 1992, 2004).

A design theory for software project scoping during software sourcing and delivery (hereafter, the theory) has been developed and validated over the period of many years in accordance with the action-design research (ADR) method (Sein et al., 2011). Improved versions of the theory have been iteratively designed based on the evolution of relevant academic body of knowledge, including several kernel theories; the building and application of artifacts ingraining the theory (hereafter, theory-ingrained artifacts or artifacts) in various organizations; and the investigation of emergent interactions between the versions and the organizations applying them to uncover unforeseen redesigns. The theory has become stable by the time of writing this paper. It covers and combines estimation and measurement process areas. A commercial software product (hereafter, the software product) and the associated repository of data collected from more than a thousand software projects (hereafter, the repository) are theory-ingrained

artifacts supporting it. The theory benefits both providers and customers and relies on functional size based unit pricing.

There are multiple views of theory in information systems research. Gregor (2006) identifies five interrelated types of theory: (1) theories for analyzing, (2) theories for explaining, (3) theories for predicting, (4) theories for explaining and predicting, and (5) theories for design and action. Weber (2012) establishes strict criteria for theories and ascribes the term “theory” only to theories for explaining and predicting. The theory in this paper is a type 5 theory for design and action (Gregor, 2006), making the design and implementation process more tractable for providers and customers by scoping the range of effective process steps to a manageable set (c.f., Markus, 2004). The design theory incorporates the work system for software project scoping (Table 1), including the generic work role of a scope manager, the roles of customers and providers, and the activities these roles are responsible for (c.f., Alter, 2008). The theory prescribes the minimal structural and process characteristics of software project scope management that providers and customers should implement throughout the sourcing life-cycle. Table 1 and Figure 1 prescribe the core artifacts of the meta-design of the theory.

This paper probes the validity of the meta-design of the theory through the ADR method and three case studies. The ADR method was chosen because this research is practice-inspired and deals with the building and evaluation of theory-ingrained artifacts. Validation was carried out by following natural controls in case organizations because the artifacts were emergent and it was deemed more important to achieve maximum authenticity than controlled evaluation (Lee, 1989). The paper deals with two research questions. (1) Is the theory valid from customers’ viewpoint? (2) Is the theory valid from providers’ viewpoint? These questions are answered with respect to the first six design value propositions (DPs) of the theory:

DP 1: The meta-design of the theory helps customers to obtain the sourced systems faster.

DP 2: The meta-design of the theory helps customers to reduce their software sourcing costs significantly.

DP 3: The meta-design of the theory helps customers to realize DP 1 and DP 2 without giving up any functional or quality requirements.

DP 4: The meta-design of the theory helps providers to raise their productivity significantly.

DP 5: The meta-design of the theory helps providers to deliver systematically software that meets customer requirements on time.

DP 6: The meta-design of the theory helps providers to deliver systematically software that meets customer requirements on budget.

These propositions deal with summative evaluation, that is, they are outcome focused. Formative evaluation (Remenyi and Sherwood-Smith, 1999), contributing to the refinement of the artifacts and the uncovering of anticipated and unanticipated consequences of using instances of the artifacts, and a detailed coverage of the iterative and emergent design and validation of the theory in the three case organizations are beyond the scope of this paper due to space limitations.

DP 7 provides prerequisites that should be fulfilled before the six propositions can be realized:

DP 7: The meta-design of the theory helps customers to realize DP 1, DP 2, and DP 3 and providers to realize DP 4, DP 5, and DP 6 only if the customers can specify their needs and the providers can implement software.

The case organizations were chosen for this research partly because they fulfilled the prerequisites.

The next section probes the meta-design of the theory (hereafter, the meta-design) and the underlying kernel theories. The section “Validation of the Design Theory” addresses the research questions by studying two customers and one provider. The final section discusses conclusions and issues for future research.

Fundamentals of the Meta-Design of the Design Theory

The scoping process of the meta-design consists of twelve steps, covering the sourcing life-cycle from feasibility studying and initial requirements analysis through requesting tenders, contracting, estimating

work effort, managing change, and controlling progress to development program closure (Table 1). It is novel. Our extensive literature review revealed that southernSCOPE (Rollo and Wright, 2001; Wright, 2000) is the only similar approach. SouthernSCOPE has eight steps that roughly correspond to steps 1, 3, 5, 6, 7, 8, 9, and 11 of the scoping process (Table 1). It is thus not as comprehensive as the scoping process. Victorian Government of Australia (hereafter, Victorian Government) has developed and used southernSCOPE in a number of projects to facilitate software sourcing. Victorian Government (2001, p. 2) states that all projects deploying southernSCOPE delivered software that met customer needs, resulted in high levels of customer satisfaction, reduced the average budget over-run to less than 10 per cent, and provided “software value-for-money within the top 25 per cent of industry best practice.” However, southernSCOPE has never been validated scientifically or used by other customers. Moreover, the Victorian southernSCOPE user society, consisting of a group of experts that provide scope management services to Victorian Government, has not explicitly designed software project estimation and measurement systems (PEMS) to support southernSCOPE, making it cumbersome to use in practice. In contrast, the research reported in this paper shows that two customers and one provider have leveraged the meta-design to source or deliver software products and services consistently in time and in budget with the agreed upon functionality over the period of several years. To our knowledge, similar research results have not been obtained before.

The software project effort estimation process of the meta-design (Figure 1) consists of functional size measurement (FSM), situation analysis (e.g., (FiSMA, 2001)), reuse analysis (FiSMA, 2002), and delivery rate determination. Bundschuh and Dekkers (2008) assess FSM methods in a reasonably unbiased way and find that ISO/IEC 29881:2010, FiSMA 1.1 Functional Size Measurement method, represents the most comprehensive, widely applicable, easy-to-use, and accurate internationally standardized FSM method. The delivery rate is assessed in terms of the average number of development hours required in similar past development projects to deliver a function point. Situation analysis makes the estimates more precise by analyzing 21 factors that affect the development productivity. The factors are organized into 5 Project factors, 5 Process factors, 6 Product factors, and 5 People factors. Situation analysis draws upon COCOMO (Boehm, 1981; Boehm et al., 2000) as the kernel theory. Reuse analysis further perfects the estimates by determining the impact of reuse on productivity.

Step	Concept	Participants
1	Scope manager retained, customer-driven high-level requirements.	Customer + SCOPE Manager
2	Divide program into subprojects.	SCOPE Manager + Customer
3	Scope manager does early FP for each subproject and estimates total size.	SCOPE Manager
4	Scope manager and customer determine and analyze quality requirements.	SCOPE Manager + Customer
5	Customer issues request for proposal.	Customer + SCOPE Manager input
6	Customer selects supplier based on submitted unit cost per FP.	Customer + SCOPE Manager input + supplier
7	Requirements specification developed.	Supplier + Customer
8	Scope manager baselines FP size and product development.	SCOPE Manager
9	Project changes sized, cost impact evaluated (based on same cost/FP).	SCOPE Manager + supplier input
10	Scope manager quantifies progress.	SCOPE Manager
11	Project finishes and customer pays supplier based on FP delivered.	Customer + supplier
12	Experience data collected and stored.	SCOPE Manager

Table 1. The 12 Steps of the Scoping Process of the Meta-Design (Dekkers and Forselius, 2010, p. 17)

Software project estimation and measurement involves a set of knowledge-intensive business processes and is critically dependent on effective PEMS support (Forselius and Käkölä, 2009). Internal project repositories are needed to store, retrieve, and manage project history data necessary for calculating the internal delivery rates. Best-in-class providers with ample projects can benefit greatly from internal project repositories to benchmark their current projects with similar historical ones. Organizations can use national and international software project estimation and measurement systems and repositories (Bundschuh and Dekkers, 2008, 289-317; He et al., 2008; Hill, 2010; ISO, 2011; Maxwell and Forselius, 2000; Premraj et al., 2005) cost-effectively to benchmark the internal delivery rates with those of best-in-class providers and see, for example, whether the development can be done most cost effectively in-house or by domestic or international providers (Käkölä, 2008).

Experience Factory (Basili et al., 2002) is the kernel theory underpinning the software project estimation and measurement processes of the meta-design. Experience Factory is an organization that receives product development and environmental information from the development projects, processes this information, provides direct feedback and appropriate models and tools to each project based on previous projects, and produces, stores and provides baselines, tools, and lessons learned from a more generalized perspective.

During the early development of the meta-design, the project managers of providers were considered the most critical stakeholders. However, industrial ADR studies in several case organizations indicated that the project managers could not deploy the meta-design as effectively as had been anticipated. First, customers have to have a central role in scope management (Table 1; Forselius et al., 2008). For example, scope management is crucial to help customers to avoid too large and poorly defined requests for tender. Second, the theory should be deployed by a neutral party because it aims at facilitating win-win relationships between customers and providers. Project managers are never neutral. Third, project managers have to deal with ten different knowledge areas, of which scope management is only one (PMI, 2013). Scope management is a broad area in itself, covering not only project management tasks but also requirements engineering and effort estimation. Project managers cannot always be expected to master and to execute the meta-design in addition to dealing with their other responsibilities. Fourth, PEMS are typically comprehensive systems requiring ample time and effort to learn. Yet, project managers only need to use PEMS periodically in projects. Routines and skills for using PEMS may thus take a long time to develop for project managers.

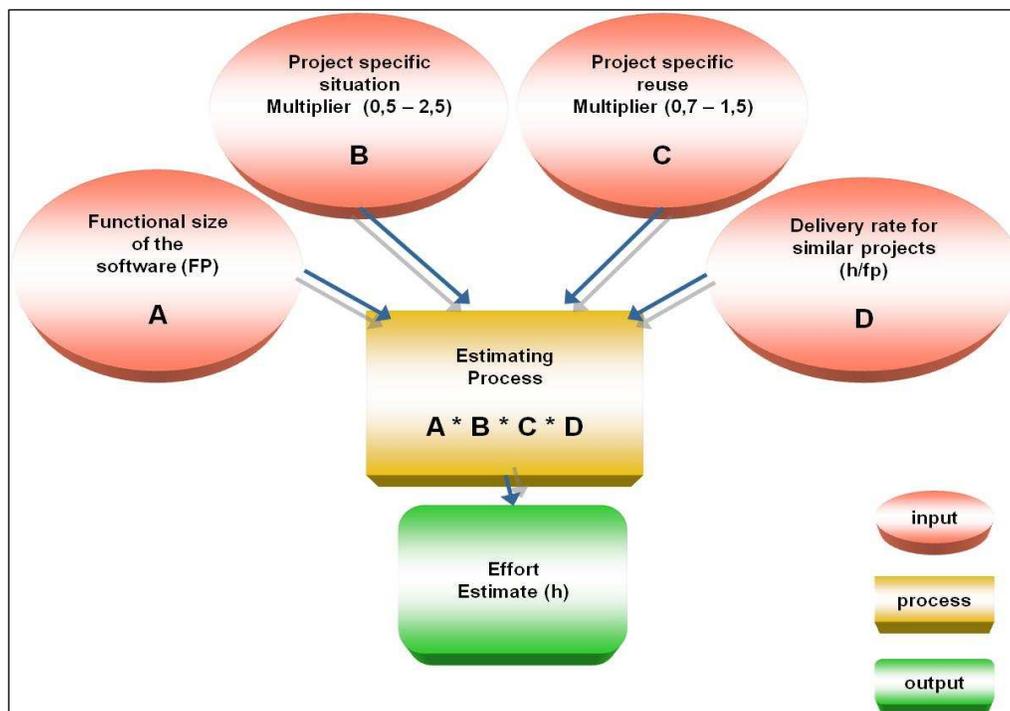


Figure 1. Software Project Effort Estimation Process of the Meta-Design

A job role of a professional scope manager (ECQA, 2015; Forselius et al., 2008; Morris, 2004; Wright, 2000) has been incorporated in the meta-design as a result of the above-mentioned emergent interactions between the theory and the organizations applying it (c.f., Sein et al., 2011). Wright (2000) defined the job role as follows: “The project engages a ‘Scope Manager’ who is a specialist in software measurement. This person provides early and realistic estimates of size, cost and duration to the sponsor and throughout the project assists both the customer and the developer to interpret changes to scope.” Scope managers are independent from provider and customer organisations, assist and advise the sourcing management teams in scope management, and use PEMS throughout the sourcing life-cycle. They are involved in ten of the 12 steps of the scoping process (c.f., Table 1).

Validation of the Design Theory

This section presents longitudinal qualitative studies (Eisenhardt, 1989) from one provider and two customer organizations to provide a holistic, systemic understanding of the phenomenon of software project scoping. It analyzes whether the organizations were able to realize the design value propositions of the theory by following the meta-design of theory. First, the use of the meta-design by a provider called TKP Tieto is analyzed between 1990 and 2004. Next, the use of the meta-design by Ministry of Justice, Finland and Ministry of Social Affairs and Health, Finland is assessed from, respectively, 2006 to 2010 and 2008 to 2012.

The Use of the Theory-Ingained Artifacts to Support Providers: Case TKP Tieto

Motivation of Finnish providers to collect measurement data independently varied significantly in 1990s. Most providers only wanted to estimate work efforts in the beginning of projects using functional sizing methods based on initial requirements specifications and delivery rates from other providers. Customers neither required careful work effort estimation nor set productivity requirements for providers. There was relatively little competition, so elaborate work effort estimation was seldom warranted. Executives of these providers thus placed limited if any emphasis on measuring productivity and establishing long term improvement programs through systematic collection and analysis of project data.

The best providers were able to institutionalize systematic data collection for at least some years. The best example is TKP Tieto delivering pension insurance information systems. The longitudinal study in TKP Tieto was initiated in 1990. TKP Tieto was partially owned by its customers and it worked with its customers as a long term partner. Changes of the key features of the public pension insurance policies and the effective dates of the changed policies were decided by the Parliament of Finland. The effective dates set hard deadlines for completing respective changes in the insurance information systems. TKP Tieto collected project data sporadically from 1990 to 1998 (Figure 2). In 1998 its executives noticed through customer satisfaction surveys that customer satisfaction objectives were not reached because projects often exceeded budgets and overrun schedules. The hard deadlines set by the Parliament had to be met to satisfy customers. All project managers were mandated to use functional size measurement and determine delivery rates for work effort estimation in the beginning of projects and to measure and store the realized functional size, costs, and resource usage in the end of projects.

One employee with extensive project management and system development expertise was assigned at a time to coordinate scope management activities and to support projects within TKP Tieto in project planning, functional size measurement, effort estimation, and project data collection and analysis. A total of three employees coordinated scope management in TKP Tieto sequentially between 1995 and 2010. One of the authors of this paper (hereafter, the co-author) trained them and the first group of project managers in software project estimation and measurement. Later the coordinators carried the necessary tool and method training inhouse. All coordinators and project managers used the software product supporting the theory. Coordinators also monitored data quality. Coordinators did tasks belonging to scope managers. However, the scope manager role is formally defined and scope managers typically work and charge the costs of two full working days per month for a project. Coordinators had informal roles and worked on projects primarily when the project managers requested it. They did not necessarily work with all projects and could never charge their coordination-related working time to projects. Therefore, they cannot be called scope managers.

Data from appr. 30 projects was collected annually from 1999 to 2004 using the software product. All projects collected data relevant to the process shown in Figure 1. Software size and development effort related information and the values of all 21 productivity factors were recorded carefully, but data about the project specific reuse was collected less systematically. TKP Tieto was aware of the generic potential of software reuse to affect development productivity but it considered software reuse to have limited productivity impact on its business.

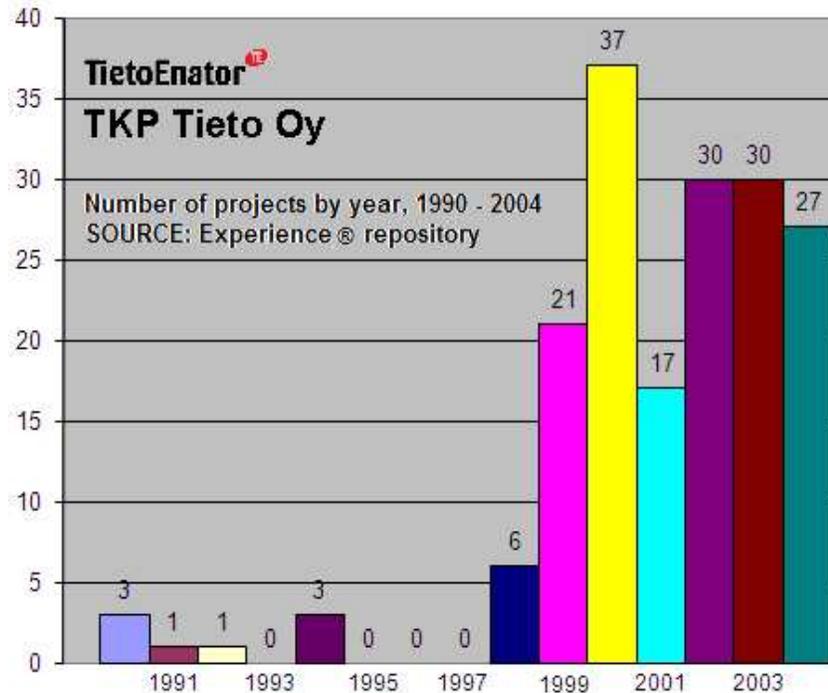


Figure 2. The number of projects of TKP Tieto delivering project data to the repository between 1990 and 2004. The colors of bars have no meaning.

Figure 3 shows the customer satisfaction of TKP Tieto biannually between 1999 and 2001. Feedback was collected regularly from customers of all projects. TKP Tieto performed and analyzed these surveys. The effectiveness of service and the overall customer satisfaction indexes increased systematically over time, indicating that TKP Tieto could meet customer requirements better and better. The most significant improvements took place with respect to the perceived reliability of budget and schedule estimates. The management was impressed about the improvements and continued project data collection and analysis to reach further improvements in subsequent years.

Figure 4 shows the accuracy of work effort estimates for 102 projects of TKP Tieto in 2000 and 2001. The estimates were deployed in budget planning of all the projects. Only one of the projects (in March 2000, the fifth bar from the left) exceeded its work effort estimate by more than 10 per cent. Others stayed very close to the original estimates and most got done with less effort than planned. The authors have never seen similar level of accuracy of effort estimates in the literature and in practice.

In sum, Figure 3 shows TKP Tieto could increasingly deliver software that met customer requirements on time and on budget. Figure 4 shows this success could be attributed at least to a significant extent to highly accurate work effort estimates that resulted from the comprehensive use of the meta-design. Therefore, TKP Tieto could use the meta-design to deliver systematically software that met customer requirements on time and on budget, realizing design value propositions 5 and 6 of the theory.

Figure 5 shows the productivity of system development work improved systematically from 1999 to 2001. In the beginning of 1999, the average work effort of all projects in TKP Tieto was almost 8 hours per function point. In 2001, the average work effort had been lowered to slightly more than 4 hours per function point. The delivery rate of less than 5 hours per function point was outstanding in the beginning of the millennium and it is rarely achieved even nowadays in the insurance industry. Therefore, TKP Tieto

realized the fourth design value proposition, that is, the leveraged parts (e.g., systematic project estimation and measurement processes) of the meta-design of the theory helped TKP Tieto to raise its productivity significantly.

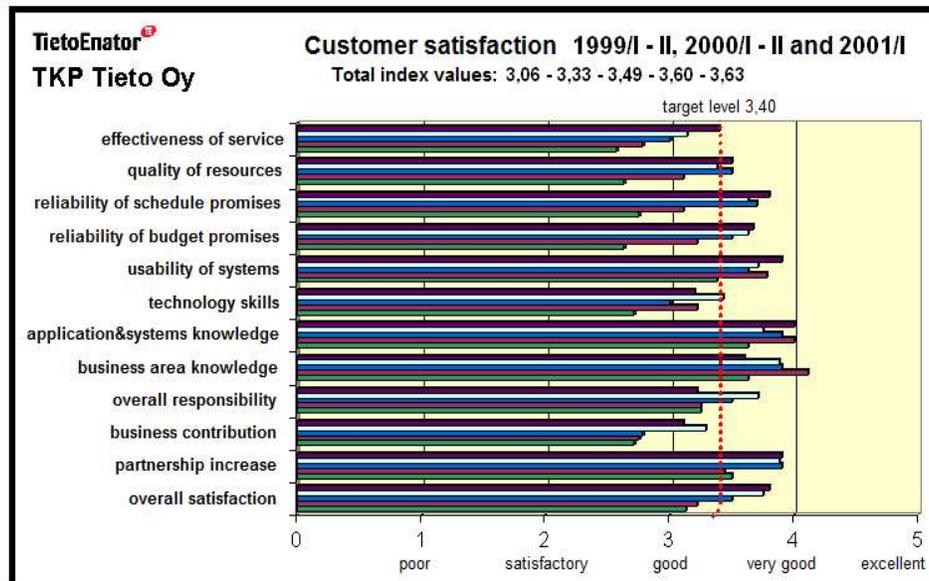


Figure 3. Index values of five biannual customer satisfaction summaries between 1999 and 2001. For each factor of customer satisfaction, the five bars from bottom to top represent, respectively, the averages of survey responses in chronological order from 1999/I to 2001/I. (Presentation material of TKP Tieto used with permission)

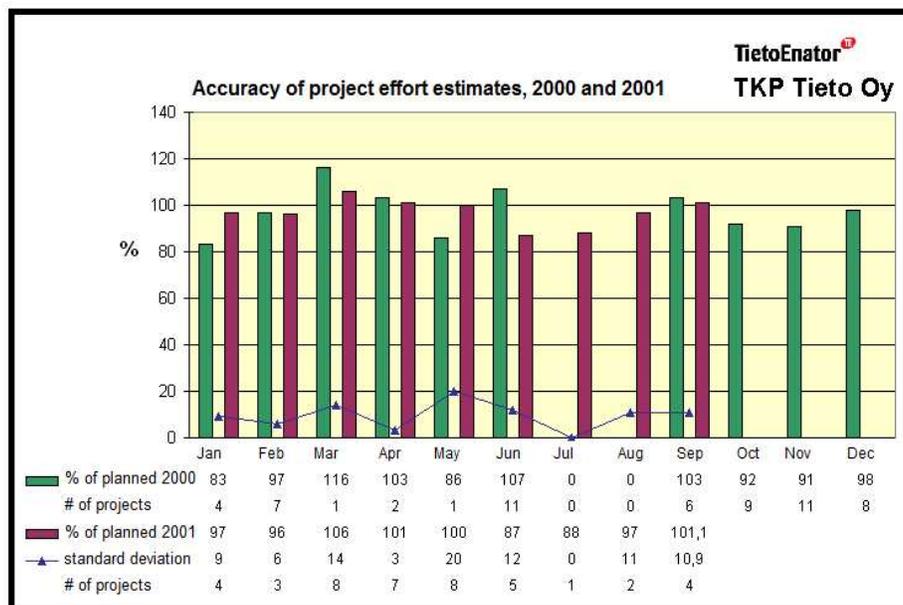


Figure 4. The accuracy of work effort estimates in TKP Tieto in 2000 and 2001. (Presentation material of TKP Tieto used with permission)

In sum, the analysis of Figures 2, 3, 4 and 5 indicates that TKP Tieto realized all three design value propositions for providers and achieved better estimation accuracy, higher customer satisfaction, and improved productivity after instituting major parts of the meta-design, including functional size measurement, systematic project data collection and analysis, and the role of a scope coordinator. During the study period, customers of TKP Tieto did not use external scope managers and the functional size

based unit pricing model in their sourcing engagements. The customers owned a majority share of TKP Tieto and perceived they could manage project scope adequately with their trusted partner without using these theory-ingrained artifacts. As a result, the artifacts remained beyond the scope of this validation study.

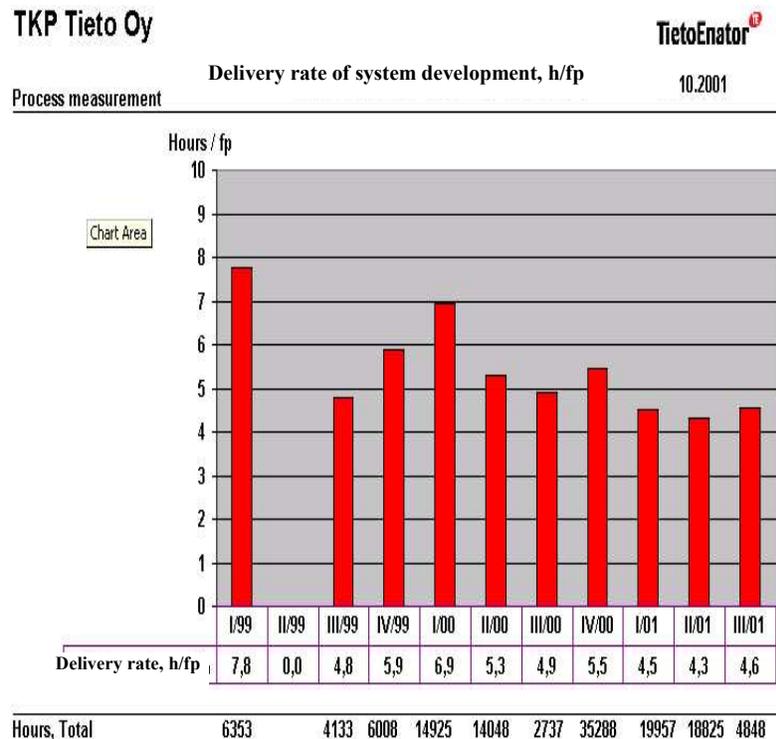


Figure 5. The delivery rate of system development in TKP Tieto from the beginning of 1999 to the end of the third quarter of 2001. (Presentation material of TKP Tieto used with permission)

The Use of the Theory-Ingrained Artifacts to Support Customers: Cases of Ministry of Justice, Finland and Ministry of Social Affairs and Health, Finland

For customers, the most essential measures of success are the total business value obtained from the sourced system and the user satisfaction perceived by the users of the system. However, the theory does not directly contribute to achieving these measures of success. The unit price of the system (e.g., €/fp) and the speed of system delivery (e.g., fp/month) are other measures of success for customers. The theory, in accordance with its design value propositions 1 and 2, directly contributes to success with respect to the last two measures. This section will show evidence for this claim.

Figure 6 draws upon a sample of all 107 public administration projects from the repository. Projects in this kind of sample (Hill, 2010) represent probably the best quartile of public administration software projects: they realized software functionality, measured the functionality in function points, and reported the duration and the price. Many of the projects in the other quartiles delivered documents but never realized a single piece of software. Such failed projects are not stored in the repository because they have not delivered any measurable results in terms of unit price, speed of delivery, and other measures. Figure 6 shows that most of the projects could deliver less than 50 function points monthly, having a relatively poor speed of delivery. Most of these slow projects were also pricey, costing more than 500 €/fp.

Six projects in the sample from Ministry of Justice and Ministry of Social Affairs and Health have applied the theory-ingrained artifacts. The co-author served as an external scope manager in the six projects completed between 2006 and 2012. They are numbered from 1 to 6 in Figure 6. The six projects are among the best in terms of the unit price, that is, far less than 500 €/function point. All six projects

delivered faster than the average speed of delivery and four of the projects were among the five projects with the fastest speed of delivery.

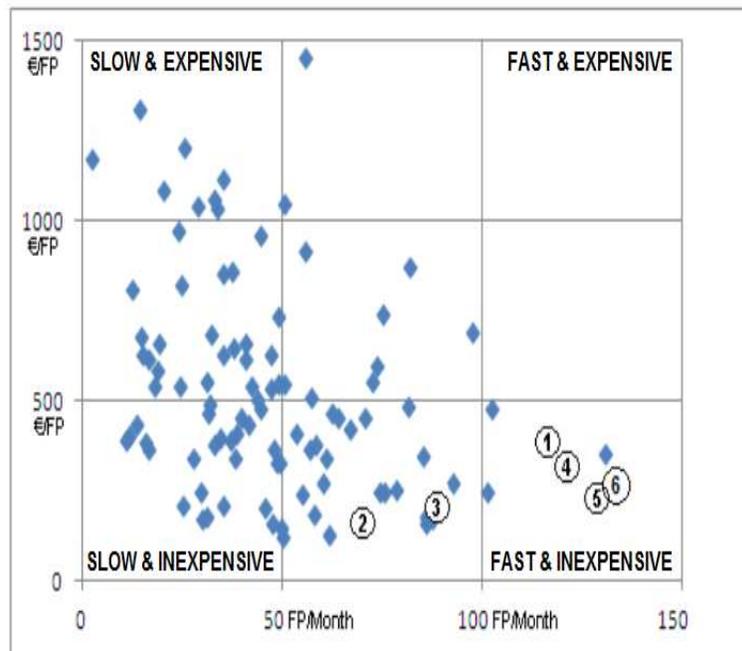


Figure 6. Speed of system delivery (realized function points/calendar month) and unit price (€/fp) of a sample of 107 completed information system projects in the Finnish public administration (retrieved from the repository)

The only project (shown in Figure 6 as a data point right above projects 5 and 6) among the five fastest projects that did not deploy northernSCOPE™ delivered a system for collecting, storing, and distributing information about all educational programs in Finnish universities. Potential students can use the information to select and apply for appropriate programs. Project team members were exceptionally experienced and could allocate adequate working hours without overly pressure for progress. The system was realized using Oracle databases and tools optimal for the chosen centralized mainframe environment. The system was smaller and simpler than the ones delivered by projects 1, 4, 5, and 6 that utilized heterogeneous web-based technologies such as Java, HTML, and SQL, respectively, for programming, user interface design, and database management. The excellent circumstances of the project and the simplicity of the delivered system thus made it possible for this one project to reach the same speed of delivery as projects 1, 4, 5, and 6.

Four projects leveraging the theory-ingrained artifacts were sourced between 2006 and 2010 by Ministry of Justice, Finland. This study covered the entire five year period. The projects 1, 2 and 3 developed the election information system for arranging national elections, municipal elections and referendums. The development program lasted five calendar years and was challenging for a variety of reasons. For example, the system has a life span of app. 20 years. It can have no software errors and has to support numerous stakeholders exactly during the predefined periods when elections and referendums are held. The software code, databases, and all other parts of the system have to be easily auditable and the system has to meet most stringent security requirements to ensure, for example, that the election results cannot be manipulated through the system. Project 4 developed an information system for public legal aid, providing individuals with the possibility to obtain assistance for legal matters electronically and fully or partially at the expense of the state. Citizens needing legal assistance can use the system to check whether they are eligible for financial aid and submit applications for assistance online. If they are eligible, the lawyers helping them can charge and be reimbursed for their services through the system. Public legal aid offices distributed throughout Finland use the system to process the applications and pay for the services of the lawyers and other expenses. The system deals with complex processing rules, interfaces with large systems (e.g., the Finnish population information system), and is used by thousands of citizens annually.

Martti Karjalainen, a scope coordinator at Ministry of Justice, had used functional size measurement since early 1990s and actively encouraged other stakeholders to use it in the ministry. In the projects 1-4, Ministry of Justice deployed the theory-ingrained artifacts largely. It sourced requirements specification work from external providers, employed the external scope manager to measure the scope of each project based on the original specifications, deployed €/fp unit pricing in requests for tenders and in contracts with the chosen system providers, and used the external scope manager at least once to measure the scope and the earned value (i.e., the ratio of the functional size of already realized functionality and the expected total functional size of the completed software) of each project during system delivery. Ministry of Justice found that by using the estimating process of the theory together with an external scope manager, it was better able to control costs and manage functional outcomes, schedules and quality of software development. The use of an external scope manager also increased the quality of requirements. Providers embraced the use of the functional size based unit pricing model in sourcing contracts because it enabled a common language for project scoping and co-operation between Ministry of Justice and the providers, resulting in less disagreement. Finally, the theory-ingrained artifacts lowered the unit prices from the range between 500 €/fp and 1000 €/fp to app. 300€/fp. The unit prices of projects 1-4 were, respectively, 409, 140, 167, and 320 €/fp. The average unit price of projects 1-4 was 281 €/fp, that is, approximately a half of the average unit price (551 €/fp, see Table 2) of the 101 other projects of the sample that did not use the theory-ingrained artifacts. Based on these results, Ministry of Justice has expressed an interest to expand the use of function points and unit pricing to include enhancement requests conducted on their legacy systems.

Projects 1-4 do not significantly differ from the other projects in the sample in terms of the hardware and software platforms the developed systems were running on, the development tools used, and the system architectures and the functional complexity and scope of the developed systems. All projects in the sample represent the same customer business area, that is, public administration. Three different providers conducted the four projects. These providers have delivered several other systems in the sample but without the control provided by the theory-ingrained artifacts and with a slower speed of delivery and a higher unit price. It is thus reasonable to assume that the use of the theory-ingrained artifacts has been an important common denominator in the four projects, helping the customer and the providers to reach lower unit prices and faster speeds of delivery.

Projects 5 and 6 belong to the VALTIMO program of Ministry of Social Affairs and Health. They were sourced between 2008 and 2012 to deliver subsystems of the Vera system for the use of five regional departments of Occupational Safety and Health (OSH) Administration. OSH inspectors visit different employer organizations to oversee that employers continuously monitor the safety of the work environments and practices and take appropriate remedial actions whenever employees report risks discovered in working conditions, methods, or tools. According to the productivity objectives of the Ministry of Social Affairs and Health, each inspector should perform about 100 on-site inspections annually, collect, and store a structured set of data about the inspected organization during the inspection, so the data is comparable and available for reporting and for assigning inspectors to organizations that most urgently need to be inspected. Inspectors have traditionally collected data in unstructured ways and stored the data in various information systems *after* their visits, requiring lots of office work. Project 5 delivered a subsystem enabling inspectors to collect and store structured data and prepare inspection reports on-site. Project 6 delivered a subsystem utilizing the collected data to help to plan, allocate resources to, and monitor inspection work.

Ministry of Social Affairs and Health created the initial requirements specifications in close collaboration with the representatives of the inspectors for projects 5 and 6. Teija Inkilä, manager of the VALTIMO program, had become aware of the theory while working in Ministry of Justice. She hired the external scope manager in the beginning of both projects to review the specifications and ensure they were written in a format facilitating the use of functional size measurement. Some usability, system architecture, and other consultants were also deployed but they had a minimal part of the total work effort. Both projects used the theory-ingrained artifacts throughout the sourcing life-cycle from initial requirements specification through requesting a tender, evaluating potential providers, contracting, managing changes, and monitoring progress to the post-mortem and closing of the project. At the time of writing this paper, two other projects have been started to realize two new subsystems of Vera. They also follow the theory fully. All Vera projects have been provided by Gofore, a software company committed to following the theory during its collaboration with Ministry of Social Affairs and Health.

Projects 5 and 6 do not significantly differ from the other projects in the sample in terms of the hardware and software platforms the developed systems used, the development tools used, and the system architectures and the functional complexity and scope of the developed systems. The projects have the fastest speed of delivery in the sample. They exhibit some of the lowest unit prices (200 €/fp and 225 €/fp). The unit prices are far less than a half of the average unit price (551 €/fp, see Table 2) of the 101 other projects of the sample that did not use the theory-ingrained artifacts.

Both the provider and the customer have been satisfied with the improved (1) transparency the theory-ingrained artifacts have provided during the projects and (2) mutual understanding of what has already been achieved and what will be done next (Välikoski, 2012). Markku Marjamäki (2012), chair of the steering committee of the VALTIMO program, states:

“The selection of the provider and the contracting with the provider [in the VALTIMO program] were largely based on the price of a function point. This has proven to be a good solution. Detailed contractual negotiations between the customer and the provider have taken place effortlessly. The FiSMA 1.1 Functional Size Measurement method (ISO, 2010) has been used broadly ... to analyze the initial project scope, to measure the functional size of the completed information system, in contractual arrangements between the customer and the provider, and to monitor the progress of the provider.”

Gofore reached perfect scores in terms of customer satisfaction and the meeting of customer needs from a customer survey in the VALTIMO program (Gofore, 2013). Ministry of Social Affairs and Health is now one of the reference customers of Gofore and the parties have established an ongoing partnership.

Table 2. shows the comparison between the six projects, which applied the theory-ingrained artifacts, and 101 other projects of the sample. The six projects investigated by the two studies had the average unit price of 244 €/fp, which is far less than a half of the average unit price (551 €/fp) of the other projects of the sample. The studies show the meta-design was comprehensively leveraged and provide strong support for the realization of the design value proposition 2 by the case organizations. The six projects had the average speed of delivery of 107,7 fp/month, which is more than a double of the average speed of delivery of 49,9 fp/month in the other projects of the sample that did not use the theory-ingrained artifacts. The studies provide strong support for the realization of the design value proposition 1 by the case organizations. Both case organizations got the functionality they needed without sacrificing any major quality requirements in the six projects, realizing the design value proposition 3.

Subset of data	N	Unit price (€/FP)			Speed of delivery (FP/month)		
		Min	Max	Average	Min	Max	Average
Sample	101	125	1454	551	2.6	131.1	49.1
Projects using theory-ingrained artifacts	6	200	409	244	71.4	131.8	107.7

Table 2. Comparison of the 6 projects using the theory-ingrained artifacts and 101 other projects of the sample

About the Research Methodology Deployed for Validation

The scientific validation of the theory is a part of the doctoral dissertation project of the co-author. In addition to his research activities, he is the primary owner of a company that provides services based on the theory, develops and sells the software product supporting the theory, and maintains the repository. The case organizations are clients of the company. In the roles of an entrepreneur, an external scope manager, and an action-design researcher, the co-author has collected and analyzed an enormous amount of software project scope management related data from not only the case organizations but also dozens of other organizations in many countries during the period of 25 years. This data has played a key role in iteratively refining and generalizing the theory over the years through action-design research (Sein et al., 2011).

The role of an entrepreneur can bias the research results. For example, good results can support business development and marketing of the company. Several measures have been taken to eliminate this possible bias.

First, this research has been conducted extensively based on documents produced by the studied organizations to ensure authenticity. For example, Figures 3, 4, and 5 are documents produced by TKP Tieto. The repository, containing data mostly produced by providers, has also been used (e.g., Figures 2 and 6).

Second, the workload of the co-author as an external scope manager has never been more than 14 hours per month in any of the six case projects of the ministries, the working hours of the co-author have amounted to less than 1 % of the total development effort, and the co-author has always acted in an advisory role without decision-making authority. TKP Tieto institutionalized major parts of the theory to the extent it enacted the theory independently of the co-author for years. Therefore, the personal impact of the co-author could hardly be significant for the success of all case organizations in realizing the six design value propositions.

Third, the other authors have been involved with software project scope management research for many years. They are professional information systems researchers having nothing to do with the company. They have independently investigated the two customers and the provider in the case studies. In 2010, they interviewed in TKP Tieto the director responsible for system development practices, methods, and tools, including scope management, and two of the three scope coordinators. All interviews were recorded, transcribed, and analyzed. The interviews revealed that TKP Tieto did not achieve its success solely based on scope coordinators' and project managers' widespread use of the software project estimation and measurement processes prescribed by the theory. It developed a level of software project scope management expertise reaching beyond the theory. For example, it could (1) select the most appropriate effort estimation methods considering the specific characteristics of projects and (2) use multiple effort estimation methods in parallel to produce triangulated, accurate effort estimates. Generally, the interview results were in line with the lessons learnt by the co-author. For example, the system development director of TKP Tieto supported the use of scope management and wanted to invest more in scope management training in his organization. The analysis of developments of TKP Tieto since 2005 is beyond the scope of this paper due to confidentiality reasons.

Finally, Teija Inkilä, manager of the VALTIMO program, and Markku Marjamäki, chair of the steering committee of the VALTIMO program, have reviewed and accepted the section of this paper that presents the projects of Ministry of Social Affairs and Health. Paula Männistö, a scope coordinator at TKP Tieto during the study period, has reviewed and accepted the section dealing with TKP Tieto. Martti Karjalainen, a scope coordinator at Ministry of Justice during the study period, has reviewed and accepted the sections focusing on Ministry of Justice.

Conclusions and Future Research

This paper presented a theory supporting *customers and providers* in function-point-based planning, pricing and contracting, delivery tracking, and change management throughout the software sourcing life-cycle. To validate the theory, this paper leveraged the six design value propositions of the theory to analyze how and why the theory-ingrained artifacts have helped (1) Ministry of Justice, Finland and Ministry of Social Affairs and Health, Finland to source software and (2) TKP Tieto to improve its productivity and software delivery. TKP Tieto instituted the theory-ingrained artifacts from project initiation to closure to the extent it became completely autonomous in enacting the theory. A minor deviation was the lack of systematic reuse analysis during the software project effort estimation process (Figure 1). The role of an internal scope coordinator was critical for institutionalization. Customers owned a majority stake in TKP Tieto and did not use external scope managers and the functional size based unit pricing model in their sourcing engagements with TKP Tieto. In sum, during the rigorous study spanning the period of more than 15 years, TKP Tieto learnt to learn from its experiences and to manage project scope through software project effort estimation and measurement processes aligned with the theory. Ministry of Justice and Ministry of Social Affairs and Health employed the co-author in the role of an external scope manager and used the functional size based unit pricing model in six projects. During the five-year-long case studies, they successfully leveraged the meta-design from project initiation to closure in the projects.

The answers to both research questions of this research paper were positive. The studies provided evidence that the theory is likely to be valid from both customers' and providers' viewpoints. Ministry of Justice, Finland and Ministry of Social Affairs and Health, Finland deployed the theory-ingrained artifacts to systematically source software faster without giving up any functional or quality requirements while reducing software sourcing costs to a half. TKP Tieto deployed the central tenets of the theory to almost double its productivity in dozens of software development projects annually and to systematically deliver software that met customer requirements on time and on budget. It achieved these results for years independently, providing strong evidence for the validity of the theory from the providers' viewpoint.

Specific sets of theory-ingrained artifacts were deployed by the case organizations to achieve the results. However, this does not mean that only the use of those sets could yield similar results in other organizations. For example, the ISO/IEC 29881:2010, FiSMA 1.1 Functional Size Measurement method is one of the artifacts deployed in this research. All ISO/IEC standardized functional size measurement methods are conformant with the concepts and definitions provided by the ISO/IEC 14143-1 standard (ISO, 2007). Because the ISO/IEC 29881 standard shares these common characteristics (ISO, 2007; ISO, 2012) with the other ISO/IEC standardized functional size measurement methods, using any of the other methods might yield similar results. Future research is needed to investigate other sets of artifacts in other organizations.

This research found that when excellent project teams create small and simple systems using methods and tools optimized for the systems, they do not necessarily need to use the theory-ingrained artifacts to reach the productivity and speed of delivery similar to the projects using the artifacts. Therefore, it can be hypothesized that projects dealing with small and simple systems may not benefit from the theory to the same extent as larger projects dealing with systems that are more complex. Future research is needed to investigate this hypothesis. It should also be noted that the theory should not be expected to help, if the customers cannot specify their needs or the providers cannot implement software. If they can, as was mostly the case with the three case organizations, the theory can be expected to help them to avoid scope creep by focusing on the amount of delivered software artifacts.

Both ministries hired the co-author to provide scope management services in executing the theory. The success achieved cannot necessarily be attributed solely to the theory because other external scope managers might have facilitated the achievement of different results using the theory. However, Wright (2000) found that other scope managers could achieve equally valuable results for the customers. Moreover, the co-author spent an insignificant number of working hours without decision-making authority in the case organizations. Therefore, his personal impact on the results manifested by the studies was negligible.

One external scope manager (i.e., the co-author) and three Finnish case organizations participated in this research. The organizations are relatively small and are likely to deal with relatively small and simple systems when compared to organizations having similar responsibilities in much larger countries such as China, India, and the USA. However, the theory can be expected to be scalable and generalizable to larger organizations and systems because one of the very purposes of the theory (e.g., step 2 of the 12 step scoping process, Table 1) is to divide large systems and development programs into small and manageable subsystems and subprojects. The size of a participating organization should not be an issue, if the 12 steps are applied correctly. This was the case in the three organizations studied in this paper. Almost all of the studied development programs in these organizations were divided into more than one subproject. To examine the generalizability, scalability, and validity of the theory, future research is needed to investigate (1) larger providers and customers dealing with public administration in such larger countries; (2) providers and customers outside public administration, and (3) the results providers and customers obtain through the theory-ingrained artifacts by employing other scope managers to facilitate software sourcing and delivery.

This research also has theoretical implications that need to be addressed by future research. The theory deals with software project scoping during the sourcing life-cycle. Providers and customers may benefit from comprehensive sourcing life-cycle models to conduct systematically not only the scoping, estimation, and measurement processes but also all other major processes in various phases of the life-cycle. However, most extant sourcing life-cycle models (e.g., Hefley and Loesche, 2010; Hyder et al., 2010) do not explicitly support the implementation of the theory presented in this paper. Future research is necessary to create and validate sourcing life-cycle models for customers and providers that combine

software project scoping, estimation, and measurement processes and functional size based unit pricing with other major processes throughout the sourcing life-cycle (Käkölä, 2008; Käkölä and Lu, 2015).

References

- Alter, S. 2008. "Service System Fundamentals: Work System, Value Chain, and Life Cycle," *IBM Systems Journal* (47:1): 71-85.
- Basili, V.R., Caldiera, G., and Rombach, H.D. 2002. "Experience Factory," in *Encyclopedia of Software Engineering*, John Wiley & Sons.
- Boehm, B. 1981. *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice-Hall.
- Boehm, B.W., Abts, C., Winsor Brown, A., Chulani, S., Clark, B.K., Horowitz, E., Madachy, R., Reifer, D., and Steece, B. 2000. *Software Cost Estimation with COCOMO II*, Englewood Cliffs, NJ: Prentice-Hall.
- Bundschuh, M., and Dekkers, C. 2008. *The IT Measurement Compendium: Estimating and Benchmarking Success with Functional Size Measurement*, Berlin, Germany: Springer.
- Chrissis, M.B., Konrad, M., and Shrum, S. 2011. *CMMI®: Guidelines for Process Integration and Product Improvement*. Addison-Wesley Professional, SEI Series on Software Engineering (Third edition).
- Dekkers, C., and Forselius, P. 2010. "Scope Management: 12 Steps for ICT Program Recovery," *CrossTalk: the Journal of Defense Software Engineering* (23:1), pp. 16-21.
- Eisenhardt, K.M. 1989. "Building Theories from Case Study Research," *The Academy of Management Review* (14:4), pp. 532-550.
- European Certification and Qualification Association. 2015. *ECQA Certified SCOPE Manager*, <http://www.ecqa.org/index.php?id=38>.
- FiSMA. 2001. *Experience ND21 Situation Analysis Method*. http://www.fisma.fi/wp-content/uploads/2006/09/fisma_situation_analysis_method_nd21.pdf.
- FiSMA. 2002. *FiSMA Reuse Measurement Method, FiSMA RMM version 2002*. http://www.fisma.fi/wp-content/uploads/2006/09/fisma_reuse_analysis_method_10.pdf.
- Forselius, P., Dekkers, C., Karvinen, M., and Kosonen, M. 2008. *Program Management Toolkit for Software and Systems Development*, Helsinki, Finland: Talentum Media.
- Forselius, P., and Käkölä, T. 2009. "An Information Systems Design Product Theory for Software Project Estimation and Measurement Systems," in *Proceedings of the 42nd Annual Hawaii International Conference on Systems Sciences (HICSS)*, IEEE.
- Gofore. 2013. *STM-Työsuojelun Tarkastusjärjestelmä-Julkisen sektorin tunnetusti tehokkain tietojärjestelmäprojekti*, <https://gofore.com/asiakkaat/#stm-tyosuojelun-tarkastusjarjestelma>.
- Gregor, S. 2006. "The Nature of Theory in Information Systems," *MIS Quarterly* (30:3), pp. 611-642.
- Gregor, S., and Jones, D. 2007. "The Anatomy of a Design Theory," *Journal of the Association for Information Systems* (8:5), pp. 313-335.
- He, M., Li, M., Wang, Q., Yang, Y., and Ye, K. 2008. "An Investigation of Software Development Productivity in China," in *Proceedings of the International Conference on Software Process*, Q. Wang, D. Pfahl, and D.M. Raffo (eds.), LNCS 5007, Berlin, Germany: Springer, pp. 381-394.
- Hefley, W.E., and Loesche, E.A. 2010. *The eSourcing Capability Model for Client Organizations (eSCM-CL)*. Van Haren Publishing.
- Hevner, A.R., March, S.T., Park, J., and Ram, S. 2004. "Design Science in Information Systems Research," *MIS Quarterly* (28:1), pp. 75-105.
- Hill, P. 2010. *Practical Software Project Estimation: A Toolkit for Estimating Software Development and Duration*, McGraw & Hill Education.
- Hyder, E.B., Paulk, M.C., Heston, K.M., and Hefley, B. 2010. *eSourcing Capability Model for Service Providers (eSCM-SP)*, Van Haren Publishing.
- ISO. 2004. ISO/IEC 15504, Information technology -- Process assessment -- Part 1: Concepts and vocabulary, Geneva, Switzerland: International Organization for Standardization, <http://www.iso.org>.
- ISO. 2007. ISO/IEC 14143-1, Information Technology -- Software Measurement -- Functional Size Measurement-Part 1: Definition of Concepts, Geneva, Switzerland: International Organization for Standardization, <http://www.iso.org>.

- ISO. 2010. ISO/IEC 29881:2010, Information Technology -- Systems and Software Engineering -- FiSMA 1.1 Functional Size Measurement Method, Geneva, Switzerland: International Organization for Standardization, <http://www.iso.org>.
- ISO. 2011. ISO/IEC 29155-1, Systems and Software Engineering -- Information Technology Project Performance Benchmarking Framework -- Part 1: Concepts and Definitions, Geneva, Switzerland: International Organization for Standardization, <http://www.iso.org>.
- ISO. 2012. ISO/IEC 14143-6, Information Technology -- Software Measurement -- Functional Size Measurement-Part 6: Guide for use of ISO/IEC 14143 series and related International Standards, Geneva, Switzerland: International Organization for Standardization, <http://www.iso.org>.
- Käkölä, T. 2008. "Best Practices for International eSourcing of Software Products and Services," in *Proceedings of the 41st Annual Hawaii International Conference on Systems Sciences (HICSS)*, IEEE.
- Käkölä, T., and Lu, Y. 2015. "Best Practices for International eSourcing of Software Products and Services from the Service Providers' Perspective," in *Proceedings of the 48th Annual Hawaii International Conference on Systems Sciences (HICSS)*, IEEE.
- Lee, A. 1989. "A Scientific Methodology for MIS Case Studies," *MIS Quarterly* (13:1), pp. 33-50.
- Marjamäki, M. 2012. "IT-hankinnoissa Voidaan Onnistua," *Helsingin Sanomat*, October 1, 2012, Helsinki, Finland, <http://www.4sumpartners.com/wp-content/uploads/2013/01/HS-mielipide-Oct-10-2012.pdf>.
- Markus, M.L., Majchrzak, A., and Gasser, L. 2002. "A Design Theory for Systems that Support Emergent Knowledge Processes," *MIS Quarterly* (26:3), pp. 179-212.
- Maxwell, K.D., and Forselius, P. 2000. "Benchmarking Software Development Productivity," *IEEE Software* (17:1), pp. 80-88.
- Morris, P. 2004. "Metrics Based Project Governance," in *Proceedings of the 14th International Workshop on Software Measurement (IWSM 2004)*, Aachen, Germany: Shaker Verlag.
- Premraj, R., Shepperd, M.J., Kitchenham, B.A., and Forselius, P. 2005. "An Empirical Analysis of Software Productivity over Time," in *Proceedings of the 11th IEEE International Symposium on Software Metrics*, IEEE Computer Society.
- Project Management Institute (PMI). 2013. *A Guide to the Project Management Body of Knowledge (PMBOK)*, 5th Edition.
- Rollo, T., and Wright, T. 2001. "SouthernSCOPE - A Method for Acquiring Custom Built Software," in *Proceedings of ESCOM 2001 Conference*, April 2-4, 2001, Maastricht, The Netherlands: Shaker Publishing, pp. 183-192.
- Remenyi, D., and Sherwood-Smith, M. 1999. "Maximize Information Systems Value by Continuous Participative Evaluation," *Logistics Information Management* (12:1), pp. 14-31.
- Sein, M., Henfridsson, O., Purao, S., Rossi, M., and Lindgren, R. 2011. "Action Design Research," *MIS Quarterly* (35:1), pp. 37-56.
- Simon, H. 1996. *The Sciences of the Artificial*, 3rd edition, Cambridge, MA: MIT Press.
- Van Aken, J. 2004. "Management Research Based on the Paradigm of the Design Sciences: The Quest for Field-tested and Grounded Technological Rules," *Journal of Management Studies* (41:2), pp. 219-246.
- Victorian Government, Australia. 2001. *SouthernSCOPE Puts the Control of Software Development in the Hands of Business*.
- Välikoski, A. 2012. "Työsuojeluhallinnon Tietojärjestelmäprojektista Kohulööppien Sijaan Kehuotsikoita," *Gofore*, http://issuu.com/gofore/docs/gofore_lehti_2012_aukeamittain, pp. 11-13.
- Walls, J.G., Widmeyer, G.R., and EI Sawy, O.A. 1992. "Building an Information System Design Theory for Vigilant EIS," *Information Systems Research* (3:1), pp. 36-59.
- Walls, J.G., Widmeyer, G.R., and EI Sawy, O.A. 2004. "Assessing Information System Design Theory in Perspective: How Useful was Our 1992 Initial Rendition?" *Journal of Information Technology Theory and Application* (6:2), pp.43-58.
- Wright, T. 2000. *Controlling Software Development: a Breakthrough for Business, Software Engineering Australia*, November 2000.