Association for Information Systems
AIS Electronic Library (AISeL)

ECIS 2008 Proceedings

European Conference on Information Systems (ECIS)

2008

The Nipo Grid - A Construct for Studying Systems Development Practices in Organizations

E Larsen
University of Agder, even.larsen@uia.no

T Päivärinta
University of Agder, tero.paivarinta@uia.no

K Smolander
University of Agder, kari.smolander@lut.fi

THE NIPO GRID – A CONSTRUCT FOR STUDYING SYSTEMS DEVELOPMENT PRACTICES IN ORGANIZATIONS

Even Larsen, University of Agder, Department of Information Systems, Serviceboks 422, 4604 Kristiansand, Norway, even.larsen@uia.no

Tero Päivärinta, University of Agder, Department of Information Systems, Serviceboks 422, 4604 Kristiansand, Norway, tero.paivarinta@uia.no

Kari Smolander, Lappeenranta University of Technology, Department of Information Technology, P.O.Box 20, 53851 Lappeenranta, Finland, kari.smolander@lut.fi

Abstract

This paper presents intermediate results from a study of issues which influence shaping of information systems development (ISD) practices in systems and software development organizations. In line with previous research, the target organizations of our study tend to create local development practices by picking individual practices from (possibly different) methods combined with emergent practices, rather than by deploying completely pre-defined methods. ISD practices-in-use may vary with regard to the scope of their intended enactment and to their actual utilization. This article suggests a construct by which any defined development practice (or absence of defined practices) can be categorized according to two dimensions: the intended scope of defined practices vs. the scope of actually enacted practices. With regard to the both of these dimensions, any particular practice may be defined versus actually followed at four levels: not at all (N), individually (I), project-level (P) or by the whole development organization (O). The resulting construct, "The NIPO Grid", integrates the previous discussions and concepts about how ISD practices are actually shaped in organizations, such as amethodical systems development, the method-in-use, and software process improvement. We propose it as a construct, a dependent variable, which helps our research towards forming a theory or theories why definitions and actual enactment of particular types of development practices vary among development organizations.

Keywords: information systems development, software development, practice

1 INTRODUCTION

From very early on ISD research has focused on how development should be carried out (e.g. Dijkstra 1965, Parnas 1972). This led to the construction of numerous formal systems development methods (SDMs) (Jayaratna 1994, Avison & Fitzgerald 2002). Adherence to methods was regarded by default as useful and methods thus were to be rigorously pre-defined and applied as intended by the method developers (Humphrey & Snyder & Willis 1991, Jarke & Pohl & Rolland & Schmitt 1994). On the contrary, a number of ISD researchers have argued that SDMs undergo a great extent of pragmatic adaptation during their adoption and use, if used at all (Stolterman 1992, Fitzgerald 1998, Kautz 2004, Vogelsang & Kensing 2006, Päivärinta & Sein & Peltola 2008). One stream of research even suggested that "amethodical" ISD represents an alternative view on development practice in contemporary ISD organisations. That is, methods would have no prescribed role at all, while ISD practice emerges through contextual interaction and improvisation (Truex & Baskerville & Travis 2000).

A few empirical studies have focused on the viewpoint of individual systems developers and how they perceive the relevance of methods and utilize them in their work. These include works on the relationship between developer experience and modes of SDM use, or determinants of developer intentions to use methods (Hardgrave & Davis & Riemenschneider 2003). Education of reflective systems developers with regard to SDMs has also received attention (Mathiassen & Purao 2002). A few studies have discussed ISD project exigencies in relation to local method adoption, where the unit of analysis has been a project (e.g. Kautz 2004, Madsen & Kautz & Vidgen 2006).

A good stream of research has discussed methods in the context of systems development companies and organizations beyond individuals and particular projects. For example, Fitzgerald et al. (2003, 2006) focused on internal software development in large IT industry organizations. Mathiassen & Vogelsang (2005) and Päivärinta et al. (2008) have followed longitudinally how particular development methods have been adopted and adapted in professional development organizations. Beyond the discussion about methods, the "reflective systems development" (Mathiassen 1998), a.k.a. the "professional work practice" (Iivari & Lyytinen 1998), approach has focused on improving actual development practices in development organizations. It has focused especially on action research and local improvements based on contextual circumstances (Mathiassen 1998). In the 1980s (Mathiassen 1998), it was one of the first research programs which challenged the belief that development methods as such would improve the effectiveness of ISD (Iivari & Lyytinen 1998).

We shared this focus by asking how and why actual practices are shaped in professional systems and software development organizations. In the early stage, we met the challenge to conceptualize the patterns of defined and enacted ISD practices in our case organizations. Few unifying frameworks for this purpose exist. For example, Fitzgerald, Russo and Stolterman (2002) presented the "method-in-action" framework, which explain the influence of a number of factors on contemporary ISD practices. It has been used in a number of works, e.g. by Madsen et al. (2006) to describe how a local method emerged in an individual systems development project. Unlike such works as Kautz (2004) and Madsen et al. (2006) which focused only on individual systems development projects, we have an extended view that includes the level of systems development organizations, which, of course, may take part in numerous projects. Software process improvement frameworks such as CMM (Humphrey 1989) build on more detailed models of the development process at the level of the organization. However, their intended use is to assess organizational capabilities by benchmarking the actual ISD practice to a predefined set of key practices, which leaves the practice outside the key practices unexplored.

This paper suggests an analytical construct by which any defined development practice (or absence of defined practices) in a professional development organization can be described and categorized according to two dimensions: the intended scope of a defined practice vs. the scope of an actually used practice. With regard to the both of these dimensions, any particular practice may be defined versus actually followed at four levels: not at all (N), individually (I), project-level (P) or by the whole

development organization (O). The resulting construct, "The NIPO Grid", integrates the previous discussions and concepts about how ISD practices are actually shaped in organizations. Moreover, we argue for the use of the construct for forming theory or theories about why particular types of practices may be differently shaped across development organizations.

After providing a conceptual and theoretical background for our study in section 2, section 3 declares the research process in more detail. The empirical grounding for the NIPO grid is illustrated in section 4. Section 5 discusses the previous concepts related to ISD practices in light of the construct. Section 6 concludes with suggestions for further research.

2 CONCEPTUAL AND THEORETICAL BACKGROUND

It is important to declare what we mean by the concepts 'method' and 'a practice'. A *systems development method* (*SDM)* can be defined as "an organized collection of concepts, beliefs, values, and normative principles supported by material resources" (Lyytinen 1987). All thorough methods define the process, a set of description or coordination techniques to be used in the process steps, a concept structure to be used in the description or coordination tasks, division of work by stakeholder roles and a set of values which guideline the work (Tolvanen 1998). The description techniques may be formal, semiformal, or informal and they express a viewpoint on the system to be developed. A method may be supported by a set of tools supporting the descriptions and notations and the execution of the process. It can be targeted at a specific application domain and it has a certain perspective implying certain values and beliefs for the development process. (Iivari & Hirschheim & Klein 1998, 1999).

While the definition of an SDM above states that a method represents a well-defined structure for development work, ISD may include also more loosely organized practices. Therefore we use the concept of *a practice* when we refer to an ISD activity which is conducted repeatedly in a somewhat similar manner in systems development. Here, we follow a standard dictionary definition for a practice which means *"something people do regularly"* (Collins Cobuild English Dictionary 1989). An SDM used in an organization always embodies a predefined practice or a set of them, whereas a practice is not always defined at the level of a method (at least with regard to all elements of method knowledge declared by Tolvanen 1998). A practice can be explicitly defined, e.g. as a technique specified by a method, or emerge as a habitual response to a recurrent situation. In the former case, there exists an explicit definition of it and the organization has somehow expressed its importance.

During the data analysis we noticed quickly that not all practices were defined in a documented form and, on the other hand, not all defined practices were followed. This led us to make an analytical distinction between the *intended scope* of a practice in question vs. its actual *enactment*, i.e. the performance with regard to whether the practice in question is actually followed or not. Theoretically, this distinction resembles Argyris' and Schön's theory of action (1974) with regard to their influential concepts of espoused theory and theories-in-use. Theory-in-use is the implicit theory held by a human "that actually governs his actions", whereas espoused theory involves assumptions about more general or socially acceptable expectations concerning human behaviour in the given situation (Argyris & Schön 1974, p. 6-7). However, our concepts focus perhaps less on the fit between "inner feelings" of the stakeholders and their espoused theories of action, which is the focus of Argyris and Schön. Rather, our work plainly sensitizes that actions do not always follow the pre-defined practices.

3 RESEARCH PROCESS

This study represents the first phase of an ongoing project focusing on the issues that shape Information Systems Development (ISD) practices in organizations. We wanted to find an analytical concept to discuss what practices exist (and do not exist) in the target organizations. The empirical results are, in total, based on 39 interviews with developers and managers in 10 IS development organizations (Table 1). The interviews were tape-recorded and transcribed.

Org	Employees	Business	Interviewees
A	6, one site	Small web-based e-business solutions	Sr. Developer (13 years of work experience)
B	8 (Norway, one site) + 10 (India)	Logistics solution for electricity networks	Developer (5 y exp.) Development manager (well-experienced)
C	42, one site	System development and administration	Developer (8 y exp.) Project manager (30 y experience)
D	25 (Norway, one site) , (Serbia)	Systems development and consulting	Developer & Project manager (12 y exp.) Developer (13 y exp.)
E	4 (local office) , 184 (Norway altogether)	Systems development and consulting	Developer/Consultant (8 y exp.) Developer/Consultant/ Project mgr (8 y exp.)
F	800 (in multiple countries)	Operative systems for industry	3 Upper managers (11, 20, and 22 y exp.) 2 Project managers (5 and 17 y exp.) 2 Developers (7 and 12 y exp.)
G	100 (Finland, one site)	Automation systems for industry	5 Upper managers (13, 8, 12 and 20 y exp.) 2 Project managers (10 and 6 y exp.) 1 Developer (6 y exp.)
H	200 (Russia, one site)	Software development services, subcontracting	2 Upper managers (10+ and 26 y exp.) 1 Project manager (10 y exp.) 2 Developers (7 and 10 y exp.)
I	60 (Finland, one site)	Systems for a specific kind of resource planning	3 Upper managers (6, 4 and 6 y exp.) 2 Developers (5 and 5 y exp.)
J	80 (Finland, one site)	Internal systems development organization	1 Upper manager (20 y exp.) 2 Project managers (8 and 20 y exp.) 1 Analyst (5+ y exp.) 1 Architect (5 y exp.)

Table 1: Case organizations

We started open-mindedly with the grounded theory approach (Strauss & Corbin 1990), due to a lack of unified frameworks and due to our hopes of finding issues not previously found in studies based on pre-theorized factors (such as Khalifa & Verner 2000, Hardgrave et al. 2003). The data analysis in this paper is based on the developers' own descriptions of how systems development tasks are conducted in the target organizations. The final construct with its dimensions resulted from open and axial coding, and emerged gradually during discussions about what the core category should be for selective coding. However, our interpretations were surely influenced by our previous knowledge from the literature, and the resulting theoretical construct can thus be regarded as a synthesis of the previous systems development literature and the insight we gained in the studied cases.

4 THE NIPO GRID

Some practices in the target organizations were defined project-specifically, some at the level of the organization. Such standardization of a practice may originate from a decision taken by managers, project managers, project teams, or individuals. On the other hand, there could be areas of practice not defined at all, or the individuals could follow the practices they liked. The *intended scope* of a defined practice refers to this dimension, which can be: the organization, a project, an individual, or none if no decision have been made for unifying a practice. A definition of a practice can refer to a documented formal ISD method, third party or in-house. We saw also practices defined by oral agreements or common understanding of what the practice is, particularly in smaller organizations and project groups. Emerging practices may not have an explicit definition, but they can be defined by describing the enactment, if observed in the target organization.

Another dimension of a practice is the variation in enactment. A practice may be enacted in a similar way throughout the organization, or a project, or by an individual, or no practice at all may be followed.

We use the term *scope of enacted practice*. The enacted scope dimension is independent of the intended scope. These two dimensions can be represented in tabular form in "the NIPO grid" (NIPO being an acronym of the words None, Individual, Project, Organization) as shown in Table 2.

Scope of enacted practice	**Intended scope of defined practice**			
	None	**Individual**	**Project**	**Organization**
None	There is no intention to employ any practice, no practice is followed	An individual intends to employ a practice, but does not follow it.	A project intends to employ a practice, but it is not followed at all	The organization intends to employ a practice, but it is not followed at all
Individual	An individual implicitly follows a practice	An individual follows an individually defined practice as intended	A project intends to employ a practice, only some individuals follow it	The organization intends to employ a practice, some individuals follow it
Project	An emergent practice is followed by all project members.	A practice intended to be used by one individual is enacted by all project members	A project follows a practice as intended	The organization intends to employ a practice but it is only followed in some projects
Organization	A practice is emerging in the same way throughout the organization	A practice intended to be used by an individual is enacted throughout the organization	A practice intended to be used by one project is followed by all	A practice is followed as intended throughout the organization

Table 2: The NIPO Grid – Intention vs. Enactment

In the following discussion we will use the letters from the NIPO acronym as the indices referring to particular rows and columns in the grid, respectively, e.g (N-P) refers to the None row and Project column. The upper right half (the N-I, N-P, N-O, I-P, I-O, and P-O cells) of Table 1 represents situations where a decision to employ a specific practice is not followed up by actual enactment as planned. The main diagonal (the N-N, I-I, P-P, and O-O cells) represents the "ideal fit" situations (cf. Argyris & Schön 1974) where practices are enacted exactly as intended. The remaining cells represent situations in which a developer may enact a practice because of his training, or simply by habit, without consciously deciding to do so. This is represented by the I-N cell. However, existence of such practices cannot be observed through our interview data. It is possible that an enacted practice may emerge in a project or an organization for similar reasons, as represented by the P-N and O-N cells. Such standardization would occur perhaps more likely because developers decide to copy successful practices from each other (represented by the P-I and O-I cells). Finally it is possible that practices are repeated across projects (the O-P cell), e.g. because a project team simply continue the practices from their previous project.

4.1 Examples

While the space does not allow us to report from all of our target organizations, we sketch the idea through three cases. The prominent instances of ISD practices are shown in a NIPO grid for each organization. The names given to the practices were either that used by the interviewees, or a descriptive word when the practice was described but not named.

4.1.1 Organization A

Company A is a small organization with 6 employees. They develop web-based e-business systems, based on a web publishing framework. Projects are always manned with a single developer. If a project

seems to exceed the resources, the company extends the duration of the project rather than involves a larger team. On the surface, the manager had a very negative attitude to SDMs:

"I am not for use of methods at all… It is time-consuming. We work with small projects and rarely have more than one person in a project. It is really difficult to coordinate more than one programmer at a time. It takes a lot of time."

The manager continued that he typically guidelines every project with regard to their way of working, while they have "no formal way to do it." Yet, the interview revealed several very strict manifest practices with regard to contracting, post-delivery support and allocating human resources to do projects. The company offers three strictly pre-defined types of solutions. If a particular project does not fit one of these types the company simply does not take it. Furthermore, they aim to offer no post-delivery support at all: "It is better for us to be done with the customer when the project is over. It is really boring with previous customers who call for fixes." They always try to reuse code and architecture of their publishing framework, which is not always possible, to implement the system.

<table>
<tr><th rowspan="2">Scope of enacted practice</th><th colspan="4">Intended scope of defined practice</th></tr>
<tr><th>None</th><th>Individual</th><th>Project</th><th>Organization</th></tr>
<tr><td>None</td><td colspan="2" rowspan="2">Testing
Programming</td><td></td><td></td></tr>
<tr><td>Individual</td><td></td><td></td></tr>
<tr><td>Project</td><td></td><td></td><td></td><td>Architecture
Code reuse</td></tr>
<tr><td>Organization</td><td></td><td></td><td></td><td>Project "management"
Contract negotiation
Support</td></tr>
</table>

Table 3: ISD practices in company A

Programming and testing practices are left to the discretion of the developer. Based on our data, we are not fully able to locate exactly the slot in the grid where the testing and programming tasks belong to, but as no attempt has been made to "standardize" practice on the organizational or project levels, it is evident that the maximum intended scope of practice would be an individual giving guidelines to himself. Nor is there any evidence of standardized enacted practice beyond the individual level. However the company is not entirely satisfied with this, as they have experienced some quality problems recently and they consider implementing stricter testing and release practices at the level of the organization.

"Yes, there is one area in which we have thought to introduce methods, which is testing and approval. Here we plan to formalize something. A third person can go in to do a quality check and to test. Sometimes the quality produced by developers may vary a bit."

To summarize, although there was a negative attitude to use of methods in general in organization A, there were at least three areas where practices had been defined rather strictly at the level of the whole organization, "guidelined" to each project by the manager. As well, most projects followed pre-defined architectural moulds for the solutions and attempted to re-use code as much as possible from previous projects, while the actual programming and testing practices were so far left to the individual developers to decide. However, testing and release practices were under scrutiny to be more strictly standardized for the quality reasons. (Table 3).

4.1.2 Organization B

Company B is a software vendor in the energy sector providing a logistics system for electrical power networks. In addition to 8 employees at the headquarters in Norway, the company has a team of 10 software developers in India. This seems to require more standardized specifications between the sites in the future, which, however, are not seen to replace person-to-person interactions to solve the

emerging problems. In Norway, the employee turnover has been very low. The company interacts closely with their main customers, developing and maintaining the system continuously.

"We're not project-oriented, we're product oriented, and our product has a life-cycle beyond one project… in addition, we have deliveries of add-on modules… but the functionality will live on." (Development manager)

The interviewees gave a fuzzy picture about the actual status of pre-defined development practices in the organization, which indicates that at least the management had intentions for organization-wide definitions for practices, without actual influence for the development practices followed, whereas the developer highlighted his individual freedom and experience in the development process, while feeling an individual level responsibility for documenting one's work for later use. As well, the company had had intentions to implement a company-wide design tool.

"We use no formal method tools, such as UML to coordinate the development process… there exists a kind of common understanding of what we do. It has cumulated through experience, but we have also documents of how to conduct the [development] process, so we have in one way a formal description of how the development process should go…I do not want to say whether we use a method or not, but there are different ways to do that. In the methods there are different processes involved, so that sometimes we have software for an in-house project, which does not involve a GUI at all. Then a number of steps is cut off as they are irrelevant for the process. But where it is relevant, we will largely follow the same procedure to reach the solution... I can set a frame or requirements for the end product, but I cannot set requirements for the development work in itself." (Development manager)

"There are no formal guidelines here… every developer is actually pretty free. I try for my part to think that if I disappear, that the others could understand the things I have done." (Developer)

"We have used some, for example Rational Rose, for a period. The cost and benefit to use such a tool is clearly an element…" (Development manager)

The management sets requirements and specifications for the solutions, whereas the implementation takes place at the level of an individual developer. They consider experience of developers more valuable than explicit adherence to methods in the implementation and coding tasks. However, they plan to elaborate the already existing company-based practices for product specifications. (Table 4).

Scope of enacted practice	**Intended scope of defined practice**			
	None	**Individual**	**Project**	**Organization**
None				
Individual		Design, Implementation, Coding		
Project				
Organization				Iterative UI development, product and database specification, testing specification

Table 4: ISD practices in company B

4.1.3 Organization F

Organization F is a part of a multinational IT enterprise. It builds operative information systems and provides systems integration services. Its projects are typically large, business-critical, and involve multiple stakeholders and organization levels from the customer side. Many times the projects involve global aspects and international cooperation with many internal and external organizations.

The enterprise has built both an enterprise-level project model and an explicit and extensive in-house systems development method definition. At the enterprise level these are combined together, but in Organization F they are used quite differently. The project model describes project phases and deliverables and includes templates for most project documentation. The model was described as open and flexible and therefore it can be adapted to various kinds of projects:

"We have our own project model, which is actually common to the whole enterprise. […] and because it covers the whole enterprise, it is quite free. In practice, we must adapt it to each project situation. […] We use it practically in every project." (Manager)

The story of the extensive in-house method definition is somewhat different. The enterprise has at least twenty years of history of building in-house development methods, but the success of these efforts has been questioned. The developers and project managers see a number of problems in using the defined method. They feel that they should use the method, but on the other hand there are obstacles in using the method. Many regard the method as too complex and extensive and it is difficult to determine what to use in a particular development situation:

"In every situation you really have to think if you really need all those [features] like should you use [the method] as a whole or should you discard some of them and concentrate on the required things." (Project manager)

More importantly, the organization must occasionally adapt to customer processes and practices:

"In principle this [the method] is what we promote and try to get the customer committed to it, that is the way we should do the project ... in this project we have not succeeded. We have a customer too familiar wanting to do things the same way than we've done during the last twenty years." (Designer)

"Well I haven't used [the method] because I have been running an earlier project. I was in a very special project in Central Europe working at the customer side and then we had to follow what ever the customer had" (Project manager)

In addition, we interpreted that many developers lacked some required skills and knowledge to fully benefit from using the method. Because of that, not everyone was able to understand the benefits or use the defined method fully. Training was available, but it was the responsibility of business units and projects to finance the training. In a hectic project situation getting training was not easy to accomplish.

<table>
<tr><th rowspan="2">Scope of enacted practice</th><th colspan="4">Intended scope of defined practice</th></tr>
<tr><th>None</th><th>Individual</th><th>Project</th><th>Organization</th></tr>
<tr><td>None</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Individual</td><td></td><td></td><td></td><td rowspan="2">ISD method definition</td></tr>
<tr><td>Project</td><td></td><td></td><td></td></tr>
<tr><td>Organization</td><td></td><td></td><td></td><td>Project model</td></tr>
</table>

Table 5: ISD practices in company F

These two defined practices are located differently in NIPO grid (Table 5). The project model is both intended and enacted organization-wide, but the enterprise-wide in-house ISD method is used only by some projects and more commonly only by some individuals.

5 DISCUSSION

The following direct observations can be made from the data in light of the NIPO grid:

- The intended scope and an actually enacted practice can be different. For example, in company F the practices were defined at the level of the organization, whereas the individual projects often deviated from the defined guidelines.

- The scopes of intended and actually followed practices could vary inside one organization among different practices. For example, the coding/programming practices in organizations of A and B followed no specific practices while they had stricter guidelines and recommendations for contracting and system specifications, respectively.
- The scopes of intended and actually followed practices could vary across the organizations with regard to a certain area of practices. For example, in company F development practices were defined at the organization level (while varyingly followed), whereas the developers in companies A and B were rather free to decide about many of these individually.

These modest observations provided us a basis to cast a critical look at the previous literature of ISD practice. In the following, we will sketch the influential concepts of "method-in-action" (Fitzgerald et al. 2002), "amethodical systems development" (Truex et al. 2000) and CMM-based software process improvement (Humphrey 1989) in light of the construct.

In the method-in-action framework (Fitzgerald et al. 2002) a method-in-action refers the actual practice which takes place in a development project. The method-in-action may, but is not obliged to, be based on a formalized method which may be defined at the level of the organization or at the level of the project in question. It is uniquely enacted in each project by the developers. If we break the method-in-action down to the level of practices it can be seen as the collection of all manifest practices that have standardized enactment in the project (which include those possibly standardized and followed throughout the organization). If the method-in-action is based on a formalized method this would be captured by the definition of the practice referring to the formalized method. (Table 6).

Scope of enacted practice	**Intended scope of defined practice**			
	None	**Individual**	**Project**	**Organization**
None	Amethodical development			
Individual				
Project	Method-in-action			
Organization				

Table 6: Method-in-action and Amethodical development

Truex et al. (2000) introduce the idea of amethodical system development. In the amethodical view ISD is a random, opportunistic process, which is unique rather than replicable. A developer's actions will be guided by such factors as mood and social context, rather than logic, plan and method. Taken to the extreme, amethodical development actions are completely situated. In other words, enacted practices would not be standardized, and seemingly similar practices appear only coincidentally. Consequently, any decision to employ defined practices is irrelevant. (Table 6). For example, despite of their expclicit rejection of formal SDMs, organization A is not an amethodical organization as they still attempt to impose organization-wide practices in a few areas of development activities, such as contracting.

Software process improvement frameworks, such as the Capability Maturity Model (Paulk et al. 1991), provide yet another perspective on analyzing practice. They are designed to assess whether an organization is capable of repeating, or even improving, their performance from project to project. SPI builds on the assumption that certain pre-defined practices are desirable. An assessment of an organization is carried out by checking that the organization has defined certain key practices, and that they are enacted according to the definition.

An SPI assessment will focus on the recommended practices and their documentation. An SPI model can also include "meta-practices" for conscious deviation from the organization-level practice in selected projects, in order to try out new things and to improve the hitherto defined process. Used as a research tool this is a restriction. An analysis using our framework is more open, and thus more likely to uncover practices outside the recommended set, in particular undocumented practices (Table 7).

Scope of enacted practice	Intended scope of defined practice			
	None	**Individual**	**Project**	**Organization**
None	Missing definitions of practices		Unintended deviations from a project-level pilot practice.	Unintended deviations from the defined software processes in "standard projects"
Individual				
Project			Deliberate change of practice in a particular project to improve the existing development process definition, measurement	
Organization				Defined software processes, meta-practices for deviation in particular projects in order to improve the defined practice.

Table 7: Software process improvement

The examples indicate considerable variation in both definition and enactment of practices, which was not captured by the discussed frameworks. Moreover, none of the above-mentioned conceptual frameworks discuss about the individual dimensions of enacted practice, whereas the individual adoption of SDMs has been a focus of a few empirical studies (e.g. Stolterman 1992, Fitzgerald 1997). The NIPO grid now allows for investigation and discussion of these variations across a greater number of the dimensions than captured by the previous concepts. The method in action framework is built on the concepts of a method and a project. It does not take into account the practices which may be less clearly structured than a method or the individual variations in enactment of the practices. The amethodical development idea to an extent ignores the individual level of enactment, when contrasting itself to the project and organizational-level methods.

CMM and other SPI frameworks can be used to analyze IS development practices on a level of detail comparable with the NIPO construct. They also share the organizational focus, but differ by being designed as normative guidelines for improving practice. They build on the assumption that a certain set of key (or "best") practices should first be defined and then enacted according to the definition. It is outside the scope of an SPI assessment to detect emerging undefined practices. Individual practices and variations may be detected, but will be described as deviations.

Our construct and research have a few shortcomings which may require further attention. Firstly, our data collection method did not capture the nuances between individually (and sometimes unconsciously, cf. Argyris & Schön 1974) followed practices versus the totally non-defined situations. Hence, validation of the distinction between the columns of intended scope of individual practices vs. the non-intended column would require most likely observational methods in which human beings are asked to reason for their individual actions. However, such works as Humphrey's (1995) recommendations for "personal software processes" seem to assume that some defined practices would make a difference already at the individual level if compared to the non-defined situation, which legitimates these analytical dimensions at least in theory. Secondly, we found no evidence from our cases for the P-N, P-I, O-N and O-I slots, although they are theoretically possible e.g. in the method-in-action framework. Again, although being possible in theory (e.g. in cases where projects and organizations implicitly unify practices without being conscious about that), observation of emerging practices to be located in those slots would require more intensive data collection methods. Thirdly, there is no clear measurement line to decide when a practice is "truly" vs. "mainly" vs. "not at all" followed if it is defined at the organizational or the project level. However, we argue that our construct will now at least sensitize the researchers and practitioners to these issues.

6 CONCLUSIONS

The NIPO grid is a construct for describing patterns of systems development practices in organizations. The grid was constructed from empirical observations together with bringing in theoretical insight from the theory of action (Argyris & Schön 1974) and the systems development literature. We demonstrated the use of the NIPO grid with three case examples that showed the variation of intended and enacted systems development practices. In addition, we demonstrated in the discussion how the grid can be also used for discussing about literature approaches, such as "method-in-action" (Fitzgerald et al. 2002), "amethodical systems development" (Truex et al. 2000) and CMM-based software process improvement (Humphrey 1989). The grid integrates the previous approaches to a common framework.

In the future, we will continue the analysis and add more cross-case comparisons of patterns of practices in light of the NIPO grid. Moreover, we aim at grounding theories of explaining variation in particular types of practices among organizations. We expect that the NIPO grid will aid in clarifying organizational differences and similarities in adoption of systems development practices thus functioning as a dependent variable in such theorizing. The construct could perhaps also be used for explaining impact of practice standardization or non-standardization on systems development results. For practitioners, the grid may help understand the status of local practice repertoires.

References

Argyris, C. and D. Schön (1974). Theory in practice: Increasing professional effectiveness, Jossey-Bass, San Francisco.

Avison, D. E. and G. Fitzgerald (2003). Information Systems Development: Methodologies, Techniques and Tools 3rd ed., McGraw Hill Higher Education.

Collins CoBUILD (1989). English Dictionary.

Dijkstra, E. (1965). Programming considered as a human activity. In Yourdon, E. (1979) Classics in Software Engineering. Yourdon Press, New York.

Fitzgerald, B. (1997). The use of systems development methodologies in practice: a field study. Information Systems Journal 7(3): 201-212.

Fitzgerald, B. (1998). An empirical investigation into the adoption of systems development methodologies. Information & Management 34(6): 317-328.

Fitzgerald, B., N. L. Russo and T. O'Kane (2003). Software development method tailoring at Motorola. Communications of the ACM 46(4): 64-70.

Fitzgerald, B., N. L. Russo and E. Stolterman (2002). Information Systems Development: Methods in Action, McGraw-Hill Education.

Hardgrave, B. C., F. D. Davis and C. K. Riemenschneider (2003). Investigating Determinants of Software Developers' Intentions to Follow Methodologies. Journal of Management Information Systems 20(1): 123-151.

Humphrey, W. S. (1989). Managing the software process, Addison-Wesley Longman Publishing Co., Inc.

Humphrey, W. S., T. R. Snyder and R. R. Willis (1991). Software Process Improvement at Hughes Aircraft. IEEE Software 8(4): 11-23.

Humphrey, W. S. (1995). Introducing the personal software process. Annals of Software Engineering 1(1): 311-325.

Iivari, J., R. Hirschheim and H. K. Klein (1998). A Paradigmatic Analysis Contrasting Information Systems Development Approaches and Methodologies. Information Systems Research 9(2): 164-193.

Iivari, J., R. Hirschheim and H. K. Klein (1999). Beyond methodologies: keeping up with information systems development approaches through dynamic classification. Proceedings of the 32nd Annual Hawaii International Conference on System Sciences.

Iivari, J. and K. Lyytinen (1998). Research on Information Systems Development in Scandinavia: Unity in Plurality. Scandinavian Journal of Information Systems 10(1&2): 135-186.

Jarke, M., K. Pohl, C. Rolland and J.-R. Schmitt (1994). Experience-based method evaluation and improvement: A process modelling approach. Proceedings of the IFIP WG8.1 Working Conference on Methods and Associated Tools for the Information Systems Life Cycle., Elsevier Science Inc.

Jayaratna, N. (1994). Understanding and Evaluating Methodologies: NIMSAD, a Systematic Framework, McGraw-Hill, Inc.

Kautz, K. (2004). The enactment of methodology – the case of developing a multimedia information system. Proceedings of the International Conference on Information Systems. Washington, DC, USA.

Khalifa, M. and J. M. Verner (2000). Drivers for software development method usage. IEEE Transactions on Engineering Management 47(3): 360-369.

Lyytinen, K. (1987). A taxonomic perspective of information systems development: theoretical constructs and recommendations. Critical issues in information systems research, John Wiley & Sons, Inc.**:** 3-41.

Madsen, S., K. Kautz and R. Vidgen (2006). A framework for understanding how a unique and local IS development method emerges in practice. European Journal of Information Systems 15(2): 225-238.

Mathiassen, L. (1998). Reflective Systems Development. Scandinavian Journal of Information Systems 10(1&2): 67-118.

Mathiassen, L. and S. Purao (2002). Educating reflective systems developers. Information Systems Journal 12(2): 81-102.

Mathiassen, L. and L. Vogelsang (2005). The Role of Networks and Networking in Bringing Software Methods to Practice. Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS), IEEE: Los Alamitos.

Parnas, D. (1972). On the criteria to be used in decomposing systems into modules. Communications of the ACM, 15 (12): 1053-1058.

Paulk, M. C., B. Curtis, M. B. Chrissis and C. V. Weber (1991). Capability Maturity Model for Software, Software Engineering Institute.

Päivärinta, T., M. Sein and T. Peltola (2008). From Ideals towards Practice: Paradigmatic Mismatches and Drifts in Method Deployment. Information Systems Journal (in press, doi: 10.1111/j.1365-2575.2007.00256.x).

Stolterman, E. (1992). How system designers think about design and methods - Some Reflections Based on an Interview Study. Scandinavian Journal of Information Systems 3: 137-150.

Strauss, A. L. and J. M. Corbin (1990). Basics of qualitative research: grounded theory procedures and techniques., Sage.

Tolvanen, J.-P. (1998). Incremental Method Engineering with Modeling Tools.Jyväskylä Studies in Computer Science, Economics and Statistics 47, University of Jyväskylä.

Truex, D., R. Baskerville and J. Travis (2000). Amethodical systems development: the deferred meaning of systems development methods. Accounting, Management and Information Technologies 10(1): 53-79.

Vogelsang, L. and F. Kensing (2006). Utilizing systems development methods - a coneptual framework. Proceedings of the Fourteenth European Conference on Information Systems. Gothenburg.