Episodic organizational learning in system development

Ia Williamsson
Department of Information Technology, University of Borås, Borås, Sweden, and

Linda Askenäs
Department of Informatics, Linnaeus University, Vaxjo, Sweden and
Department of Information Technology, University of Borås, Borås, Sweden

Abstract

Purpose – This study aims to understand how practitioners use their insights in software development models to share experiences within and between organizations.

Design/methodology/approach – This is a qualitative study of practitioners in software development projects, in large-, medium- or small-size businesses. It analyzes interview material in three-step iterations to understand reflexive practice when using software development models.

Findings – The study shows how work processes are based on team members’ experiences and common views. This study highlights the challenges of organizational learning in system development projects. Current practice is unreflective, habitual and lacks systematic ways to address recurring problems and share information within and between organizations. Learning is episodic and sporadic. Knowledge from previous experience is individual not organizational.

Originality/value – Software development teams and organizations tend to learn about, and adopt, software development models episodically. This research expands understanding of how organizational learning takes place within and between organizations with practitioners who participate in teams. Learnings show the potential for further research to determine how new curriculums might be formed for teaching software development model improvements.

Keywords Software development model, Organizational learning, Reflection on action

Paper type Research paper

Introduction

IT systems complexity increases constantly, and organizations must constantly develop and modify IT systems. To develop purposeful IT systems, different professions must cooperate. They face continuous unpredictable changes and challenges, so they must learn new practices and adapt to new circumstances. Demands on how the team should cooperate come from different directions and to different degrees. Therefore, it is important to constantly consider and develop (model) the IT systems development process. Many different models exist. Some are detailed, to be used strictly to control every step of the systems development life cycle. Other models do not claim full control. Certain models are more general and can be adapted and extended based on needs (Sommerville, 2016; Avison & Fitzgerald, 2006).
To achieve high-quality products (software), models are important. Research shows that improving models leads to improved productivity and more efficient development projects (see for instance Clarke & O’Connor, 2013). However, more work must be done to improve model maturity and improve reflective work practices within and between organizations, regarding the use and modification of models.

Ever since the “Agile Manifesto” (Fowler & Highsmith, 2001) was formulated, agile models are extensively used and continuously improved. However, despite extensive research, we are still finding out how to preserve knowledge and experience. For instance, Giachetti, Luis de la Vara, & Marín (2023, p. 1) argue that “it would be valuable to count on appropriate support to preserve and systematically use this expert knowledge”.

Who is in charge of learning about and knowing the model: the team, the manager or the model itself? For example, Fagarasan, Popa, Pisla, & Cristea, (2021) recommend organizations to start using a model (agile or waterfall) either by “following the methodology’s principles and adopting the agile mindset at the organization level” or “to blend the traditional approaches that made the organization successful to date with the Agile principles”. Simultaneously, they argue that managers can make use of extensively documented know-how, experiences from the software development project model. The question is whether this is actually done, and whether the shared experience is of value, or is too situated in a particular team’s collective learning.

This study aims to understand how practitioners share experiences within and between organizations by using their insights from software development models use.

Organizational learning should diffuse the individual’s knowledge in the organization, giving others access to it, even if individuals leave the organization (Argote, 2011). Individuals in organizations learn by reflection-in-action or reflection-on-action (cf. Schön, 1983). However, for organizational learning to work, the sum of an individual’s learning is not merely the sum of all individuals’ learning (Fiol & Lyles, 1985). Understanding the learning processes for spreading knowledge between individuals has a major impact on the organization’s ability to achieve organizational learning (Senge, 2006). Another perspective is that the knowledge is more situated and bounded in collective learning and triggered by contradicting views rather than harmonization of experiences (Engeström & Sannino, 2010). Instead of seeing contradictions as something bad in software development projects, they can be used to reflect on and learn about new ways of working in the activity system (Dennehy & Conboy, 2017).

Understanding how and when models, and experiences of using models, are spread among software development projects is essential for problematizing cooperation, productivity, continuous learning, risk management and changes among team member constellations. We believe that understanding these issues can enable teams to leverage collective expertise between organizations, make informed decisions and adapt to the evolving demands of software development.

Research questions

**RQ1.** How do practitioners in software development projects learn and share knowledge about system development models?

**RQ2.** How does collective learning in software development projects give rise to organizational learning?
In this article, we start by discussing different approaches to organizational learning and their relevance to software development. We present the organizational learning perspective, the analytical model used and the study methodology. Then, we introduce our empirical work and discuss the findings.

Software development process models
Software development processes models refer to the methodologies, frameworks and practices used to create software applications or systems. This encompasses the entire life cycle of software development, from gathering requirements to designing, coding, testing, deployment and maintenance. Various models exist, including waterfall, agile, V-model, spiral model and DevOps, each with its own approach and set of principles. The goal of a model is to ensure efficient and effective development, delivery, maintenance of software products and for achieving a common conceptual view (Brown, 1997; Andersen, Helleskog, & Forsberg, 1994; Bubenko, 2007).

Before agile models evolved, research was unprepared for rapid changes and could not anticipate the spread of agile methods (Ryan, 2016). It has now evolved to be about monitoring software development organizations to provide insights and recommendations to help the organization adapt and improve its practices and about overcoming challenges that new technologies entail (e.g. AI, machine learning) and streamlining the development process (see for instance Ochodek, Staron, Meding, & Bosch, 2020; Lwakatare, Crnkovic, & Bosch, 2020). For that, it is necessary to learn and understand the software development process, no matter the type, size or approach. This study does not focus on the model per se, but on how it is learnt, in the organization where it is used.

The organizational learning approach
In organizational learning, knowledge in software development projects may be represented in documentation, people, business processes, working routines and transactive memory systems. However, there are few studies of organizational learning in software development projects, and these usually focus on few areas of software engineering and organizational learning concepts (Menolli, Reinehr, & Malucelli, 2013). A new project, new project members, new requirements from business and/or customers lead to demands for learning new ways of working, but “there is no ‘one-fits-all’ approach, and knowledge and experience play an important role when trying to adopt the approach” (Schaefer et al., 2012, p. 488). There is an embedded tension or dilemma in applying generic models in situated projects (cf. Virkkunen, 2006) or even use experiences from one project to another both within and between organizations.

Learning systems and reflective practice
Software development projects will probably continue to increase in number and scope. The creation, re-creation and transfer of model knowledge introduces challenges for the future. A new generation of workers changes jobs more often, requiring that we learn how to manage the learning processes.

Each software development project in this research uses a model providing a common way of working. The project members must engage collectively in the activity system and effectively transform the way they work. There are therefore two knowledge aspects to consider: partly, tacit knowledge from experience of model use; also, knowledge of the development process and how the model develops in the specific situation. This is referred to as situated learning (cf. Lave & Wenger, 1991) and is a probable future concern for development of sustainable learning about models. There is also an
embedded tension in the individuals’ experiences from different projects within and without organizations and the collective learning process. The dilemma is about whose experience to learn from within the team. We know knowledge resides in the activity systems (ASs) of the substantive area and is created, saved and reused among individuals, ASs and projects, and thus becoming fragmented knowledge, what DiSessa (1988) calls “knowledge in pieces”. Those “systems of fragmented knowledge” consist of several “learning settings” encompassing people, symbols and technologies that together “construct and reconstruct understanding of social and organizational action” (Bruni, Gherardi, & Parolin, 2007, p. 83).

The substantive area in this research has several ASs: different organizations, individuals in an organization; teams; the organization itself. In the case of consulting firms, the organization also has constellations of people in projects, where other organizations sometimes participate (Hansen, Nohria, & Tierney, 1999). Transfer of knowledge between ASs is important and impacts project outcomes. However, the effectiveness of knowledge transfer is an ongoing discussion (Erden et al., 2008), and literature on tacit knowledge management and organizational learning theory is usually at the individual level (Gourlay, 2006). Yet, the organization must handle building blocks for effective organizational learning (Garvin, 1993; Basten & Haamann, 2018). This reveals also a tension in the organizational level to gain and make use of experience from multiple projects, to gain new knowledge from outside the organization and still running the particular project in the best way (cf. Virkkunen, 2006). The dilemma is to understand which experiences to gain from and to understand if it will be worth the effort for the organization to establish new models. One way to understand organizational learning in practice is by reflection, either during or after the project (Schön, 1991; Høyrup, 2004). We also know that the future means new situations and increased expectations for software development projects. How individuals learn and the variety of sources they learn grows constantly. Therefore, it is important to consider the dynamics of learning and the depreciation of knowledge (Argote, Beckman, & Epple, 1990), particularly as individuals change jobs at an increasing rate, in a global market, where projects are common.

We have argued that there are different dilemmas to consider when it comes to understanding the use of models in system development projects. One theory that focus on learning and particular expansive learning to occur is CHAT. CHAT offers a philosophical and cross-disciplinary perspective that is used as both experimental and analytic method in different settings and to elaborate with different units of analysis (as in our case individuals, teams, organizations and between organizations). The theory has been used in software development contexts (Vakkayil, 2010; Chita, 2018; Chita, Cruickshank, Smith, & Richards, 2020). We will use CHAT as a frame of analysis by revealing the ASs new models enters and express potential contradictions and tensions that give rise to dilemmas that either enhance or restrict learning on the organizational level.

**Cultural–historical activity theory – CHAT**

The history of activity theory or cultural–historical activity theory (CHAT) is characterized by the contributions of various researchers, including Engeström, who built upon Vygotsky’s ideas of how social and cultural factors shape human activity and learning. Activity theory continues to be a valuable framework for studying and understanding the complex interactions between individuals, their social and cultural contexts and activities in which they are embedded (see e.g. Engeström, 2001; Engeström & Sannino, 2010). Activity theory emphasizes the importance of collective and collaborative knowledge creation, where individuals with different backgrounds and
perspectives come together to solve complex problems and develop new ways of working. Engeström bases his definition of knowledge on Nonaka & Takeuchi (1995) but describes it as a collective and dynamic process emerging from social interactions and participation in complex work activities. In his view, knowledge is not a static object, to be transferred from one person to another, but rather a social and cultural phenomenon shaped by the context and the practices in which it emerges. This approach to knowledge recognizes the importance of context and the situatedness of knowledge and emphasizes the role of social interaction and collaboration in shaping knowledge and learning. Crawford & Hasan (2006) claim that it is useful in dynamic situations where people, their objects and tools are constantly changing.

The AS in which learning occurs can also be interconnected. In our empirical work in software development project there are often collaboration with different organization and the project team can consist of members from different organizations. Activity theory in this case is used in three different analytical levels. One level is the project team itself as they work and learn together. As a second level, it is the organization (AS) where the team is located. It can also be seen as a collaboration between different ASs working together within a project. If we discuss, as we do in this study, also the use and transfer of system development models between organizations we have a third level. The unit of analysis on this level focus more on the processual relationships than the structure or elements of a specific AS (cf. The fourth generation of activity theory in Engeström, 2020). To investigate the connections where learning happens, the unit of analysis was the learning and use of system development models. Models used and learned about on the team level, in different AS and in-between AS, see Figure 1. By using activity theory and also aspects of expansive learning theory (cf. Engeström, 2001, originating from Vygotsky & Cole, 1978) on different levels we were able to visualize and analyze the knowledge transfer or absence of knowledge transfer in the multitude of possible scenarios.

![Figure 1. The substantive area of the study and its AS of episodic organizational learning (cf. Engeström, 2001)](image)

**Source:** Developed by the authors
Expansive learning

Expansive learning is a process of acquiring new knowledge and skills that go beyond what we currently know and do. It implies moving beyond existing boundaries, exploring new ways of thinking and acting. When practitioners learn, they explore and seek new combinations of experiences and models to use in new contexts, challenge their current understanding and push themselves to think in different ways (Engeström & Sannino, 2010; Engeström, 2015; Engeström, 2020). The expansive learning cycle challenge and question important parts of existing knowledge and explore new perspectives. It is also about experimenting and reflecting on what is learned. Boundary crossing refers to crossing boundaries or barriers that separate contexts, disciplines or perspectives (Engeström & Sannino, 2010). It involves leaving our comfort zones, engaging with ideas, knowledge and people from outside our usual boundaries. Crossing boundaries goes hand in hand with expansive learning. It exposes us to new perspectives and knowledge, promoting expansive learning by expanding our understanding, enhancing problem-solving abilities and promoting adaptability.

To understand learning approaches in the substantive area, we use the "matrix for the analysis of expansive learning" (e.g. Engeström, 2001, p. 152). Activity theory and expansive learning theory can be explained by four questions about learning and five principles of activity theory; AS, multi-voicedness of activity, historicity of activity, contradictions and expansive cycles, where the AS is subject for analysis, see Table 2. In our case the learning and use of model is the unit of analysis, still it is focused on how it is used and learned in and between AS.

Study method

Our aim was not to follow the use or implementation of specific models or a specific AS, but to understand the processual relationship how models were diffused and learned between different AS. In our case, ASs were represented by different organizations that had been involved in system development projects. To understand how different ASs used and learned about models, we focused on how different teams initiated (e.g. by learning from others) and used new models or experiences, but also if it had an impression or not to give rise to expansive learning on the organizational level. We interviewed experienced system developers and project leaders who had been active in multiple projects, who could reflect on projects and organizations where they had learnt new models. We did not intend to generalize or quantify but rather to gain an understanding of different ways models could be integrated into an AS.

The data collection

The study is based on a qualitative interview study of nine practitioners from large-, medium- and small-size organizations (see Table 1). The selected practitioners were managers with previous and present experience of responsibility for work procedures and models, and all had extensive experience of model development and use. Two participants (No. 8–9 in Table 1) were both academics and practitioners (consultants). The interviews were carried out in Sweden, in Swedish companies or units of an international enterprise. The length of each interview was between 1 and 1.5 hours. The substantive area in this study is: the experience of practitioners’ use of models. This paper builds on data collected in an explorative interview study from 2017 (X), on the development and use of software development processes models. The initial aim of the study was to investigate how models are developed and diffused between academia and practice.
This paper uses a hermeneutic approach to empirical data, using the hermeneutic circle as the underlying analytical framework (Cole & Avison, 2007; Klein & Myers, 1999). By first examining the meanings of individual parts and then integrating them to comprehend the studied phenomenon. The analytical investigation of the data is operationalized through three key steps: understanding, explanation and interpretation, which facilitate the development of connections between the parts and the whole. The following section details these steps and describes the circular analysis of the data.

First, the data were reinterpreted. The first author had conducted the interviews with open-ended questions about which models participants used, how they are inspired by others and whether they prefer to use models from research. The material was rich enough to discuss whether models established elsewhere could give rise to organizational learning. The second author read the transcripts and together the authors drew mind-maps from each of the nine interviews. The mind-maps were used as an initial artefact to understand and explain the learning of models in several AS, and between individuals, teams and organizations, as illustrated in the findings section.

Second, for a better understanding of the field, theories of organizational learning and reflection in practice were consulted (Senge, 2006; Schön, 1991; Høyrup, 2004; Garvin, 1993).

### Table 1.
The informants of the study experiences

<table>
<thead>
<tr>
<th>No.</th>
<th>Informant</th>
<th>Years of work</th>
<th>Type of business, sector, size of business</th>
<th>Model need and experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Product line manager</td>
<td>&gt; 20</td>
<td>Finance and payroll software company, 12,500 employees (emp). The development department, 500 emp</td>
<td>Governance of legacy products; develop and modify models for different businesses and teams</td>
</tr>
<tr>
<td>2</td>
<td>Business development manager</td>
<td>&gt;10</td>
<td>IT consulting, corporation, 5,000 emp. Development and management of intranets, 200 emp</td>
<td>Work according to customer requirements and modify models</td>
</tr>
<tr>
<td>3</td>
<td>Solution architect</td>
<td>&gt;10</td>
<td>Management and IT consulting company, &gt;20,000 emp. System development and management for customers, 50 emp</td>
<td>Used different models for different types of projects; developed own models and modifying models</td>
</tr>
<tr>
<td>4</td>
<td>Section manager</td>
<td>5</td>
<td>Government authority organization, approx. 2,000 emp, IT department, &gt; 40 emp</td>
<td>IT development and governance; usually working with the same simplified model. models</td>
</tr>
<tr>
<td>5</td>
<td>Tech. project manager</td>
<td>3</td>
<td>IT consulting, approx. 30 emp, IT system maintenance and operations</td>
<td>Used modified models for common view of how to implement and for communication</td>
</tr>
<tr>
<td>6</td>
<td>Service delivery manager</td>
<td>&gt;10</td>
<td>Large international corporation, &gt; 15,000 emp in Sweden, IT department, approx. 350 emp</td>
<td>Used different models for different types of projects, develop models together with consultant, buy model from vendors</td>
</tr>
<tr>
<td>7</td>
<td>IT manager</td>
<td>&gt; 20</td>
<td>Municipality in Sweden, approx. 7,000 emp, IT unit, approx. 40 emp</td>
<td>Use modified models for operation, development, maintenance and support of the municipality’s IT environment</td>
</tr>
<tr>
<td>8</td>
<td>Project manager, researcher</td>
<td>&gt; 20</td>
<td>International software engineering company 130,000 emp. Research institute approx. 1,000 emp, 9 emp. in Sweden</td>
<td>Used different models for different types of projects where teams are large and international. Develops models, modifying models</td>
</tr>
<tr>
<td>9</td>
<td>Researcher, consultant</td>
<td>&gt;21</td>
<td>IT consulting, small company. Research institute &gt; 500 emp</td>
<td>Develops models for other companies</td>
</tr>
</tbody>
</table>

*Source:* Developed by the authors
Table 2. Matrix for the analysis of expansive learning (cf. Engeström, 2001)

<table>
<thead>
<tr>
<th>As a unit of analysis</th>
<th>Multi-voicedness</th>
<th>Historicity</th>
<th>Contradictions</th>
<th>Expansive cycles?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who are learning?</td>
<td>Multiple requirements: Developers, end users, buyers, colleagues, suppliers, etc</td>
<td>System developers</td>
<td>Individual – team</td>
<td>Team</td>
</tr>
</tbody>
</table>
| Why do they learn?    | New situations (projects) that shall meet the requirements of the customer      | Different ways of doing system development, e.g. waterfall and agile methods | *Common work practices – freedom to choose  
*To learn for future possibilities – learn for the present  
*Customers need – habits  
*What works satisfactorily – uncertainly in new ways  
*Method – product  
*Simplicity – complexity | *Old habits rules  
*New ways of working are forced by circumstances |
| What do they learn?   | *Requirements and needs  
*How to collaborate and manage the project  
*Common concepts and procedures  
*Technology circumstances | *To work together  
*Team habits and routines  
*To focus on project results  
*Focus on requirements from costumers  
*Models and procedures and tools  
*Technology  
*Theories in practice  
*Models, methods and tools  
*How to adjust theory to practice | *Learning from experience of former projects and what works in practice  
*Management established models  
*Education programs | *Cumulative experiences  
*New people, customers, ways of working etc.  
*How to adjust working to new circumstances |
| How do they learn?    | *By building from experiences  
*Trial and error  
*Active searching internet  
*From experts  
*By doing | *Learn from each other’s perspectives  
*Discuss to consensus.  
*External experts or internal guidance  
*Test and try different approaches | *Time to learn – time to produce  
*Reflection in action – on action | *Readjustments in common knowledge  
*In episodic manner enforced by new postulates |

Source: Developed by the authors
Gaps in organizational learning strategy were found, as it was unclear how, when or which knowledge or experience of models were used in system development projects. When re-reading the interview transcripts, new patterns were found, adding to the previous findings. Three things stood out:

1. there seemed to be very few cases of system development projects that actually included reflection upon their experience of models;
2. new models were developed “ad hoc” and had a sparse use of learning strategies; and
3. knowledge of models exists, but it is unclear where it originates from.

In the third step of interpretation, we wanted a clearer understanding of the learning processes. The findings showed that the practitioners had experience and knowledge of how to manage system development projects and were confident when using and modifying models together with their team. The primary focus was shown to be achieving a common view of how to work, within the team. To grasp the team-level collective learning revealed in the data (which previous theoretical lenses did not emphasize), we used theoretical lenses from activity and expansive learning theory (Engeström & Sannino, 2010; Crawford & Hasan, 2006; Engeström & Sannino, 2010). To systematize the discussion of AS and expansive learning, we take inspiration from the AS model (Figure 1) and the matrix (Table 2) according to Engeström (2001) and Engeström & Sannino (2010) known as third-generation activity theory. The matrix was used to understand learning approaches in the substantive area, which we rephrase as the AS. Also, we added the concept of fragmented learning, to understand the use/non-use of parts of models. We identified commonalities, patterns and generalizable aspects from the project teams, thereby analyzing it as one AS (the utilization of more generic models in situated practice) with people from different organizations. This gave insights in challenges of organizational learning that can be applicable in different organizational contexts. Another methodological challenge was that we observed historical events in different organizations from the perspective of one participant rather than following one setting and observing the expansive learning circle over time with multiple participants (cf. Engeström, 2001, Haapasaari, Engestrom, & Kerosuo, 2018). However, by using the lenses of activity theory we could expand the interpretation of the data to deepen our understanding of the phenomena, using the theory to expand the view of the practitioners. That helped us to “illuminate and articulate what generally goes unnoticed because it is ubiquitous, common-place, and everyday” (Cole & Avison, 2007, p. 821).

Integrating expansive learning theory and activity theory led to a comprehensive and practical framework for understanding the complexities of human activities. We gained a deeper understanding of: how learning and development occur within and between ASs; the transformative potential of learning new models; the significance of collaboration and social interaction when learning new models; the importance of addressing contradictions and conflicts of understanding when expansive learning occurs; and the “dormant” potential that resides in the individual’s ability that can lead to expansive learning in coming activities.

Findings
Here, we present the findings from the interviews. The text is structured under four headings, following the four questions of expansive learning.
TLO

Who are learning?
The informants of this study represent a team (AS). Together with their team members, they represent a unit of analysis. Beside the project manager, the AS consist of developers, customers and suppliers. All informants are project managers and must ensure that the AS knows how to collaborate in an efficient way. Thus, the manager needs knowledge of model requirements, common model concepts and working procedures and other important technological aspects:

Initially, a discussion takes place with a couple of people at the company, then a vision of a solution is sold in. Then decisions are made by a group of managers, and engineers who have often shouted for a solution to the problem, as well as knowledge of […] (Informant: 9).

The AS must also adapt to new situations, such as new projects with new team members or new constellations of members within the organization or new customers. Project managers and team members learn this through experience and trying new working routines. Sometimes successfully and sometimes not. When the AS requires a new model, the project manager or a member of the AS, searches the internet for inspiration and facts about models.

Why do they learn?
All practitioners in the study, with the exception of two informants who have worked for 20 years or more, state that they use models to support their work process. These two exceptions explain that although they did not initially use models, they gradually began using models to structure their work. Important reasons for using models are to have a common view of how to implement software, use concepts, follow procedures and describe how to communicate with team members during a software development project. If software developers talk in the same terms, it becomes easier to jump into new projects and teams:

I think models are really important […] to keep track of that we all work in a similar way. We use many diverse types of models, depending on what it is we are doing. It is easy to jump into another project (Informant: 6).

In some cases, the team itself, clinging to old habits, can be a barrier to using new models. They may be unwilling to learn how to work according to a new model or do time-consuming documentation. Sometimes new ways of working are forced upon the team by customer requirements or management. The team must learn new ways of working, which may be difficult to operationalize, and people can fall back into previous ways of working, “what we usually do” or “this is our way”:

It can be a scourge to use a model if it describes that you should make a lot of time-consuming reports that none cares about, that just generates a lot of extra work for no benefit. There is much to say about the models, but most concocts to the experience of individual’s skills and abilities (Informant: 5).

What do they learn?
In this study the number of practitioners who developed new models for a software development project and those who adopted existing models are almost equal. The practitioners looked at existing models, took useful parts from them and used them as a foundation for the new model. In one case the practitioner claimed that they always start a new project (or when a new model is needed) by investigating if someone has done something similar, before starting to develop or modify a model:
One does not need to reinvent the wheel, but to follow what someone else has already developed (Informant: 4).

Concerning model modification, practitioners describe how models are adjusted to fit their purpose. For instance, in content and language, or to be “the way we are used to working”. Models can also be modified according to customer requirements, or by adding specific procedures and processes for what should be delivered (e.g. a server, application support, quality of IT). They also describe how models are used “by the book” in the beginning, but that gradually some parts are removed to make the model smaller and more suitable:

Using what we believe are good parts of the general model. Sometimes the customer has requirements of the model and we work accordingly (Informant: 2).

Most practitioners name evidence as the reason they do not try to learn from research or models in academic research. They require evidence from practice that the model works well in similar practice or that the model is standard practice. Another explanation for not collaborating with academia is that practitioners do not have the time and money to collaborate with researchers. The two consultants believe that models are useful to visualize and communicate phases and other issues in the development process and that learning from academic research is beneficial:

It can be beneficial to ‘bring home something academically from time to time’, if ‘something scientific’ is needed to convince people and get some inspiration and examples (Informant: 8).

There is no tradition of practitioners applying academic theories in their daily work. Instead, they take models, procedures and tools from practice. If they must learn to work according to a new model that someone in the organization has heard about, and to overcome uncertainty and give inspiration, a lecturer, who has experience in using the model in practice gives basic knowledge of it. However, the organization, individuals in the team and customers is new to the lecturer, and in most cases some of the individuals are new to the team:

It’s good to have someone show that it works in other companies that are similar (Informant: 1).

Most practitioners describe how principles guiding the choice of model are connected to people in the organization or the team who will work with the model or are interested in learning about models. If the practitioner is a consultant, models are chosen at the initiative of the customer or the consultant. Decisions about whether the organization should use a new model are made by either top management, an individual or in the project team. This is followed by further discussions in the team on how the model should be used. In few cases the model experience was reflected upon, collective or individual. Rather, most new models were developed \textit{ad hoc} and had sparse use of learning strategies. Only one practitioner, a consultant, used academic theories and experience from practice when developing a new model to be used by a team.

\textit{How do they learn?}

The freedom to select a new model is extensive and depends on the practitioner’s habits and experience. Other individuals in the team also bring their experiences to the model selection. Whether or not practitioners are trying to learn about models in research depends on the team developing the system, and if they have an interest in finding new knowledge and learning from academia. However, they may not have thought at all about using academic knowledge in projects:
We have quite a lot of freedom to choose models and it is often different from project to project. It depends largely on the individual’s interest. (Informant: 3).

All practitioners studied computer science at university level, where models are included in the systems development syllabus, but none of them mentioned this knowledge as an obvious and useful starting point. There is no reflection about the origins of models or lessons learned that may be used in following projects. On the contrary, to maintain a good model and make use of common model knowledge, it is seen as crucial for the team and organization to continuously communicate and reflect on the model. So, knowledge about new models consists of what is found on the internet. Internet is also used to benchmark and see what other organizations do. To find new ways of working, organizations also use the experience and knowledge of their employees, as it is seen as everyone’s task to “know your field”. Here, the search is expanded to include fairs, lectures, newspapers, Internet and networking:

It can be hard to find out what is available. We google and talk with colleagues. We believe that we know what’s available in our area. We think so anyway. Some people bring the models, methods and practices from previous work (Informant: 7).

Concerning the possibilities of finding new knowledge, practitioners say that looking for research or models from academia would require seeing something interesting to learn or receiving research that is packed differently than it is now. Suggestions for “packaging” were for instance receiving an email containing targeted information about a chosen topic or incorporating the development and testing of models in academic education:

I would need to receive a mail in order to open my eyes to interesting new research findings, and also that we at that time have the need for it (Informant: 6).

Few practitioners say they have previous experience of cooperating with academia for learning purposes. One case concerned finding new model aspects. In another case, they say that they might have used research unexpectedly. For practitioners to look for new knowledge in academia, would require being in a project where there is a need to look at models that may not yet be available, or being in a group working “closely with the academic world”. To learn from academia requires well-founded cooperation and to achieve a functioning knowledge transfer requires time. The practitioners that never searched for new or improved models in academia have instead learnt about other projects from their networks or by searching on Google:

We look at practical experience, it can be your own or if you work with someone who has done something new or has visited a customer. We work according to how the customer wants it because we are a consulting company (Informant: 2).

In some organizations, academics in IT departments are said to contribute with a structured mind-set. However, including academics in an industrial setting in software development projects is in an immature state. Practitioners mean that there is “a lot left to do” and that missing knowledge exists in academia which could be added to the models.

Analysis and discussion

Learning helps us understand how we should work together, and in an organizational context, there are strategies to manage learning in teams. This research focuses on how the team is introduced to and incorporates new knowledge and experience, which becomes common knowledge, shared directly between people (face to face) and not saved in any (digital) format for further use (e.g. Hansen et al., 1999). Software projects must often gather
model information and knowledge from various competences inside and outside the team and organization. The knowledge gained from experience in using models is thus not explicit and only partly remains in the organization after the project ends. Findings show that reorganizing a team into new teams requires a learning process, and experiences must be shared, and agreements reached for the new working processes. To be able to work efficiently toward the goal of each project, they must agree on how to collaborate, on work routines and the meaning of different concepts, where the goal is the software product. In doing so, they learn from common and shared individual experiences, external searching and just by “start doing”. Reflection is performed in the process instantly, thus reflection-in-action (Schön, 1983). Insights and situated knowledge reside in people and appear in project situations when appropriate, transformed to the new project or shared on the team level. The experience and knowledge from model use is thus mainly stored as tacit knowledge that follows the person from one project or organization to another. Even if tacit knowledge is viewed as the root of all knowledge (Nonaka & Takeuchi, 1995), complex learning processes are necessary for it to become knowledge that can be used in other settings. To understand the complex learning process, we use activity theory and the matrix for understanding expansive learning (Engeström, 2001; and Engeström & Sannino, 2010).

**Multi-voicedness**

Many perspectives must be considered when starting a project represented in fragmented knowledge, such as different knowledge repositories, experiences, models, standards and artifacts (Bruni et al., 2007; DiSessa, 1988). These perspectives are handled different in each organization. There is a need to learn and consolidate fragmented knowledge throughout the project, in what the individual’s emphasize and to capture customers’ requirements. There are also embedded practices from different actors, inscriptions in explicit models and tools. The practitioners talk about the need to achieve satisfactory coherence on how to work. For a new task or project, they may need to find new knowledge outside the team, their AS and use other colleague’s experience, external experts, content from internet searches. Thereby, benchmarking and finding “best practice” and learning from others, as one of the building blocks for organizational learning (Garvin, 1993; Basten & Haamann, 2018). They seem to learn by starting to do and then adapting, but the benchmarking work and experimentation with new models is not systematic, neither is knowledge from experience from previous projects. Again, this does not correspond to effectiveness in organizational learning and systematic problem solving, rather they are ad-hoc efforts (Garvin, 1993; Basten & Haamann, 2018).

**Historicity**

Models and methods for systems development focused first on good quality code, and possibilities to reuse code (Brown, 1997; Munassar & Govardhan, 2010). Software development models are continuously developed in academia but evidence from use in practice is vague (Zannier, Melnik, & Maurer, 2006; Sjöberg et al., 2005). Many models exist to support and guide system development, with origins and improvement strategies from both research and practice (Bellini & Storto, 2006). System developers have for many years worked freely, concerning collaboration and participation with networks. The practitioners in this study tell the same story, that they have had extensive freedom to choose or adjust models to particular projects, or to what they were comfortable with. The building blocks for effective organizational learning are lacking in their descriptions of their work of finding and/or developing new models (Garvin, 1993; Basten & Haamann, 2018). When reflecting on previously used models, most of them name experience from practice, from other
practitioners, and perhaps from parts of established theory, or from their education. From a theoretical perspective, this demonstrates that reflection-in-action may sometimes be done intuitively, non-systematically, but not through reflection-on-action as retrospective evaluation (Schön, 1983). There is also the challenge previously discussed by Fuggetta and Di Nitto (2014) about quality assurance and external validity of the models. For transfer of expertise and knowledge between practice and academia, they suggest long-term relationships between academia and practice.

**Contradictions**

There are various contradictions, whereof one concerns “who is learning?”; the practitioner or the team. Successful project management involves team cooperation, finding common ground for working within a model. A precondition for organizational learning is to ensure that experiences are spread to the work system, getting as close as possible to best practice (Bellini & Storto, 2006). In this study, the strategy for learning from others is through benchmarking similar projects to find best practice, but not in combination with a systematic approach or experiments and not by consciously reflecting-on experiences that were possibly saved and accessible. They seem to consider some of the five building blocks suggested by Garvin (1993) but not consciously or systematically. Individual experiences of models are used but must be accepted and understood by all in the AS. The importance of communicating is emphasized, to complete the task with a satisfied customer, but what must be learned and why? Here, the major effort is to find good practice to enable solving the team’s task rather than to find the optimal way of working.

There is also a contradiction in the lack of systematic efforts to documenting work routines, experiences and/or use a transactive memory system. To succeed in using a new model, knowledge and experience are necessary (Schaefer et al., 2012), but knowledge and experience are not made explicit or implicit for organizational learning purposes.

Further, there is a contradiction in the intention to use a generic model for a specific project, whereas in all situations models require modification. For a complex project (with several stakeholders and a large project) the model cannot be complex, since all individuals need to understand and use it. On the other hand, a small and short project cannot afford an exhaustive model. Time and resources are also concerns, regarding learning and reflecting on which model to use, where the organizational learning-strategy contains no time for reflection. It is faster to use a routine everyone is familiar with and accepts, than spending time and effort to learn a new model.

**Expansive cycles?**

To solve the task, learning must take place in each new project. The practitioners must understand multiple stakeholders’ differing views, underlying contradictions and find ways to work as effortless as possible. This is however not done systematically to maintain knowledge in the project, as advocated by organizational learning theories (see Basten & Haamann, 2018; Garvin, 1993). They do not consciously reflect on the process or make explicit what they gain for future projects. Each project is unique, you never know how the knowledge might be useful in future projects. The experiences gained seem to be dormant knowledge in the organization, connected to the individual or team that holds the experience and may or may not be useful for future projects. The previous learnings are used unconsciously (e.g. Menolli et al., 2013). Practitioners do report extensive learning in work practices, albeit episodically. That is the case for newcomers in a team or for the team when they are directed or inspired by managers, customers or strong internal or external
individuals to use a new way of working (such as a particular model). Nevertheless, it always needs to be adjusted to their way of working.

**Episodic learning cycles**
Organizational learning is said to be important in dynamic environments (Argote et al., 1990). A system development project can be seen as a learning project where practitioners try to understand customer needs and find and develop satisfactory solutions (cf. Bellini & Storto, 2006). When doing so multiple times in new projects, defining new problems and solutions, we can argue for the benefits of developing a way of working that re-uses knowledge. Practice, however, tells another story. Learning is focused on finalizing the current project and finding working procedures that fit the team’s current collective knowledge base, articulated as common experiences, as illustrated in the upper part of the triangle in Figure 1. This dilemma in an AS of solving acute problems or aiming for more long-term benefits, is also described in Virkkunen (2006).

Management of these projects seems to be unreflective based on habit. Practitioners say they interact to reach an agreement on how to work based on experiences from other projects – we interpret that as collective learning. They say also that they do not document their progress and experiences from the project. The knowledge created is stored in participants’ memory and will differ for all practitioners, even within the same project. This becomes dormant knowledge, possible for an individual to transfer to a new project, where it could fit and be agreed upon. Teams seem to have transformative agency to initiate modifications in the models (Virkkunen, 2006; Haapasaari et al., 2018), but they must be agreed on and considered worth the effort for the collective. For fragmented knowledge (knowledge of models that are bounded to specific situations) to be used, it must be merged into the new AS and learned by the collective. For the duration of the project is more like a community of practice, where individuals interact and share experiences and information (Wenger, 1998), but projects are usually short-lived. There is a lack of systematic ways to address recurring problems and sharing information does not seem to be according to what Garvin (1993) prescribes for effective organizational learning. Neither is learning from experience, through reflection-on-action (Schön, 1983) described as a common way.

Organizational learning happens within an AS which may stimulate or restrict learning. Expanding the view, to understand the historical and organizational context, we see a lack of support to enable models to cross organizational boundaries, or even to consolidate them in text. By using the perspective of an AS to analyze how models used in another AS are used, to understand why organizational learning does or does not occur, we can illustrate both dominant structures and barriers to organizational learning that were found in our study (see Figure 1). Practitioners say it takes time to reinforce change. It is not enough for management to give an order; they must be persistent in their communication. For example, one of the practitioners tells how a person who was familiar with waterfall-type models became scrum master but continued acting as a project leader – “not many of the agile principles were implemented”. There is an apparent lack of resources for organizational learning. The basic rule in the organization is to have successful projects. Understanding the uniqueness of projects, not knowing if the knowledge gained in one project will be useful in future project, you focus the effort at hand on making the best situation for the current project (illustrated in Figure 1 with bold arrows). The input of knowledge and codified experience from external AS are symbolized as triangles in the upper left corner of Figure 1. There is a mismatch between the team focus and that of the organization, as illustrated as arrows at the base of the AS (the lower part of the triangle). The outcome of this AS becomes an un-reflexive management of experiences gained in the projects, where expansive learning
is occasionally enforced on an organizational level, as discussed in the section expansive learning, a dilemma that is problematized and explored further in Virkkunen (2006).

In this kind of AS, the organizational learning seems to be episodic. Learning happens sporadically on different levels and at different times. Overall, practice seems unreflective, based on different constellations that happen to work together. Knowledge is gained and experience is built on team level, but it is dormant knowledge for the AS. If initiatives for change and learning are taken, the models will be proven to function well in practice and not only for research. However, the change is still based on the existing experience structures of the team and modified to fit not only the projects characteristics but also the habits of the AS.

Conclusion

In the projects in the substantive area of this study, organizational learning does not occur as such around soft system development models. Learning is well integrated in everyday teamwork, within and between organizations. However, the organization as a whole has a fragmented or dormant knowledge structure where learning occurs in projects but is sporadically reused in new projects. The model in use is learned collectively in the beginning of projects, mainly based on individual experience. The model consists of practical experience from similar projects, and of pieces of established system development models. To coin a new expression, the findings show an episodic organizational learning pattern on each analytical level. First, learning through using models occurs in episodes and the models are subject to reflection-in-action prior to introducing them to the project. Consequently, models are constructed on the fly and when needed. Second, knowledge from experience in a previous project does not reside in the organization, but is individual property, thus dependent on the composition of the team. Third, learning from academia or other practitioners seem to be episodic, by occasionally inviting a lecturer or benchmarking another project’s use of models.

Implications for education and research

One could argue that there is potential in enhancing and managing organizational learning. However, none of the practitioners in the study mentioned that as a need, and they do complete their projects satisfactorily. The lack of interest in using findings and models from research raises questions about the value of research findings – are researchers investigating the right problems and questions? Do they find the right solutions for improvements, properly evaluated and tested in practice by academics and practitioners? Practitioners indicate that system development more resembles a learning process, that follows a certain work practice, following a collectively learned work procedure. If we expand the view to the fourth generation of CHAT (Engeström & Sannino, 2010), the issue of using research-based models in practice can be seen and interpreted on the societal level, concerning which research organizations, publication organizations and the founding agencies of research perspective are included in the learning cycles.

There are implications for education, whether the content and learning goals in curriculums support the students to become professional system developers. Curriculums and textbooks are based more on research-developed content than what is common in practice. Traditionally we teach models developed by research. Will students benefit when being examined on how well they follow methods and models that are of limited interest and use in practice – should students focus instead on learning processes, and be educated in how to modify and integrate different models to the characteristics of the task and the constellation of stakeholders in a practical context?
References


Further reading

Corresponding author
Ia Willamsson can be contacted at: ia.willamsson@hb.se