Software Process Technology and Software Organisations

M. Letizia Jaccheri, Reidar Conradi and Bård H. Dyrnes

Norwegian University of Science and Technology, Trondheim, Norway
letizia@idi.ntnu.no, tel: +47-73-593469 fax: +47-73-594466

Abstract. SPT which have been developed by our research community is not yet used in software organisations. This paper tries to discuss the relation between software SPT and software organisations: it first describes some examples of applications to the software industry. Then it tells about an ongoing case in which we are trying to capture SPT requirements from one software organisation which is operating with its own process representation and associated tools.

1 Introduction

"So far, we have substantially failed to demonstrate that SPT is useful". [CF98]. This assertion will be discussed in this paper. In this work we concentrate on the subset of SPT which consists of methods and guidelines, languages and formalisms and tools to support elicitation and description of software processes. We do not focus on process enactment. We will discuss evaluation and validation experiences of languages and tools to support the description and elicitation of software processes.

Software organisations are not using SPT to support their processes. Software organisations are rather using ad hoc technical solutions to represent their software processes. These solutions include informal diagrammatic languages to express data flow among activities and connections between tasks and roles. Some organisations rely on the intra-net to represent and communicate processes.

There are several published cases, which conclude that, the examined SPT is useful [ADH94] [BFL+95] [EB94] [KH89] [BRB95] [TSK+95] [JBD99] [Ang98] [Høy97]. The common factor of these cases is that one or more researchers use some sort of SPT to support a problem that is derived from the industrial world, and they discuss the bane and benefit of such application. But still there is no evidence that industry has been convinced to adopt SPT on the basis of such cases. These papers and the process modelling community in general base the research work on the assumption that an explicit process representation is the starting point for process understanding, improvement, and communication. This is also the view shared by some improvement methods such as for example CMM and ISO9000.

On the other hand, some researchers from other research communities, such as the computer supported co-operative work one (CSCW), argue that creative
work with strong co-operation aspects, such as software development, does not benefit from static descriptions [Sac95]. They also argue that such work descriptions, when present in organisations, are not followed. These assumptions are always based on empirical studies. The process modelling community has made some efforts to capture requirements for SPT from the observation of software organisations and projects [RKHS94]. However the main part of the research work has concentrated on development of SPT, and eventually application of such technology to the real world for evaluation. Our work in the last years also falls mainly in this last category. It is only recently that we are trying to revert to SPT requirement capture without focusing on the technology we have developed.

This paper is structured as following: Section 2 is about five SPT validation trails that we have been performing in the last years. It also discusses the main weaknesses of our approaches. Section 3 tells about an ongoing work in which we are observing the process definition activities in a company and we try to capture the requirements toward SPT. Section 4 provides references to related work. Some conclusions are given in section 5.

![Diagram](image)

**Fig. 1.** On the Relationship between SPT and Software Organisations
2 Applying SPT to software organisations

Here we will give a summary of a series of attempts we have done in the last years to validate SPT. If we look at figure 1, all these cases try, in some ways, to apply SPT to organisations and to learn from these applications.

The investigative methods used in the trials can all be considered some kinds of case studies [Sta94]. Trial 1, 2, 3 and 5 were instrumental case studies since the cases were examined to provide insight into software process modelling issues. The cases played a supportive role for improving our understanding of the phenomena under investigation. Trial 4 was a collective case study, since data was collected from several organisations/cases which merely played a supportive role. The trials are all discussed according to summary, goal, and an evaluation section.

2.1 Trial 1 Iveco

Summary In 1993, when the \( E^3 \) project was started, we decided to concentrate on software process description (understanding and static analysis) and not on execution. We started by using object-oriented techniques (Coad-Yourdon) to model the software process described in a quality manual we had obtained from Iveco (a division of Fiat). From this first modelling phase we deduced the requirements for the \( E^3 \) system. After the first modelling phase, we had one meeting with the process owners. During the meeting we communicated to them the problems (mainly inconsistencies) we had found in the original description and we showed a Smalltalk simulation of the model. They seemed to like our approach, but they did not commit to use our model instead of the original description.

During the years, we have been using the Iveco process as a reference for internal benchmark [TB97] for subsequent implementations of the \( E^3 \) system. The study is documented in [JBD90]. The requirements coming from this case, were: multiple representation levels, inheritance, process specific constructs, inspection and analysis, associations, and formal syntax and semantics.

Goal Our goal was to validate our initial idea that process descriptions are important per se although they are not executable. Moreover, we wanted to capture requirements on SPT from the interaction with the software organisation.

Evaluation The strengths of this case were: We validated that process descriptions are important, which was our initial idea. We succeeded to capture some requirements from the organisation.

The limitations of this case are:
1. the interaction with the real world is mainly based on the sharing of a document (the quality manual), rather than human interaction with the process owners.
2. As a consequence of the point above, the requirements stem mainly from a modeller point-of-view and not from the process owner one.
2.2 Trial 2 Software Engineering Education

Summary Starting from 1994, the $E^3$ system has been used for four years (1994-1997) in a software engineering course being taught at Politecnico di Torino to describe the software process prescribed for all the student project activities [JL96] [JL97].

Goal Our main goal was educational, i.e., we want the students to learn how to read, understand, and work according to a defined software process model. The research goal was that of evaluating the suitability of $E^3$ description for process understanding.

Evaluation We achieved both goals, since the students had gained a thorough understanding of working with processes and $E^3$ were considered suited for description. The limitations of this study (from the research point of view) are:

1. Students were 5th year computer science students who are well acquainted with modelling techniques. This is not always the case for process users and especially process owners (e.g. managers).
2. No modification of the model was required. In that way students never tested the capabilities of the $E^3$ supporting tool, but only the legibility of $E^3$ FML models.

2.3 Trial 3 Olivetti

Summary One process modeller worked in strict collaboration with the process owners at a Department of Olivetti [FJ99]. The modelling phase triggered changes in the original process descriptions. The process owners seriously considered adopting $E^3$, but a subsequent reorganisation of the Quality Department caused that this never happened.

Goal We had two goals: first, to evaluate the second implementation of the $E^3$ system in general and the partitioning mechanisms in particular; second, to evaluate the suitability of the $E^3$ system for modelling of general manufacturing processes.

Evaluation This case study had a set of positive results. First, the $E^3$ research team gained inside in the activities related to process description and maintenance at Olivetti. Second, the process owners admitted their need for an automatic tool that could help in process description management. We found out that it is often the case that two or more processes from different departments have to be merged together and this merging process often lead to inconsistent models. Third, we took into consideration the requirements of the process owners and not only those of the modellers as in 2.1. The crucial limitation is that the modeller belongs to the $E^3$ research team, is very acquainted with the system and manages to overcome the system limitations. In this way, almost no "problems" of the system came out from the case.
2.4 Trial 4 Modelling in SPIQ

Summary This work was done in the context of the SPIQ\textsuperscript{1} project [Ang98].

In the context of the SPIQ project, larger software companies have a rather formal "house" PML (NOVIT, Storebrand, Tandberg Data, Telenor FoU, Telenor 4Tel, and Ericsson) and a home-made corresponding tool. Software organisations have models of their software processes.

A master student received three process description fragments from three different software companies and modelled them by using the PML $E^3$ and IDEF0. He also developed a Goal Question Metric model [VB94] to evaluate the two PMLs.

Goal The goals were to develop evaluation criteria for a PML and to evaluate extended IDEF0 and $E^3$ PMLs.

The goals were:
1. Gain experience from modelling real world software processes as process modeller.
2. Develop evaluation criteria for PML as a process modeller.
3. Evaluate extended IDEF0 and $E^3$ as PML used by process modeller.

Evaluation The results were concerning understandability $E^3$ was considered more appropriate than IDEF0 which respectively was considered more appropriate than the original process description. The same pattern was identified for ease of creating new models and suitability for the organisation. However, neither of the three process owners were convinced to adopt a formal PML and its corresponding tool.

The advantages of this case with respect to the other ones are: the modeller does not belong to the $E^3$ research team and has very little contact with the team itself. In the context of this case, a goal-oriented metric is defined and measurement are collected from several cases. The limitations are in that the same person designed the metrics, modelled the processes, and collected the measurement data.

2.5 Trial 5 Software Process Improvement Education

Summary This work has been performed in the context of a bigger case in which 40 students (organised in 10 groups) from a software quality and software process improvement course interacted with a major Norwegian\textsuperscript{2} telecom company, named company A. During the case the students came in contact with 3 actors: the quality manager, the manager of the process group, and a project leader. Among other tasks [Jac99], the students were asked by the manager of the process group to model a process fragment, and to report about the encountered problems. Five groups modelled with $E^3$ and 5 modelled with IDEF0.

\textsuperscript{1} SPIQ is a Norwegian project on Software Process Improvement. 1997-1999. SPIQ means Software Process Improvement for better Quality.

\textsuperscript{2} The department we interacted with, has more than 500 employers and this is classified as big in Norway.
**Goal** The research goals are: First, to improve our understanding of the relationship between software process models and software process improvement; second to get an evaluation of E3 and IDEF0 by overcoming the limitation of the previous case (Trial 4 Modelling in SPIQ) in which the modeller and the researcher coincides.

**Evaluation** The advantages of this instrumental case study with respect to the previous ones are: The modellers do not belong to the E3 research team and do not have direct contact with the team itself. They are not part of the evaluation (research) team, nor are they interested in giving a positive evaluation of SPT in general or of E3 and IDEF0 in particular. The evaluation shows that the overall research goal can be considered successful. Both the E3 and IDEF0 languages are considered more appropriate for comprehensibility and model evolution/change than the original description. However, the students reported problems in interpreting the original process description, and said that both E3 and IDEF0 were easy to understand. They also reported problems with the tools.

The reported problems with E3-pdraw are:
- It is difficult to learn how to choose the right associations.
- The tool crashed several times.
- The tool does not provide on-line textual description of the buttons it provides.
- The information regarding associations are not updated automatically as the user creates new associations.
- The tool is poor concerning basic facilities such as cut and paste and printing.

Concerning the AIOWIN tool, only two groups out of five used it, the other three groups decided to use the general purpose drawing tool Visio. This may mean that E3-pdraw is easier to learn and use as all the groups, which were supposed to use it, decided to use it.

The two groups reported about the following problems:
- the AIOWIN version which was available only allowed the creation of 15 activities.
- the decomposition of activities into sub-activities was not intuitive. Moreover, the strict decomposition rules in IDEF0 were not considered appropriate for modelling the original process description.
- Difficulties differentiating between inputs and control flows in the IDEF0 models were also reported.

The reaction of the process owner resembles the ones discussed in trial 1 Ivec (section 2.1) and trial 2 Olivetti (section 2.3). He is positive to the new descriptions, but he does not mention the possibility of adopting a formal PMLs and the associated tools.

### 3 Eliciting SPT requirements from Software Organisations

All trials in section 2 try to apply existing technology to software organisations and to learn from this application. From these trials we have deduced require-
ments towards process technology and we have get feedback about own languages and tools. The common results of these trials is that the proposed technology has not been adopted by the organisations. This is not necessarily a negative feedback for the technology under evaluation as this can be due to organisational reasons.

From our trials we have also learned that organisations have defined processes and they have their own languages and sometimes tools to represent their processes.

In this section we describe and discuss our requirement capture experiences. This case has been performed in the same context as the one in trial 5 Software Process Improvement Education (section 2.5). It is an instrumental case which is still ongoing. If we look at figure 1, we want to start from the software organisation and try to capture its possible requirements toward SPT.

**Description** We have invited three actors (the quality manager, the process group leader, and a project leader) from company A to give a 2 hours presentation each.

The three actors should answer four questions discussed and agreed with the teacher. The questions are:

- Q1: Describe a software system, also by help of quality attributes.
- Q2: Which processes exist around this system?
- Q3: Which are the improvement initiatives around these processes?
- Q4: How general is the software system and the respective improvement initiatives in the general context of the company?

**Goal** The educational goal: students must elicit an example of software process improvement initiatives from a real organisation which belongs to their culture. Each group is asked to write a document and to give a presentation about their understanding of the case. Students are invited to discover consistencies and inconsistencies both among the three actors and between actors, lectures, and text book [Hum97].

The research goal: in addition to the research goal described in 2.5, we want to observe how the company software development processes look like, how they are described, used, changed, customised, and maintained.

**Preliminary evaluation** The process model development was started in October 96 and its current version 2.1 was revised in February 99. The process group leader gave a short presentation of the process web support and provided us with a limited copy of it.

A brief description of some of the relevant points related to the process are given her. This study is still ongoing.

**The software development processes and its representation** The process covers Requirements analysis, Design, Implementation, Verification & Validation, Course & User Documentation Development, and Product & Documentation Administration. The main purpose of the process is to increase focus on the early phases and Verification & Validation activities, increase precision and speed by repeating the events, reduce the dependency of experts and generally improve control and understanding
by visualising the tasks. The process is product, method, language and tool independent. It hence specifies activities, documents, and activity ordering documents are mostly formalised in Adobe FrameMaker.

The process descriptions consist of templates, checklists, review logs, surveys and general project documentation on the Intranet. The Figure 2 gives a flavour about the level of formality exhibited by a process fragment as represented on the company Intranet. The descriptions include templates (both html, word, pdf, and framemaker) for 18 document types, including Specification, Market Requirements, Test cases, etc.

**Process Usage** The process descriptions are available to process users on the company Intranet. Since we do not have talked to process users yet, we cannot tell in which degree and how the process is used by performers.

**Change and Adaptation** Adaptations to new projects are performed in three steps: modify rules and guidelines, create a new server application using the new rules and guidelines, and distribution on the Intranet. The company has performed four adaptations used in 12 projects.

The project leader talked about an adaptation used in a project which involves the use of an online SDL-manual, templates and guidelines for both UML [JRB99] and MSC (Message Sequence Chart). They used prototyping, which is not described in the original company process. Model adaptations and changes are not automatically supported. We are observing that it is not easy for the company to change their models and that errors (inconsistencies) are introduced during model change.

**Technical support to the process** The supporting web based system offers the following functionality:

- Process Overview
- Document Overview
- Activity Overviews
- Sub-process Descriptions
- Document Instructions with check-lists
- Document templates
- Milestone Descriptions with check-lists
- Review Process with templates
- Baseline and Change handling
- Working Methods
- Methodology Interface
- Links to corresponding processes
- Tool-links
- Introduction Course

The quality manager talks about the process as an item in a more complex business process. She never mentions if the process is used or not. The project manager mentions the process, but she never discusses the advantages or the problems bound to the use of it. It is not clear which level of visibility project participants have on the process. This will be investigated as part of a Master thesis by the end of '99.
Fig. 2. The structure of the software process as available on the company A Intranet.
Company A is now in the process of transforming the process from a waterfall like model to an evolutionary incremental one. The nature of process descriptions, i.e., Framemaker drawings makes it difficult to reuse the original process description when making a new one.

4 Related work

Section 4.1 is about trials of technology application to software organisations. These trials are similar to our ones and each one is strongly dependent on the SPT under evaluation. A common factor is that modelling help in discovering inconsistencies.

Section 4.2 compares our ongoing work on requirement elicitation with similar works.

4.1 Applying SPT to software organisations

A survey of previous work shows that much effort have been spent applying SPT to software organisations.

The work of Barghouti et al. [BRB95] is about two case studies that have been performed with the aim of determining the feasibility, utility, and limitations of using a process support tool to model and analyse real processes of organisations. The case studies aim at assessing the utility of formal models and enactment techniques provided by the Marvel 3.1 PSEE. This is similar to our trial 3 Olivetti, except from the enactment issues. There is not evidence that Marvel was adopted by the target organisations.

Kelner and Hansen [KH89] report on a modelling experience whose main goal is the development of a meta-process model for process improvement. The resulting model consists of three sub-models: a functional model, represented by Activity Charts; a behavioural model, represented by Statecharts; a structural model, represented by Module Charts. Textual schemas describe the connections. The Meta-process starts with an information gathering phase with interviews with the people involved in the process. Second, a preliminary process model is developed by use of DFD-diagrams. Third, a set of requirements for software process modelling is developed. Finally, the model is implemented and simulated by use of Statemate. A tool based on Statechart, originally conceived for reactive system specification. The results of the case study have two directions: the first is process modelling understanding, the second is a set of recommendations for change to the methods and procedures in the original process. This is similar to our trial 1 Iveco in which we also performed a process simulation.

Process Weaver have been investigated in at least two case studies: the one reported by Aumatre et al. [ADH94] and the one reported by Ett et al. [EB94]. Aumatre et al. [ADH94] report both modelling of the identified process model specifications and validation of the model by enactment in a simulated environment.
The Spade environment has been evaluated [BFL+95] in the context of a case in co-operation with the Business Unit Telecommunication for Defence (BTUD) of Italtel. The quality manual has been modelled by The SLANG language and the modelling process has lead to the detection of inconsistencies, ambiguities, incompleteness and opportunities for improvement.

Tanaka [TSK+95] reports about an experience of software process improvement at OMRON corporation. One of the goals were to describe and define the current software process correctly and in detail by a Petri net, and estimated the benefits obtained. It was confirmed that, compared to a similar project, approximately 10% of the total effort/KLOC was reduced at test phases as a consequence of better control.

4.2 Eliciting SPT requirements from software organisations

Industry is not likely to publish experience with new tools or methods (e.g. UML or SCM tools). So silence here does not need to indicate that SPT is not used.

Ethnographic studies have been extensively performed in the CSCW community [SK97] and not so much in our SPT community. There are however few exceptions [SR92]. Rodden et al. [RKHS94] [ISS95] report about a series of ethnographic studies. These studies are intended to gain insight into the nature of software development as it actually occurs in practice and to inform the development of effective support for the software development process. Based upon how the organisations performs its everyday work, like "cutting corners" and informal "bending rules", they warn against over prescription, but they advocate the construction of appropriate lightweight mechanisms to support the central features of the process i.e. communication between software developers. These mechanisms should be augmented with facilities that allow process models to be referred to as a common resource for the work taking place.

5 Conclusions

Our SPT application trials and the ones surveyed in literature show that some form of technical support for modelling, consistency check, and browsing helps in developing understandable models and to remove errors and inconsistencies. Moreover, all the persons from software organisations who have been in contact with SPT have been positive to it thought they did not choose to adopt it. Then we cannot conclude that SPT is not useful, but rather that we can show some evidence for usefulness. We are not sure about the reasons about why they did not choose SPT. We assume that there is a combination of technical and organisational reasons. The technical reasons may be re-conducted to the facts that the SPT we have tried to introduce is still in the prototypical stage (as shown for example by trial 5 Software Process Improvement Education 2.5).

This consideration opens two directions for process researchers: first re-engineering of the existing prototypes and further validation of them. The second is less technical and can be performed in collaboration from experts with other
fields, e.g., business administration people, sociologist, etc. This is about discovering the reasons why SPT has not been adopted.

Software organisations have a software process and they use some form for process representation and management. Company A has a software process and has built a technical support for it. They declare that the main reason for having their own “SPT” is process sharing among users by help of intra-net. The developed SPT does not offer any support for consistency check nor changing. At the same time we are observing that the company is having problems with model change. Also model adaptations are not automatically supported.

From our case at company A, it is still not evident which use of the process (both general and customised) is made at project level. Deeper investigations at project level must and will be done by talking to process users.

From our trials and our case with company A, we came in contact with people who believed in the importance of having a defined process and who actually worked with process descriptions. However, this does not prove that software process descriptions are useful. Deeper investigations at project level together with a deeper study of related field literature (e.g., CSCW) will eventually give SPT researchers better answers to these questions.

References


