An Experience in Improving an Industrial Software Process Through Formal Modeling

Sergio Bandinelli Alfonso Fuggetta
Luigi Lavazza and Gian Pietro Picco

April 13, 1994

The growing demand for software applications in critical domains imposes strong quality requirements not only on their performance and behavior, but also on the process followed in developing them. High-quality products are the result of well engineered high-quality processes. However, in real practice, not enough effort is placed in stressing the importance of the definition, analysis, execution, and improvement of the process followed in crafting a software product. Thus, while software investments are increasing worldwide, software processes remains difficult to manage. Actually, there are no broadly accepted rigorous grounds on which guidelines for measuring, improving, and reengineering a software process can be based.

Nonetheless, it is beyond dispute that in order to improve and reengineer the software process, an organization must have a clear understanding of it. In our opinion, is essential to achieve this goal through an explicit and formal description of the software process (i.e., a *formal process model*). In this way, it is possible to enhance the comprehension, communication, analysis, and simulation of the process, making available also the opportunity for process enactment automation.

This paper reports an experiment aiming at restructuring and improving an industrial software process by using formal process modeling notations. The experiment [BFL+94] was started in October 1992, and is still ongoing. It is jointly carried out by CEFRIEL, a university-industry consortium for applied research, and by Italtel, the largest Italian company operating in the telecommunications market. The first phase of the experiment [Pic93] was terminated in June 1993. The goal of this phase was to improve the software process used in one of the Italtel departments, the Business Unit for Telecommunication Defence (BUTD).

The improvement focused on the existing process description, the BUTD Quality Manual [Ita88]. This document is written in natural language, thus providing no formality and leaving the door open to several deficiencies, which we classified as follows:

- **Inconsistencies.** The same concept is described by conflicting statements in different parts of the Quality Manual.
- **Underspecified parts.** Process parts that are not specified can roughly be classified in incompleteness and ambiguity. In an incomplete process description, entire process portions are missing. In an ambiguous piece of process description, the choice among different alternatives is left open and it is not clear how to select among them.
- **Poor organization.** The process description lacks adequate organization and needs restructuring to improve readability and clarity (which does not imply changes in the process semantics).

These problems are mainly caused by the complexity and size of the process, the very high cost of maintaining quality manuals (in terms of quantity and quality of the resources needed), and the lack of an effective support to guide and, whenever possible, automate the process itself.

The experiment goals can be understood in framework described by Figure 1. The figure shows a feedback model of software processes [BFL+94] in which there exists an ideal software process, that is in the head of the Process Owner, which determines the policies, rules, tools, and any other choice pertaining to the software process. The Process Owner describes the Desired Process and produces an Official Process, which in our case is constituted by the Quality Manual. On the

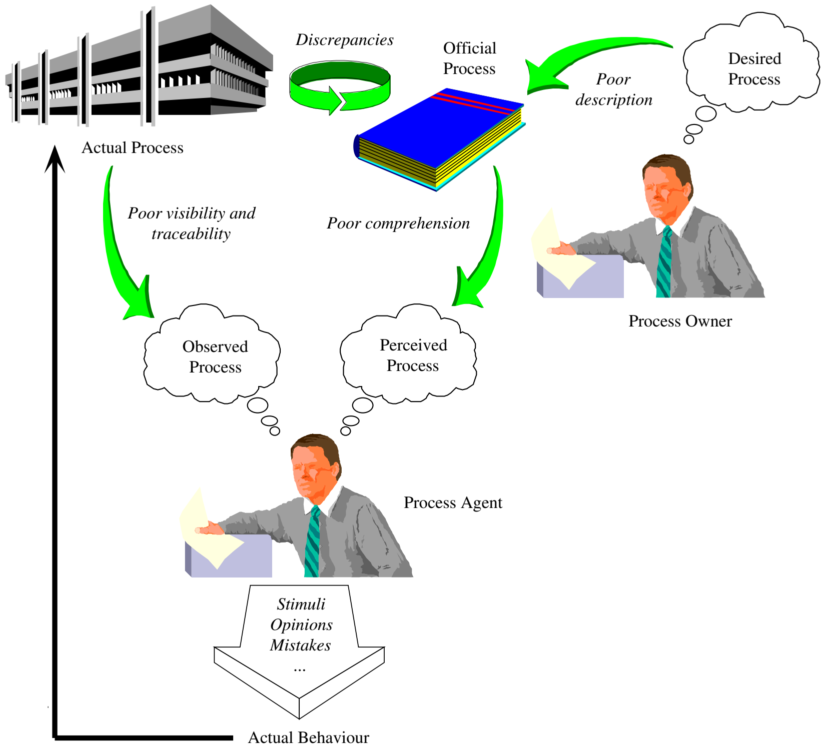

Figure 1: The feedback model of software processes.

other hand, process agents learn about the process in two ways: through the Official Process and through the Actual Process, which is the real process they see during every-day practice. Every agent has its own understanding (its own Perceived Process) of the Official Process, often different from what the documentation is supposed to convey. These different views influence each Process Agent actual behavior. The behavior of all agents is, in turn, the Actual Process. Eventually, we obtain an unstable feedback system that causes the Actual Process tends to progressively diverge from the Desired Process. The stability of the system stability can be enforced only by reducing the sources of divergencies among the Desired, Official, Observed, Perceived, and Actual processes.

In this perspective, the goals of our experiment are:

- To improve the correspondence of the Official Process with the Desired Process.
- To reduce the inconsistencies, incompleteness, and ambiguities in the Official Process.
- To reduce the discrepancies between the Quality Manual and the actual process followed in daily activities (by reducing the gap between the Perceived Process and the Official Process, i.e., by enhancing the understanding of the Quality Manual).
- To establish effective means and procedures to continuosly update and improve the Quality Manual according to the best practices and innovations experienced in the process.

The basic assumption that has generated and driven the experiment discussed here is that *the adoption of a formal process modeling language to describe the Official Process can help in pursuing the goals described above.*

We have used two formal notations, namely SLANG, a Petri nets based process modeling language [BFG93][CEF93], and Finite State Machines (FSMs). The expected benefits derived from the use of a formal notation include:

- Inconsistencies and ambiguities in the process description can be removed or reduced.
- A formal process language can help improve the organization and structure of the process documentation.

- It is easier to assess the weaknesses of the process and to identify opportunities for improvement.
- Discrepancies between the Desired Process, the Official Process, and the Perceived Process can be reduced or, at least, put into evidence.
- Each agent's Observed Process can be more easily checked against the formal definition, thus facilitating the identification of deviations and misunderstandings.
- By introducing environments based on the formal definition of the Official Process it is possible to reduce and better control the discrepancies between the Official Process and the Actual Process.

The first phase of the experiment faced a critical portion of BUTD process, namely the management of anomaly reports (that covers nearly 200 natural language pages of the Quality Manual). This is a typical change process, where configuration items have to be modified to fix problems signaled either by verification and validation activities or by users. Consequently, in addition to straight anomaly management, also configuration management issues had to be taken into account.

The whole work was mainly accomplished by two persons: the CEFRIEL's process engineer, an expert in process modeling technology and methodology, and in process modeling formalisms used in the experiment, and the Italtel's process owner, member of BUTD and with a deep knowledge about the considered process.

The experiment yielded some relevant results, including:

- **Restructuring and disambiguation of process documentation.** We mentioned before the problems found in BUTD Quality Manual. Using a formal notation to describe the process allowed us to disambiguate many underspecified portions of Quality Manual. In addition, we made a heavy restructuring work on the FSMs that were already used in BUTD, to describe the allowed status of configuration items (e.g. source code, documentation) and control forms (e.g. Anomaly Reports). These FSMs were lacking uniformity and contained less states than required, resulting in a description (the Official Process) not matching the Actual Process, and in a reduced support for process metrics.
- **Discovering the possibility to follow the Configuration Management procedures stated in BUTD manual in a fully automated way.** BUTD owns a well-established set of procedures used during daily work, both for anomaly management and straight development. Most of this (routinary) tasks are carried out by humans, with a few support from dedicated tools. The model developed by the process engineer, however, evidenced that a smart restructuring of the information attached to items, would allow the creation of a process model that does not require any human intervention.
- **Description "from scratch" of human resource allocation in anomaly management.** Despite the outstanding importance of human resource allocation in any industrial process, the Quality Manual contained no information about this issue. Of course, well-established rules are enacted during every-day operations. Our activity produced a model of human resource allocation involved in anomaly management that is both comprehensive and well integrated with the rest of the specification.

The experiments at BUTD are still ongoing. Presently, we are taking into account a larger part of software process. In addition, we plan to use the SPADE environment [BFG93], in which SLANG models can be enacted, to guide and support those parts of the process which have been already modeled. In particular, one of the key issue that we are addressing in this second phase is to define suitable means to collect process and product-related metrics, in order to steer the process towards the common engineering practice of monitoring both the process and the product through quantitative data. This is essential to provide the rationale and the foundations for future process improvement initiatives.

References

[BFG93] Sergio Bandinelli, Alfonso Fuggetta, and Carlo Ghezzi. Software Process Model Evolution in the SPADE Environment. *IEEE Transactions on Software Engineering. Special Issue on Software Process Evolution*, 1993.

[BFL+94] Sergio Bandinelli, Alfonso Fuggetta, Luigi Lavazza, Maurizio Loi, and Gian Pietro Picco. Modeling and Improving an Industrial Software Process. Technical Report 94000, CEFRIEL, via Emanueli 15, 20126 Milano, Italy, February 1994. Submitted to *IEEE Transactions on Software Engineering.*

[CEF93] CEFRIEL Group. The SLANG 1.0 Process Modeling Language Reference Manual. Technical report, CEFRIEL-Politecnico di Milano, 1993. Esprit Project 6115 (GOODSTEP) deliverable.

[Ita88] Italtel BUTD. *Manuale del Sistema di Qualità Software*, 1988.

[Pic93] Gian Pietro Picco. Modeling a *Real* Software Process with SLANG. Technical Report 93059, CEFRIEL, via Emanueli 15, 20126 Milano, Italy, July 1993. CEFRIEL Master Thesis.