TAPISTRY: A Software Process Improvement Approach Tailored for Small Enterprises

Joey van Angeren (3227162) Group 2
Department of Information and Computing Sciences, Utrecht University
Princetonplein 5, 3508 TB Utrecht, the Netherlands
J.vanAngeren@uu.nl

1 Introduction

Software process improvement is an emergent theme ever since the expansion of the software industry. From the 1990s onwards, a lot of attention from both industry and academia has been devoted to the development of various complex assessment methods and maturity models to measure and improve software processes. A software process is defined by Cook and Wolf (1998) as a: “set of activities applied to artifacts, leading to the design, development, or maintenance of a software system”. Organizations embrace the need to streamline and optimize the development and maintenance of their software or software products and in doing so tend towards these complex standards. An example of such a method is BOOTSTRAP (Kuvaja & Bicego, 1994; Kuvaja, 1999). The BOOTSTRAP approach enables large organizations to improve their software processes through a guided process assessment with accompanying questionnaires and an algorithm to sketch a detailed maturity profile. This assessment is then followed by provision of guidelines on the main areas to improve. Often, organizations are assisted in this by certified trainers or consultants.

Most of the models, methods and ISO standards, are targeted at process improvement for large organizations. That is why, some of the initiators of the BOOTSTRAP method introduced a self-assessment for smaller enterprises called BootCheck (Doiz, 1997). This self-assessment is targeted at enterprises with sixty or less employees and is a slim version of the initial BOOTSTRAP; an assessment that only focuses on process areas and expertise found relevant for small to medium-sized enterprises (SMEs). With this BootCheck as an assessment tool, a new method was introduced to facilitate SMEs in improving their software processes; tailored application of software process improvement techniques for small enterprises (TAPISTRY) (Kuvaja, Palo, & Bicego, 1999). TAPISTRY is a software process assessment and improvement method in the form of a two-day workshop. During these two days, managers and project leaders from SMEs systematically work on improving the processes of their enterprise under supervision of certified institutions. This method succeeds respectively by identification of business needs, assessment of the current state of software processes, definition of organizational goals and by defining and allocating process improvements actions and budgets.

This paper presents a formal method description of TAPISTRY. For this, principles from the domain of method engineering are employed. Brinkkemper (1996) defines method engineering as: “the engineering discipline to design, construct and adapt
methods, techniques and tools for the development of information systems”. The formal method description is described in a standardized way, by means of a Process-Deliverable Diagram (PDD). This meta-modeling technique bundles an UML activity diagram to describe the sequence of activities that together constitute a method and an UML class diagram that describes the deliverables that accompany one or more of these activities into one diagram. This standardized method description helps organizations understand the process of a TAPISTRY session and facilitates them in performing a self-assessment of their software processes. Furthermore, separate method fragments from TAPISTRY can be included or employed to create new methods.

The remainder of this paper continues with an overview of scientific literature related to TAPISTRY and software process improvement in section two. Section three contains a description of an example scenario in which TAPISTRY is used, it facilitates in understanding the method and provides some examples of method deliverables. The fourth section contains the formal method description of TAPISTRY with its accompanying description.

2 Related Literature

The predecessor of TAPISTRY is BOOTSTRAP. BOOTSTRAP was initiated out of an ESPRIT project that received funding from the European Commission (Kuvaja & Bicego, 1994; Kuvaja, 1995, 1999). The initial version of BOOTSTRAP was introduced in 1993 and was based on the SEI model, the initial version of CMM (Kuvaja, 1995, 1999). In version 3.0 BOOTSTRAP also adopted SPICE and its accompanying ISO quality standards (Kuvaja, 1999). The BOOTSTRAP approach focuses on process assessment, benchmarking and improvement in a specific software production unit (SPU). Since the method is targeted at large organizations this is often one specific team or department within the organization. Because of the large-scale assessment, an organization has to consult certified assessors that perform BOOTSTRAP. Where the initial BOOTSTRAP assessment focuses on thirty-five process areas, the BootCheck self-assessment that is used in TAPISTRY addresses nineteen areas. These areas have, according to best practices, been found most applicable for smaller enterprises. An overview and categorization of these process areas, as summarized upon by Kuvaja et al. (1999) is drawn in figure 1.

Hansen, Rose, and Tjorneho (2004) classify software process improvement methods based on their original rationale. According to their classification, both BOOTSTRAP and TAPISTRY are prescriptive and norm-driven methods. This means that both methods stem from the notion that software processes are measureable and that there is a sort of optimum, which as a consequence means maturity models play a vital role in process improvement with these methods. Other models classified by Hansen et al. (2004) in the same category are CMM, TSP and SPICE.

With the rising interest for software process improvement, also the number of software process improvement methods tailored to the needs of smaller enterprises grew, ever more since a study conducted by Dyba (2003) did provide empirical evidence that there is an equal need for software process improvements in small enterprises compared to their larger equivalents. Mishra and Mishra (2008) and Mishra and Mishra
Fig. 1. The processes that are covered in the BootCheck self-assessment

(T2009) described and compared respectively five and six process improvement methods for small enterprises. According to their comparison, almost all of these are based on existing reference models such as the CMM. The same applies for TAPISTRY that did adopt the reference model used in BOOTSTRAP but can also be combined with, for example, SPICE. In that sense Habra, Alexandre, Desharmais, Laporte, and Renault (2008) propose a method for very small enterprises (e.g. twenty-five employees or less) that can also be used in combination with multiple reference models. The method in fact contains an activity in which the suitability of certain aspects of these models in a specific setting is evaluated. Mishra and Mishra (2008) also found that most phases and activities show similarities, for instance, similar to TAPISTRY most methods start out with the identification of business needs. Although comparison studies of different software process improvement methods exist, no case studies have been published about the actual implementation of TAPISTRY.

3 Example Scenario

Suppose a small software development firm, with approximately forty employees. The current management is dissatisfied with their current software processes. Furthermore they want to receive external quality accreditation for their process quality in order to participate in a bid for a large project. After a board meeting, they decide to attend a two-day regional TAPISTRY workshop to have certified trainers help them in increasing their current process maturity. They decide to send the board member with the most
knowledge about the organization and its processes, since this is a prerequisite for a successful self-assessment.

**Definition of business needs** After the company representative has received the necessary introduction, the first step in the process is the identification of business needs. With the help of other workshop attendees and coaches the board member fills in a SWOT matrix. By filling this matrix, the board member reflects on the main strengths, weaknesses, opportunities and threats to sketch a detailed business profile of the firm. This awareness will aid in performing the self-assessment as in prioritizing improvements later on.

**Performance of self-assessment** When the business needs are determined, BootCheck is employed to assist the board member in performing a self-assessment. For the nineteen process areas that are part of the BootCheck self-assessment, the employee has to fill a questionnaire, very similar to the one employed in the original BOOTSTRAP (Kuvaja, 1995). A fragment of such a questionnaire is included in table 1. After the questionnaire has been finished, a report is generated that provides insight in the results, followed by an analysis.

<table>
<thead>
<tr>
<th>Question #</th>
<th>Process</th>
<th>Level</th>
<th>Absent</th>
<th>Weak</th>
<th>Fair</th>
<th>Extensive</th>
<th>Not Applicable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.1</td>
<td>MAN.1</td>
<td>1</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.2</td>
<td>MAN.1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.3</td>
<td>MAN.1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

**Definition of improvement goals and priorities** The final deliverable of previous step is a maturity profile that sketches the current state of software processes. A description of the maturity levels used for this (Kuvaja et al., 1999) are listed in table 2. Based on this profile and the business needs as defined in an early stage of the TAPISTRY workshop, missing or weak processes are ranked and prioritized. Prioritization of processes proceeds based on the relevance and criticality of a process. In the scenario, for example, the board member finds out management and lifecycle independent processes are lacking, out of which the lifecycle process especially are found critical because they are important in the project bid.

**Definition of process improvement actions** Based on the ranking of all missing processes a selection has to be made of processes that will be actually implemented, addressed or improved. Again, organizational goals but also limitations play an important role in this. With regard to the sketched example scenario, the organization decides
Table 2. Process maturity levels for TAPISTRY

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0. Incomplete</td>
<td>There is general failure to attain the purpose of the process. There are little or no easily identifiable work products or outputs of the process.</td>
</tr>
<tr>
<td>1. Performed</td>
<td>The purpose of the process is generally achieved. The achievement may not be rigorously planned and tracked. There are identifiable work products for the process, and these testify to the achievement of the purpose.</td>
</tr>
<tr>
<td>2. Managed</td>
<td>The process delivers work products according to specified procedures and is planned and tracked. Work products conform to specified standards and requirements.</td>
</tr>
<tr>
<td>3. Established</td>
<td>The process is performed and managed using a defined process based upon good software engineering principles. Individual implementations of the process use approved, tailored versions of standard, documented processes. The resources necessary to establish the process are also in place.</td>
</tr>
<tr>
<td>4. Predictable</td>
<td>The defined process is performed consistently in practice within defined control limits, to achieve its defined process goals. Detailed measures of performance are collected and analyzed, and performance is quantitatively managed. The quality of work products is quantitatively known.</td>
</tr>
<tr>
<td>5. Optimizing</td>
<td>Performance of the process is optimized to meet current and future business needs. Quantitative process effectiveness and efficiency goals, for performance monitoring are established, and improvement is achieved by analyzing the results.</td>
</tr>
</tbody>
</table>

To improve its process related to software implementation & testing, maintenance and quality management.

Define process improvement budget and time schedule  Based on the aforementioned improvements, a project has to be initiated to reach the desired goal. A clear specification of costs and cost categories as well as a detailed project plan must be created. Amongst others, the board member has to create a work breakdown structure in which he allocates resources to the to-be carried out process improvement project. This, to aid the organization in executing the actual improvement after the workshop is finished.

References


