Using Software Inspection as a Catalyst for SPI in a Small Company

Lasse Harjumaa, Ilkka Tervonen and Pekka Vuorio

Department of Information Processing Science,
University of Oulu
P.O. Box 3000,
90014 OULUN YLIOPISTO
lasse.harjumaa@oulu.fi, ilkka.tervonen@oulu.fi

Buscom Oy, Elektroniikkatie 4
90570 OULU
pekka.vuorio@buscom.fi

Abstract. Process improvement activities in small and medium size enterprises (SME) are challenging due to small number of personnel and projects, people have to perform in a variety of roles. Assigning process improvement activities to the overburdened personnel may be seen as a threat to ongoing projects. Both management and staff should become convinced of the benefits of the improvement actions before taking the first step of the process. Even in this situation the software inspection provides a tempting starting point for process improvement. It is a focused and well defined subprocess and enables high return on investment benefits even in short period use. Our experiment in a small software company confirms that software inspection provides the justified starting point for process improvement. By means of the inspection maturity model the company recognises the weak points in their review practice and inspection patterns help in discovery of improvement actions.

1 Introduction

Several methods and tools exist for determining and improving the quality assurance function in software organizations. The most widely used are CMMI, ISO9001 and SPICE. However, small companies have difficulties in applying these models in their full extent [1]. SPI is still possible, but it requires simplification of the improvement models. [2] In small companies, all SPI activities have to be weighted differently: the goal is not to achieve certain level of maturity, but to concentrate on the real quality and business goals of the company. [3]

Software inspection is amongst the most effective methods when evaluating the return on investment of different quality assurance techniques [4, 5]. Inspections are useful from several viewpoints: In addition to clean-up and prevention of defects, they serve as training method [6] and valuable data source for knowledge capturing and

management [4, 6]. The inspection process even provides means for process measurement and management [7, 8], and can enhance team collaboration and communication [4].

The original software inspection process, as described by Fagan [9], or its later variations are very rigorous. There may be occasions when a customized or slightly restricted version of the process is more applicable. For example, the inspection team may be geographically scattered, or the organization may have limited resources for running an exhaustive inspection process. Even though these more flexible forms of inspection exclude certain details of the traditional processes, they also are capable of producing excellent results by the means of product and process quality. Furthermore, inspections can be carried out to all types of software artifacts, including requirements and design documents. [10, 11]

Being such an effective method, software inspection provides a feasible starting point for small companies for improving their software production processes. The goals of process improvement in small organizations are probably different from those in large ones. While the main focus in large companies is to achieve better efficiency and reliability at lower cost, small companies aim at controlling their growth and fighting chaos.

[12] suggests that implementing a limited set of well-defined quality assurance processes is the most beneficial SPI approach for a small company. A process should be a tool for producing the software, not a purpose in itself. Processes should also be tailored for the needs of the organization.

From inspection process improvement viewpoint none of the general or testing tailored models is enough oriented to inspection evaluation. The really justified improvement suggestions are also rare, which comes from the laborious data collection, analysis and interviews required for them. Due to these reasons we in this paper introduce a light capability model tailored especially to inspection process evaluation that looks for weak points through indicators and base practices. The model is called i3GO (improved inspection initiative group in Oulu) capability model, and it provides a skeleton for semi-formal interviews and thus help in the planning and implementation of evaluation.

The main idea of the tailored capability model comes from BOOTSTRAP [13], i.e. we look for the base practices of the inspection process and evaluate the grade of each practice in the company/division/project (whether it is in use totally or partially). The evaluation is based on indicators, which can be enabling or verifying ones. The idea is that, if we find indicators of a base practice, we then have some justifications for assuming the existence of that base practice. If some practices are at a low level or missing, they should be targets for inspection process improvement, usually based on discussions with company staff. We also have generated a preliminary set of improvement patterns by means of which companies can find improvement activities more easily.

We will first present some rationales behind the model, and then introduce the software inspection process with the indicators and base practices. We will then explain the usage of the model and improvement patterns in more detail and go on to present some further research ideas.

2 Background of the Model

There are a number of process capability determination models in the area of software engineering, including tailored testing models such as the Testing Maturity Model (TMM) [14] and Testability Maturity Model [15]. From an inspection process improvement viewpoint none of the general or testing models is sufficiently oriented towards inspection evaluation, although BOOTSTRAP and SPICE [16] allow focused process assessment and improvement, i.e. they allow one to choose the process to be improved (e.g. review/inspection. Proper reporting of inspection improvement initiatives has also been rare, although there are some good examples e.g. at Bull NH Information Systems [8] and Hewlett Packard [17], which focus on the analysis of current status.

Most software development organizations in Finland are relatively small, or at least they are divided into rather autonomous units. To stay competitive, it is important that they upgrade and continuously update their software production processes. However, the most popular software process improvement models have a number of deficiencies where the small companies are concerned – full-sized improvement projects are far too costly and time-consuming. [18, 19, 20]

Software process improvement models and tools should be adapted for the needs of SMEs. [21], for example, suggests a computerized tool, which aims at speeding up the improvement and reducing the amount of necessary resources. Another approach is to focus the SPI on the most crucial parts of the development. By taking one clearly defined practice, the software inspection, into the focus of the improvement, several advantages may be gained:

- An organization can "practice" the SPI in manageable scale. Later, a full-sized SPI programme can be launched.
- Benefits of the inspection on product quality are considerable and measurable. This facilitates management and personnel commitment to the SPI effort.
- Inspection has positive side-effects on the process: improved communication and knowledge transfer.
- Single quality practice – especially one defined as rigorously as inspection – is easier to understand and manage when introducing substantial changes to it.

Software inspections are gaining increased acceptance in the software engineering industry. Even though industry experience shows that inspections reduce cycle time, lower costs, increases process visibility, improves programmers' capability and, of course, improve quality by preventing defects [9, 6, 22], the inspection process has not gained widespread usage according to the Software Engineering Institute [23]. To promote the utilization of inspections, we suggest inspection improvement patterns for easy installation and usage of the process.

The use of abridged and focused assessments to launch SPI has been successfully experimented and reported in [24]. Focused assessments and SPI activities can also be applied to a single project within a large organization to find out the maturity of a particular development effort. Focused interviews can be used to determine the current maturity [24]. For the best results, these interviews should be carried out in a short time frame, have participants that possess adequately broad views on the

development process and concentrate on achieving consensual opinion about the current state of the project or organization.

According to [25], an improvement model aimed at small companies should focus on the software processes that are most important to it, and provide fast return on investment. Inspections can provide that. Richardson also states that an improvement model have to be flexible and easy-to-use. [25]

Human factors are emphasized in small organizations. They are more dependent on individuals, as developers usually become experts on certain domains. Expert knowledge is rarely documented and thorough documentation is probably not required at all in an immature process. Due to small number of employees and projects, people have to perform in a variety of roles. Assigning additional SPI activities to the personnel may be seen as a threat to ongoing projects. Furthermore, development processes and tools will more likely change more often than in large organizations. Processes have to be adaptable to the specific requirements set by each customer. [26]

Due to the limited resources and divergent environment, the model is not allowed to inhibit the creativity and dynamics of projects by enforcing arbitrary or artificial procedures, but provide the essential quality operations to help the projects to concentrate on the most important details [26]. To sustain the improvement, the SPI assessment and improvement activities should also be repeatable at a reasonable effort.

Generic process improvement models typically describe the ideal outcome of the improvement actions in adequate detail, but do not provide exact guidance on how to proceed in the improvement. For example, [27] report that even though persons involved in SPI understand what need to be improved, they require more specific guidelines about how to conduct the improvement. Instead of discussing the processes and assessment items at a generic level, the improvement model should deal with concrete and well-known issues that are relevant to the company.

We introduce a pattern approach to direct the implementation of the inspection improvement activities. Patterns describe general solutions for frequent problems. In software engineering, design patterns for object-oriented development have been widely used and recognized as a valuable add-on in program design and implementation. [28] In addition, patterns have been applied to software processes. For example, [29] represents pattern approach for software process improvement to overcome some typical difficulties. Patterns have also been applied to capture and refine the software engineering knowledge into reusable form [30].

Patterns also allow adaptation of the SPI implementation for each organization. Such tailorability is a significant factor in successful improvement and reduces natural resistance against change [31]. Patterns are concrete and manageable descriptions of improvement actions needed in various situations.

Finally, the inspection improvement model should take into consideration generic, full-scale process improvement models. The i3GO model attempts to support the company in its course to a greater level of maturity, and if the company is interested in outside certification, such as ISO, the i3GO improvement model must not hinder, but promote that. Compliance can be achieved by using similar terminology and concepts as in generic models. Furthermore, the inspection process is not a separate function in a development cycle. Inspections cannot be truly capable, if there are not

other quality assurance activities in use in the organization. For example, metrics that are gathered in inspections have to be defined, managed and reported. Thus, the improvement model should encourage and prepare the organization to improve other development and management processes as well.

3 The i3GO Model

The i3GO model is based on the ideas and structure of the Bootstrap methodology [13]. The model forms the basis for the software inspection process assessment by defining the process as precisely as possible, yet allowing company-specific tailoring of the process.

Inspection is traditionally defined in terms of steps such as entry, planning, kickoff meeting, individual inspection, logging (inspection) meeting, edit, follow-up, exit and release [6, 9]. The structure of the ideal process of inspection which we use as a reference model in evaluation is based on these steps.

Although the ideal process is defined in the BOOTSTRAP model as a set of base practices, our interpretation of these differs in that we also include organisational and supporting activities among them. The discovery of base practices has been guided by six specifically defined goals:

1. to identify defects in an artifact,
2. to estimate the quality of an artifact,
3. to improve product quality,
4. to provide data for process improvement,
5. to provide the means for knowledge transfer, and
6. to improve the effectiveness of the development process.

These practices can be classified into three groups (cf. Figure 1): organisational activities (at the top), that ensure continuous improvement and efficient organisation of the inspection process, the core set of activities (the middle bar), that are the essence of any implementation of the inspection process (as defined in inspection books, e.g. [6]), and supporting activities (at the bottom), that help in carrying out an instance of the inspection process.

The organizational activities are: P.1. Establish and improve the inspection process, P.2. Organise the inspection and P.3. Train participants. Work products, reports and other material related to these activities are represented in the figure by arrows. The core activities are P.7. Check the preconditions for inspection, P.8. Plan the inspection, P.9. Find issues in the artifact, P.10. Categorise defects, P.11. Make corrections and P.12. Conclude the inspection. Finally, the supporting activities are P.4. Support with computer tools, P.5. Maintain rules and checklists and P.6. Refine information.

Figure 1 also depicts data flows between activities. According to the BOOTSTRAP model, the data flows are used for evaluating the existence of a base practice, and we thus call them enabling and verifying indicators. The capability model has been experimented in five software organizations, and it has been refined based on the usage experiences. Experiments are described in more detail in [11].

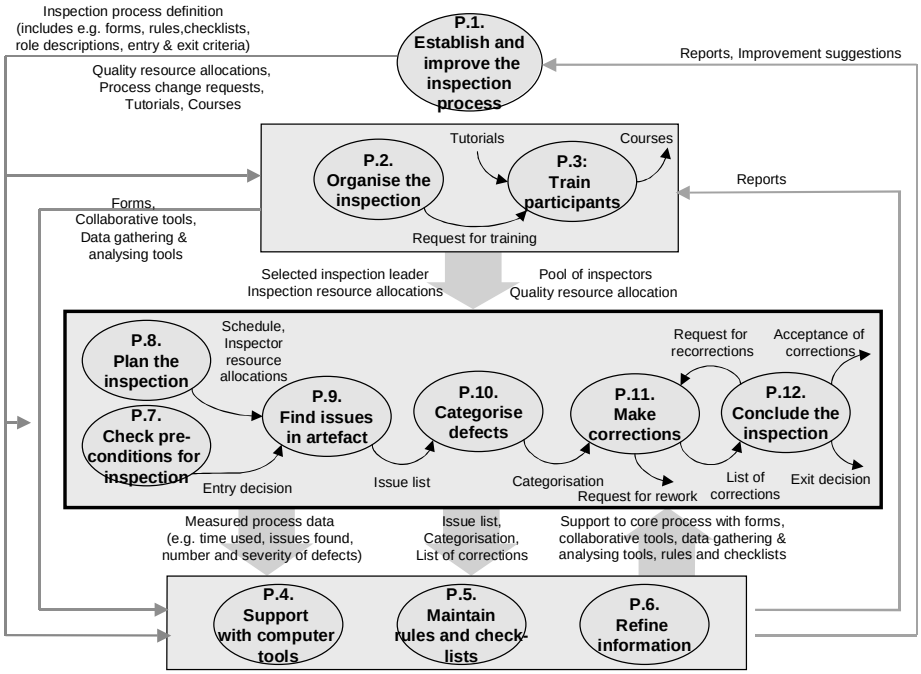


Fig. 1. The i3GO base practices and indicators.

Base practices and indicators form the skeleton of the capability model. For example, the base practice P.9, Find issues in the artifact, is one of the core activities and serves all six goals of the model. Table 1 lists the indicators related to the practice.

Table 1. Enabling and verifying indicators.

Possible enabling indicators	Possible verifying indicators
Forms Rules Checklists Schedule Inspection resource allocation	Measured process data Issue list

The indicators are evaluated and the existence (and grade) of the base practices can be analysed and justified in the light of the results. For example, “issue list” points to the potential defects in the artifact. List should be sent to the author, and each issue in the list should include at least information about its location, severity and checklist reference. If any crucial information is missing, the grade will be lower. The grading depends on the needs of the organization. The process has not to be “ideal” in every

occasion. In a small company, some practices may be rather informal, but they are still adequate.

According to our experiences in the industry, company representatives were keen to discuss the potential improvement actions already during the assessment meetings. For this purpose, the outcome of the assessment is recorded into a spreadsheet-based tool, which calculates and represents the result of the evaluation instantly.

Furthermore, concrete guidelines for updating the process have to be at hand immediately after the assessment. These are provided by means of improvement patterns, which describe the most typical improvement strategies. The next chapter discusses the assessment and improvement parts of the model in more detail.

Assessments and Improvement with the i3GO Model

The inspection process improvement with i3GO model starts with an assessment, which is carried out during an interview session. It immediately shows the results of the evaluation. After that, overall goals for the improvement are set, and finally a set of actions are performed to gain the improvement. The goal-setting and actual improvement procedures are aided by the improvement patterns.

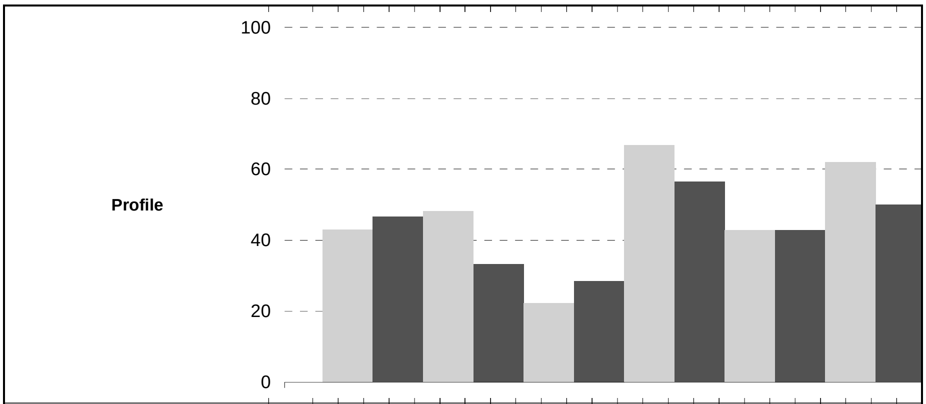
3.1 Determining the Capability

Capability determination is based on checking of the enabling and verifying indicators. An enabling indicator confirms that the preconditions for a specific base practice are met, and the absence of such an enabling indicator strongly suggests that the corresponding base practice does not exist. A verifying indicator confirms that a specific base practice has produced appropriate and adequate results, and its existence suggests that the corresponding base practice may exist, but does not guarantee this.

The matrix for capability evaluation is presented in Figure 2. The first version of the model included 35 indicators, but according to our experience in four cases we removed some indicators and added new relationships. Now there are 29 indicators to be walked through during an evaluation. The goal was to avoid overlapping indicators but to retain an essential set of indicators for determining the capability of base practices. The enabling and verifying types of indicators are described with letters E and V, correspondingly.

The existence of an indicator is evaluated on a five-grade scale:

na	not relevant to the organisation
0	not in use at all or only rarely
1	partially, exists to some degree
2	largely, exists fairly well
3	fully, always in existence



Indicator	Score	P.1	P.2	P.3	P.4	P.5	P.6	P.7	P.8	P.9	P.10	P.11	P.12
	%	43	47	48	33	22	29	67	56	43	43	62	50
1) Entry decision	2							V 4		E 2			
2) Entry criteria	1	V 2						E 1					
3) Schedule	3								V 6	E 3			
4) Inspection resource allocations	1		V 2						V 2	E 1			
5) Request for rework on another a	1											V 2	
6) List of corrections	3						E 3					V 6	E 3
7) Acceptance of corrections	1												V 2
8) Request for recorrection	0												V 0
9) Exit decision	3												V 6
10) Exit criteria	1	V 2											E 1
11) Forms	2	V 4	EV 6		EV 6					E 2	E 2		
12) Tutorials	2	V 4		E 2							E 2		
13) Courses	1	V 2		V 2						E 1			
14) Collaborative tools	1		V 2		V 1				E 1	E 1			
15) Data gathering & anal. tools	0		V 0		EV 0	E 0	E 0			V 0	E 0		
16) Rules	2	V 4				EV 6			E 2	E 2	E 2	E 2	
17) Role descriptions	1	V 2		E 1		E 1			EV 3				
18) Checklists	0	V 0				EV 0			E 0	E 0			
19) Issue list	3						E 3			V 6	E 3	E 3	E 3
20) Categorisation	0					E 0	E 0				V 0	E 0	E 0
21) Pool of inspectors	2		V 4	V 4					E 2				
22) Selected inspection leader	3		V 6					E 3	E 3				
23) Request for training	1		V 2	E 1									
24) Quality resource allocations	1	V 2	V 2										
25) Reports	0	E 0	E 0	E 0	V 0	E 0	V 0						
26) Measured process data	0					E 0	E 0			V 0			
27) Improvement suggestions	1	E 1											
28) Process change requests	1	V 2	E 1			E 1							
29) Inspection process definition	3	V 6	E 3	E 3	E 3				E 3				
	sum	31	28	13	10	8	6	8	22	18	9	13	15
	max	72	60	27	30	36	21	12	39	42	21	21	30

Fig. 2. Capability determination matrix.

As we can see from the matrix, one indicator can affect a number of base practices. In the evaluation session we walk through all the indicators. They are grouped according to core, supporting and organisational base practices, the first 10 indicators applying to the core set of base practices, indicators 11-20 being related to all groups of base practices and indicators 21-29 being mainly for organisational base practices. This order of indicators is justified because it is easier to start the assessment session by evaluating core activities and to go on to supporting and finally organisational activities. This is understandable, because indicators such as pool of inspectors,

quality resource allocation and measured process data require a deep understanding of the company's inspection process and this information is typically gathered from discussions with the company staff during evaluation session.

The result of a capability evaluation performed in an IT company is also presented in Figure 2. The diagram at the top of the matrix shows an estimate of each base practice in graphical form, and at the bottom of the matrix the estimate is represented in numeric form. The company has about 70 employees in Finland and our assessment revealed weak points in the inspection process at the whole company level.

In the capability evaluation we start by estimating the grades of existence for each indicator. This estimated value is then multiplied by the weight attached to the indicator type (enabling = 1, verifying = 2) and placed in the matrix. The fact that verifying indicators carry twice the weight of enabling ones is justified on the grounds that they really guarantee that something has been done, whereas enabling indicators merely make certain activities possible. Finally the capability value is calculated for each base practice by means of sum and max values depicted at the bottom of the matrix. The values are also depicted in the form of a diagram. As a result of the evaluation, a profile describing the capability of each activity is created, and put together these estimates give overall view of the process as a whole.

In this company, the staff is mostly young people in their first established post. The company has recently started to improve its quality assurance, however, and systematic inspecting has been introduced as a new practice. We have evaluated the company twice, the second occasion giving a more accurate result, because the interviewees had a more accurate picture of the inspection process, although both evaluations pointed to similar weak points. The core set of base practices is fairly well established, but supporting practices are almost entirely absent. The most important improvement suggestions should be focused first on core practices and after that on supporting ones, i.e. the company should start with adequate definition of the inspection process and appropriate training in this process. The definition of checklists and guides for defect categorisation would help to improve base practices P.5, P.9 and P.10 in particular, and the training would affect P.3. The company does not collect measurement data regularly nor does it refine such data into reports for the management, which means that they will have to define the metrics for inspection improvements and the tools and data required for that purpose at a later stage.

3.2 Improvement with Patterns

According to the experiments and evaluations that have been made with the i3 capability model, inadequacies in software inspection processes are often similar in distinct organizations. In addition, elements and features for the inspection improvement patterns have been extracted from related inspection literature and research.

Improvement patterns for software inspection process are introduced for straightforward implementation of improvement actions. Pattern approach provides a set of practical guidelines for actually executing the improvement actions, which are often missing in the full-scale assessment and improvement models.

After the evaluation, diverse approaches can be taken to actually achieve the process improvement. First, one can go through the indicator list once more, placing emphasis on the indicators with low scores. It is useful to have an expert to aid in this procedure, as the meaning and implications of an individual indicator may go further than the name of the indicator implies. For example, indicator "tools" have to be interpreted in a different way depending on the situation. Sometimes a comprehensive groupware package is needed, sometimes paper forms and templates are sufficient.

Even though the indicators are rated according to their importance for the organization, defining improvement steps requires in-depth understanding of the inspection process. Inspection improvement patterns are pre-defined lists of guides and procedures of the most typical action points to achieve process improvement. Each pattern defines an unambiguous goal for the inspection process upgrading. The purpose of the pattern catalog is to aid the assessor and the company representatives to focus the improvement activities on the most crucial parts of the process. Typically the weakest activities are addressed first.

When utilizing the patterns, an organization sets an overall goal for the process improvement effort. The goal can be based on the assessment report or common sense. If particular problems are uncovered during the assessment interviews, these can direct the goal setting. The most straightforward method to determine suitable objectives is to read through the symptoms in the improvement pattern descriptions.

After determining the main goal, the most suitable patterns for the situation can be selected from the pattern catalog. Currently, there are seven patterns in the catalog. For each pattern, the following characteristics are defined:

Purpose of the pattern describes how the inspection process is enhanced, if the pattern is used.

Symptoms section presents a variety of problems and difficulties. If these have been encountered during inspections, the pattern may be suitable for the situation.

Possible reasons for the problems or inefficiency are listed.

Activities related to the pattern are listed.

Action list represents a strategy to solve the problems. The procedures are derived from activities and indicators in the i3 model.

Here is an example pattern, composed in the company, whose results were shown in previous section:

Purpose

The purpose of this pattern is to stabilize the inspection process in the organization.

Symptoms

- The inspection process has been introduced just recently.
- The motivation and purpose of inspections is unclear for participants.
- Not much information about the efficiency of the inspection process is available.

Possible reasons

The process description is not up-to-date.

- Checklists are not adequately made use of.
- Defects are not classified and analysed.
- There are no tools to manage the inspection information.

Related activities

- P.1. Establish and improve the inspection process.
- P.2. Organize inspection.

- P.3. Train participants.
- P.4. Support with computer tools.
- P.5. Maintain rules and checklists.
- P.6. Refine information.
- P.10. Categorize defects.

Action list

- Allocate adequate resources for process improvement.
- Make a schedule for the improvement.
- Evaluate the process.
- Obtain feedback and improvement ideas from the personnel.
- Consult an expert.
- Define defect classification categories.
- Create checklists for different roles.
- Update existing checklists.
- Define metrics to be gathered.
- Establish a procedure for recording metrics.
- Establish a procedure for analysing and reporting inspection data.
- Establish a procedure for recording feedback and improvement suggestions.
- Ensure that inspection process description is up-to-date in the quality handbook.
- Name an owner for the process.
- Define relations to other processes.
- Promote inspections.
- Write instructions for the inspection process.
- Arrange training.
- Evaluate the need for computer tools for the inspection process.
- Employ the tools.
- Arrange training for the users of the tools.

In addition to this company-specific pattern, we have seven generic patterns, which are listed in Table 2. The catalogue is a good foundation for inspection process improvement, although it is not complete. Experiences from future inspection capability assessments and improvement projects should be captured in patterns.

Table 2. Pattern catalogue.

Pattern name	The main goal of the pattern
Greed	Aims at finding more defects during inspections.
Early bird	Aims at finding defects at earlier stages of development.
Substance	Aims at finding more serious defects in inspections.
Comfort	Aims at making the inspection process easier to run.
Promotion	Aims at promoting the process so that it is carried out more often and in larger number of projects.
Wisdom	Aims at more understandable, transparent and effective inspection process.
Precision	Aims at making the process more rigorous, thus making it more effective.

According to our experiences, the pattern descriptions need to be accompanied with concrete examples of the execution of the specific improvement actions. Furthermore, it would be useful to summarize the possible consequences of specific actions, as using patterns may initiate changes in other processes as well. For example, inadequacies were found in the descriptions of other development processes during the composing of the inspection training material, and the relevant points were updated as well. As a result, the whole quality system of the company was able to benefit from the modifications.

After a half year of running, the improved process seems to perform well. There have been certain problems concerning data-gathering tools, and the importance of continuous training and coaching was underestimated first. However, as a consequence of training, metrics program and other improvements, the effort needed to accomplish inspections has been decreased 10-20 percent and discovered defects are more serious than before.

The experiment caused a number of adjustments in the pattern descriptions. Action lists needed to be represented in more convenient order, overlapping parts of separate patterns were removed and pattern descriptions were clarified in general. However, we discovered that the pattern catalogue offers a feasible foundation for the inspection process improvement.

4 Conclusions

We have evaluated the usability and relevance of the capability model by means of five experiments. The model has been updated according to the research results. The main problem in inspection improvement is that the definition of improvement steps requires a deep understanding of the inspection process, in which a certain amount of support is called for. This support can be provided in the form of patterns, which give advices and are suited for novices in the area. Improvement patterns are pre-defined sets of actions that can be taken to upgrade the inspection process. Each pattern has a clear goal and a set of symptoms for detecting an appropriate pattern. When using these patterns, the organization has to determine a general goal for its process improvement. Goals can be based on the assessment report, common sense and feelings arising during the assessment.

A further research topic would be to develop a SPICE-compliant version of the model. The SPICE-based assessment allows evaluation and improvement to be focused on a process (e.g. review/inspection) and thus there is a need for a tailored model. Changing the model to be compliant with generic SPI models causes changes in the number of base practices and indicators to be walked through, as that version starts with a core set of base practices (six activities in the middle bar, Figure 1) and related indicators. The evaluation of adequate education and training and of tool support comes later, when evaluating the process resource attribute at the third (established) level of capability.

The presentation of a new model always requires some justification as to why this new model is necessary. We can sum up the benefits of the present model by evaluating its usefulness in terms of three questions: (1) does the model help to find

the most important activities to improve? (2) Does the model help to find the best improvement suggestions? (3) Does the model work in practice?

We can answer positively to all these questions. The idea in i3 model is that, if we find indicators of a base practice, we then have some justifications for assuming the existence of that base practice. If some of them are at a low level or missing, they should be targets for improvement activities. Improvement patterns help in determining most suitable improvement actions. Furthermore, our experiments have discovered weak points in the companies' inspection processes which were also agreed on by the companies.

Finally, a warning is justified. Although we focus on the inspection process, we assume that a company which aims at improvement in this respect has already defined the whole software development process at an appropriate level. SME companies should not focus all their process improvement measures on inspection, but rather they should improve the whole development process, with inspections as one important part of this.

5 References

1. Grunbacher, P.: A Software Assessment Process for Small Software Enterprises. In Proceedings of the 23rd EUROMICRO. Conference, Budapest, (1997) 123-128
2. Batista, J., Dias de Figueiredo, A.: SPI in a Very Small Team: a Case with CMM. Software Process - Improvement and Practise, Vol. 5 (2000) 243-250
3. Potter, N.S., Sakry, M.E.: Making Process Improvement Work. A Concise Action Guide for Software Managers and Practitioners. Addison Wesley, Boston (2002)
4. Rico, D.F.: Software Process Improvement (SPI): Modeling Return on Investment (ROI). http://davidfrico.com/dacs02pdf.htm (2002)
5. Conradi, R., Marjara A., Skåtevik, B.: Empirical Study of Inspection and Testing Data at Ericsson, Norway. Proceedings of PROFES'99, Oulu (1999) 263-284
6. Gilb, T., Graham, D.: Software Inspection. Addison-Wesley, Wokingham (1993)
7. O'Neill, D.: Issues in Software Inspection. IEEE Software, Vol 14 (1997) 18-19
8. Weller, E.F.: Lessons learned from Three Years of Inspection Data. IEEE Software, Vol 10 (1993) 38-45
9. Fagan, M.E.: Design and Code Inspections to Reduce Errors in Program Development. IBM Systems Journal, Vol 15. (1976) 182-211
10. Perpich, J.M., Perry, D.E., Porter, A.A., Votta, L.G., Wade, M.W.: Anywhere, anytime code inspections: Using the web to remove inspection bottlenecks in large-scale software development. Proceedings of the 19th International Conference on Software Engineering (1997) 14-21
11. Tervonen, I., Iisakka, J., Harjumaa, L.: A Tailored Capability Model for Inspection Process Improvement. Proceedings of the Second Asia-Pacific Conference on Quality Software, (2001) 275-282
12. Ward, R.P., Fayad, M.E., Laitinen, M.: Software Process Improvement in the Small. Communications of the ACM, Vol 44 (2001) 105-107
13. Kuvaja, P., Similä, J., Krzanik, L., Bicego, A., Koch, G., Saukkonen, S.: Software Process Assessment and Improvement: The BOOTSTRAP Approach. Blackwell Publishers, Oxford (1994)
14. Burnstein I., et al., A Testing Maturity Model for Software Test Process Assessment and Improvement, Software Quality Professional, vol 1 (1999)

15.Gelperin, D., Hayashi A.: How to support better software testing. Application Trends, May (1996) 42-48
16.Emam El K., Drouin J., Melo W.,: SPICE: The Theory and Practice of Software Process Improvement and Capability Determination. IEEE Computer Society (1998)
17.Grady R., van Slack T.: Key Lessons in Achieving Widespread Inspection Use. IEEE Software, Vol 11 (1994) 46-57
18.Zahran, S.: Software Process Improvement, Practical Guidelines for Business Success. Addison-Wesley, UK (1998)
19.Buchman, C.D., Bramble, L.K.: Three-tiered Software Process Assessment Hierarchy. Software Process – Improvement and Practice, Vol 1 (1995) 99-106
20.Horvat, R.V., Rozman, I., Gyorkos, J.: Managing the Complexity of SPI in Small Companies. Software Process – Improvement and Practice, Vol 5 (1995) 45-54
21.Sakamoto, K., Nakakoji, K., Yasunari, T., Niihara. N.: Toward Computational Support for Software Process Improvement Activities. Proceedings of the 20th International Conference on Software Engineering, Kyoto (1998) 22-31
22.Tyran, C.K., George, J.F.: Improving Software Inspections with Group Process Support,.Communications of the ACM, Vol 45. (2002) 87-92
23.O'Neill, D.: National Software Quality Experiment: Results 1992-1996. Proceedings of Quality Week Europe Conference, Brussels (1997) 1-25
24.Wiegers, K.E., Sturzenberger, D.C.: A Modular Software Process Mini-assessment Method. IEEE Software, Vol 17 (2000) 62-29
25.Richardson, I.: SPI Models: What Characteristics are Required for Small Software Development Companies? Software Quality Journal, Vol 10 (2002) 101-114
26.Kelly, D.P., Culleton, B.: Process Improvement for Small Organizations. IEEE Computer, Vol 32 (1999) 41-47
27.Herbsleb, J., Zubrow, D., Goldenson, D., Hayes, W., Paulk, M.: Software Quality and the Capability Maturity Model. Communications of the ACM, Vol 40 (1997) 25-29
28.Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns – Elements of Reusable Object-Oriented Software. Addison-Wesley (1995)
29.Appleton, B.: Patterns for Conducting Process Improvement. PloP'97 Conference, http://www.cmcrossroads.com/bradapp/docs/i-spi/plop97.html (1997)
30.Rising, L.: Patterns: A way to reuse expertise. http://www.agcs.com/supportv2/techpapers/patterns/papers/expertise.htm (1998) (referenced 01.03.2003)
31.Sakamoto, K., Kishida, K., Nakakoji, K.: Cultural Adaptation of the CMM: A Case Study of a Software Engineering Process Group in a Japanese Manufacturing Company. In: Fugetta, A., Wolf, A. (eds.): Software Process. John Wiley & Sons Ltd, West Sussex (1996) 137-154